Subject: Re: Statically linked Web Browser
Posted by seasoned_geek on Fri, 14 Sep 2018 01:54:34 GMT
View Forum Message <> Reply to Message

Novo wrote on Thu, 13 September 2018 16:41seasoned_geek wrote on Thu, 13 September 2018 16:40
Statically linked would, by definition, create a one fat "include all" module. It is the only reliable way to create a single executable which runs on many different versions of the same OS or even different OS's. A good static linked executable will run on both a Debian and RPM based distribution without having to install anything else. It's a real PITA, but the only way to have sanity. Otherwise you release something into the wild and get clubbed to death by willy-nilly updates to dynamically linked libraries which break things without a care in the world.

Thanks again for your help.

I was trying to tell you something different. According to this: CEF 3 is a multi-process implementation based on the Chromium Content API.
There will be several executables ... But they will be statically linked ...
Multi-process communication is slow. If you need something faster and smaller there are commercial apps like this one: https://coherent-labs.com/coherent-gt/
These guys have hardware-accelerated graphics in addition to HTML5. And they sell a library.

Sell, purchase, buy, license, royalty or any other term which signifies the parting with coin for software use of any kind won't fly here. I don't blame them really. Over my 30+ years in IT, I and my clients have paid far too much money to companies which simply disappeared not long after cashing the check. We may have to abandon C++ entirely for this iteration of what they want. Might have to go with Electron.

Multi-process communication should never be slow. If it is, the thing was poorly architected. I wrote roughly have of the Qt based UI source code for this device. We had lots of different executables running communicating via message queue on a CPU which did not have a GPU and we could still monitor all of the patient vitals with a snappy screen display. Everything communicated via COOA objects over a message queue. It was blindingly fast. Then again, we built our own linux taking out everything we did not need or want.

Thank you for all of your assistance.