Enabling GPU acceleration in Firefox 76 under Ubuntu 20.04 with Wayland

emkor.eu/posts/gpu-acceleration-in-firefox76

Firefox 76 came out yesterday, bringing to the table - amongst more widely known features - initial support for hardware acceleration of VP9-encoded videos for Wayland. Translating to human language: now you can watch YouTube videos without making your laptop fans spinning at full speed. Please note, however, the word *initial* there - few tweaks are gonna be required to pull the hardware acceleration off.

This guide is a compilation of tips and tweaks from various sources, like Martin Stransky's Blog, who is one of the Firefox developers, Arch wikis (1, 2) and some Reddit threads 1

Prerequisites

First, if you have NVIDIA GPU, there's no hope. As you probably know, there's not so much love for NVIDIA in open-source world, and this feature is indifferent. Other than that, if you happen to have non-ancient AMD/ATI or Intel iGPU, you're in luck.

Now, to use your GPU, you need three things: latest Firefox, VA-API support in your OS, and a Wayland-based compositor. First one is just `sudo apt-get update && sudo apt-get install -y firefox` . Fortunately, latter requirements comes by default with the new Ubuntu 20.04, which despite some serious controversies related to its Snap store, still seems to be most-polished distro out there.

Now regarding the compositor: Wayland is still a little experimental, "glitches ahead" and all, but I'm using it for over a week now without any major issues or annoying hiccups. On plus side, it's much smoother than x11 and allows for 4-finger gestures for switching workspaces - just like on macOS. Your mileage may vary, of course.

To start Wayland session, simply choose "Ubuntu on Wayland" on OS login screen (gear icon in bottom-right corner). From now on, Wayland is the default, until you switch to classic, x11-based compositor.

Second thing is support for VA-API on your OS. Ubuntu comes with default CLI utility, `vainfo` (you can install it with `sudo apt-get install vainfo`), just for the purpose of retrieving info on VA-API status. Your output should look like this:

```
libva info: VA-API version 1.7.0
libva info: Trying to open /usr/lib/x86_64-linux-gnu/dri/iHD_drv_video.so
libva info: Found init function __vaDriverInit_1_7
libva info: va_openDriver() returns 0
vainfo: VA-API version: 1.7 (libva 2.6.0)
vainfo: Driver version: Intel iHD driver for Intel(R) Gen Graphics - 20.1.1 ()
vainfo: Supported profile and entrypoints
      VAProfileMPEG2Simple         : VAEntrypointVLD
      VAProfileMPEG2Main           : VAEntrypointVLD
      VAProfileH264Main            : VAEntrypointVLD
      VAProfileH264Main            : VAEntrypointEncSliceLP
      VAProfileH264High            : VAEntrypointVLD
      VAProfileH264High            : VAEntrypointEncSliceLP
      VAProfileJPEGBaseline        : VAEntrypointVLD
      VAProfileJPEGBaseline        : VAEntrypointEncPicture
      VAProfileH264ConstrainedBaseline: VAEntrypointVLD
      VAProfileH264ConstrainedBaseline: VAEntrypointEncSliceLP
      VAProfileVP8Version0_3       : VAEntrypointVLD
      VAProfileHEVCMain            : VAEntrypointVLD
      VAProfileHEVCMain10          : VAEntrypointVLD
      VAProfileVP9Profile0         : VAEntrypointVLD
      VAProfileVP9Profile2         : VAEntrypointVLD
```

The presence of VP9 profiles is crucial for YouTube videos.

If your output lacks this information and you still believe your GPU is able to decode video with hardware support, visit this Arch wiki for further instructions, having in mind that you use Debian-derivative OS, so commands and package names are different.

Tweaking Firefox

Next step is to tell Firefox to use Wayland. Just edit (with sudo) its `/usr/share/applications/firefox.desktop` file and prepend `firefox` executable with specific flags, so such line:

```
Exec=firefox %u
```

becomes

```
Exec=env MOZ_ENABLE_WAYLAND=1 MOZ_WEBRENDER=1 MOZ_DBUS_REMOTE=1 firefox %u
```

and so on; **repeat that for all 3 lines containing** `Exec=` **.**

Now restart Firefox and check if those flags actually work by going to `about:support` and checking `Window protocol` - its value should be `wayland` (or similar), not `x11` . If that's not the case, please note that Firefox downloaded directly from Mozilla website instead of installed from PPA (which translates to "not installed with `apt-get` ") might have trouble with using those flags you just pasted.

The last step is to enable few flags in Firefox config, so while being in the browser, go to `about:config` URL and enable these flags by settings them to `true` :

widget.wayland-dmabuf-vaapi.enabled
gfx.webrender.all

and set this one below to `false` :

media.ffvpx.enabled

Verification

Restart the browser. Now it's time for one last verification that everything is correct: go to `about:support` and verify those params:

- `Compositing` should be `WebRender`
- `WebGL 1 Driver Version` should say `OpenGL something something`
- `Window Protocol` should be `wayland/drm` or similar

If that's the case, you're ready to head to YouTube and play some 4K video in fullscreen. In my case, with i7-8650u laptop, watching 4K video on 1440p external screen causes Firefox process to eat ~50% CPU measured in htop (before tweaks it was between 100-150%).

Overall rendering performance should be improved, too, so scrolling should have much less latency and be much smoother, just like on Windows or macOS. Before these tweaks my Firefox 76 installation scored around 86 points in MotionMark benchmark, after applying tweaks the result was 315 points - even higher than Chromium (207), which is known to outperform Firefox severely in rendering benchmarks.

One last thing to worry about is that each time you update Firefox, custom flags added in `firefox.desktop` will be gone and need to be updated.

Have fun browsing