Various Flatpak Tips and Tricks

Adrian Kosmaczewski

2022-07-01

I use Flatpak a lot, on my work and personal laptops. I had to find a few tips and tricks to make it work the way I wanted, using all the apps I prefer.

This page summarizes the things I've done to make Flatpak work for me, and hopefully, they'll be useful to you too.

Toggle External Editing in Joplin

Due to sandboxing, when installing Joplin via Flatpak, the "Toggle External Editing" command does not work by default. There was an issue about this, and now the project README explains how to make it work.

If you want to use an app not installed through Flatpak (for example, in this case, Visual Studio Code), you have to open the Joplin settings and use the following parameters in the "Text editor command" section of the General tab:

- Path: `/bin/flatpak-spawn`
- Arguments: `--host /usr/bin/code`

However, if you would like to use a Flatpak-installed editor such as Apostrophe or Typora, use the following parameters:

- Path: `/bin/flatpak-spawn`
- Arguments: `--host /bin/flatpak run --filesystem=xdg-config/joplin-desktop io.typora.Typora`

Of course, the command above requires you to install the corresponding app:

```
$ flatpak install flathub io.typora.Typora
```

If you prefer Apostrophe, it's the same but with `org.gnome.gitlab.somas.Apostrophe` as app identifier.

Calibre Night Mode

If you install Calibre with Flatpak, unfortunately, it does not read the current system theme at startup (or at runtime) and stays stubbornly white at all times of the day, even if the other apps are all dark and cozy.

If you would like to use it in night or dark mode, type the following command:

```
$ flatpak override --user --env=CALIBRE_USE_SYSTEM_THEME=0 \
    --env=CALIBRE_USE_DARK_PALETTE=1 com.calibre_ebook.calibre
```

After this, launching Calibre will always use the dark mode. Run the same command with `CALIBRE_USE_DARK_PALETTE=0` or `CALIBRE_USE_SYSTEM_THEME=1` if you would like to go back to the default. Unfortunately, the latter always defaults to the non-dark version, even if the system has switched to dark more.

Update, 2022-07-22: Calibre has been recently updated to 6.0 and it does not take into account any more the environment variables to select day or night mode; you can now select the mode directly in the preferences.

Signal Desktop Start in Tray

Signal Desktop for Linux has a toggle to start minimized in the system tray, but unfortunately, it's not exposed in the preferences (unlike its Windows counterpart). The trick consists in creating a shell shortcut and adding a parameter in the command. This should work for both GNOME and KDE.

First, create a duplicate of the shell shortcut:

```
$ cp /var/lib/flatpak/exports/share/applications/org.signal.Signal.desktop ~/.local/share/ap
$ vim ~/.local/share/applications/org.signal.Signal.desktop
```

Then add the `--start-in-tray` option in the `Exec` line:

```
[Desktop Entry]
Name=Signal
Exec=/usr/bin/flatpak run --branch=stable --arch=x86_64 --command=signal-desktop --file-forw
Terminal=false
Type=Application
Icon=org.signal.Signal
StartupWMClass=Signal
Comment=Private messaging from your desktop
MimeType=x-scheme-handler/sgnl;x-scheme-handler/signalcaptcha;
Categories=Network;InstantMessaging;Chat;
X-Desktop-File-Install-Version=0.26
X-Flatpak-RenamedFrom=signal-desktop.desktop;
X-Flatpak=org.signal.Signal
```

The desktop environment picks up the new command automatically, and it's ready to use. This is explained here.

The Flatpak Filesystem

In all of the explorations above, knowing how Flatpak organizes files is very useful. For example, that's how you find GIMP's configuration files, located in `~/.var/app/org.gimp.GIMP/config/GIMP/2.10`; which can be useful, for example, if you want to change the keyboard shortcuts to be more similar to Photoshop's.

There's a wiki page about it in the GitHub project, and it looks a bit like this:

```
.local/share/flatpak/
  repo
      config
      objects
          00

          ff
      refs
          heads
          remotes
              test-repo
      state
  runtime
      org.gnome.Platform
          x86_64
              3.14
                  active -> 71885e962e0daa8635cc7f33…
                  71885e962e0daa8635cc7f33…
                      files

                      metadata
                  origin
  app
      org.gnome.GEdit
          current -> x86_64/master
          x86_64
              master
                  active -> cf22ec5c375bba629b33385a…
                  cf22ec5c375bba629b33385a…
                      export

                      files

                      metadata
                  origin
  exports
      share
          applications
          dbus-1
          icons
.var/
  app
```

```
org.gnome.GEdit
    data
    config
    cache
```

Update, 2022-07-22: I also discovered `flatpak uninstall --unused` to remove unused runtimes.