Subject: Re: Statically linked Web Browser
Posted by seasoned_geek on Fri, 14 Sep 2018 12:46:29 GMT
View Forum Message <> Reply to Message

dolik.rce wrote on Fri, 14 September 2018 00:10Novo: I'm sorry, but I have to slightly correct you :) Multi-process is not always the same thing as multi-executable... Actually it is much easier to create multi-process architecture by simply forking single executable - it makes the IPC and data sharing much simpler. I'm looking at my Chrome now and it uses single executable almost for everything, the only exceptions being NaCl support and sandbox (for obvious security reasons).

Also, I guess for the OPs problem it doesn't matter much if there were multiple binaries. The main idea is that the executables must be self-contained and not depend on anything outside, that could be updated by the OS.

seasoned_geek: BTW: Did you consider other ways to encapsulate your application so it can't be broken by updates? Technologies like Flatpak or Snap wrap all the dependencies of your application into single container, so it might might give you what you want, without resorting to static linking. Or you could perhaps use Docker, which provides the same or even higher level of separation, with added benefit that you could only create a single image for all the distributions you want to support.

Honza

Honza,

No, just starting. Back when this was first done Qt was the only way. Electron and all of these other things weren't even a gleam in the eye. I'm always a bit (okay 7 continents) leery of anything new and trendy the script kiddies are all gaga about.They tend to flash and disappear. Back when this was originally done there were more than a dozen custom ISO build tools the Internet raved about. Guess what? When I went to use them every one of them was no longer supported. I had to build that script by hand.

Without trying to give away the farm. This is a custom application which must run on every flavor of Windows from XP forward (both 32 and 64-bit) on every form of hardware from desktop to laptop to those sub-$90 2:1 computers Walmart sells. It must run on Chrome Book, Android and a rash of other platforms, both 32 and 64-bit.

For all of those it must be a stand alone app which gets installed in such a way to run at startup locking the machine into the application and only those knowing the super secret password(s) can exit it. Oh, and the content of the application is dynamically customizeable at the central Web site before download. Oh, and a new version of this has to be installable over the top of the old by someone as technically inept as a Keller MBA. Third oh. None of the machines on whence this is installed have ___ANY__ Internet connection. There is one machine locked away on site with Internet access. The application is configured, downloaded to it, then copied to thumb/DVD/whatever and transferred via sneaker-net to all of the other machines.

Linux has even more oh's.

Stand alone must run on both Debian and RPM both 32 and 64-bit. Again, boot into it just like above.

Linux has the option of configuring a complete ISO with this application already installed and configured. That is this application with their current customized data, so we are dynamically popping apart ISOs, replacing things, then putting the ISO back together before download. Not just one ISO. Any version going back to 2012.

We aren't talking one-offs here. We are talking thousands at a time. Tens of thousands during certain times of the year. I am not at liberty to say what the application is, but, most of you, at some point in your life have seen some flavor of it.

Yikes! I forgot the biggest Oh of them all! If the user chooses one certain option the app hides itself for a certain amount of time and lets the user play with the computer in a completely captive account where any and all changes are written to a ram disk (or something like it) and instantly discarded once the application restarts.