
Subject: Statically linked Web Browser

Posted by [seasoned_geek](#) on Thu, 13 Sep 2018 13:47:02 GMT

[View Forum Message](#) <> [Reply to Message](#)

Is there an example anywhere of a U++ project which creates a statically linked Web browser which can automatically load locally stored pages?

Basically, a long time ago I did something which pretty much just captivated/trapped this example in an environment which could do nothing else.

<https://doc-snapshots.qt.io/qt5-5.11/qtwebengine-webenginewidgets-simplebrowser-example.html>

It sucked when we had to support it in multiple ISOs because I kept having to pull in hundreds of dependencies to a new ISO.

What I'm now looking for is a cross platform tool which will allow me to create one executable which can run, without having to install any dependencies, from Ubuntu 12 - 18.x. I want to be able to cross compile the same set of source from Linux to Android and from Linux to Windows.

Is there any U++ work on creating a STATICALLY LINKED browser? I don't care about one which dynamically links. I've already got that train wreck. I need to find an unwreckable train.

Subject: Re: Statically linked Web Browser

Posted by [Novo](#) on Thu, 13 Sep 2018 17:13:47 GMT

[View Forum Message](#) <> [Reply to Message](#)

IMHO, this problem is not related to UPP. You just need to fix a make file or a build script for your Web Browser to create a statically linked executable. After that you will just need to add a list of needed static libraries to your project in TheIDE. But the first thing to do is to compile a standalone static version of a Web Browser as a proof of a concept.

Subject: Re: Statically linked Web Browser

Posted by [seasoned_geek](#) on Thu, 13 Sep 2018 19:15:20 GMT

[View Forum Message](#) <> [Reply to Message](#)

Thank you for the reply. Not what I was asking, but thank you for the reply.

>>You just need to fix a make file or a build script for your Web Browser to create a statically linked executable.

What we currently have is basically this.

<https://doc.qt.io/qt-5/qtwebengine-webenginewidgets-simplebrowser-example.html>

Although ours was built many years ago with a much older version of Qt. We don't have "a browser." I was asking if U++ bundled with it a browser library like Qt does and if anyone here had tried to compile something like that "Simple Browser" example program with U++ then if someone had tried it statically.

There exists (or at least existed) a technical reason why the Qt browser library could not be statically linked. I forget what it was, but, I'm certain nobody has fixed it. When I actually start on this project I will go down that road once again, but, I was hoping against hope that U++ had a path where one could build such an application from that which it contains. Ideally one which would also allow something along the line of WebSockets like Qt does so the Web pages could once again use local storage, etc.

This is a captive application. It doesn't connect to the Web. It force loads a continuous display of various locally stored Web pages. Why Web pages? When a company has several hundred Web developers and no embedded systems coders you're going to get a Web based "solution."

Looking into slightly better browser options now.

Thanks though.

Subject: Re: Statically linked Web Browser
Posted by [dolik.rce](#) on Thu, 13 Sep 2018 20:08:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Roland,

There is a ChromiumBrowser package in bazaar. It is based on Chromium embedded framework, which seems to be statically linkable.

However note that I have never tried it and making both the CEF and the U++ package work with static linking might present some (or a lot of) additional work.

Best regards,
Honza

Subject: Re: Statically linked Web Browser
Posted by [Novo](#) on Thu, 13 Sep 2018 20:10:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

Have you checked "bazaar/ChromiumBrowser"?

I'm not an expert in CEF, but I found this:

<https://stackoverflow.com/questions/28697196/compile-chromium-embedded-3-project-statically-with-libcef-static-lib>

Theoretically, you should be able to compile CEF as a static library.

AFAIK, CEF has a multi-process architecture, so, you won't be able to create one fat "include all"

executable.

You can check for similar functionality with other web-browsers. Maybe, they allow to use single-process architecture. I'm not an expert in Web-UI. A huge Web-UI based project my employer was working on for many years has failed, although we had a lot of success with Flash-based vector GUI, which we implemented as a static library.

Qt browser library should use a third-party rendering engine. It is practically impossible to develop another decent in-house rendering engine and JavaScript engine.

It is much easier to create regular GUI app with UPP and convert it into Web-GUI using TURTLE (example: reference/WebWord) :)

Hope this helps.

Subject: Re: Statically linked Web Browser

Posted by [seasoned_geek](#) on Thu, 13 Sep 2018 20:35:48 GMT

[View Forum Message](#) <> [Reply to Message](#)

Oh, it is going to be a lot of work, no doubt about it. Probably as much work as I had last time creating a single Linux executable and creating multiple custom ISOs with all of the dependencies installed. That was joyous. Took several weeks since we were spanning about 8 different OS versions.

Thanks for your input.

Subject: Re: Statically linked Web Browser

Posted by [seasoned_geek](#) on Thu, 13 Sep 2018 20:40:27 GMT

[View Forum Message](#) <> [Reply to Message](#)

Novo wrote on Thu, 13 September 2018 15:10: Have you checked "bazaar/ChromiumBrowser"?

I'm not an expert in CEF, but I found this:

<https://stackoverflow.com/questions/28697196/compile-chromium-embedded-3-project-statically-with-libcef-static-lib>

Theoretically, you should be able to compile CEF as a static library.

AFAIK, CEF has a multi-process architecture, so, you won't be able to create one fat "include all" executable.

You can check for similar functionality with other web-browsers. Maybe, they allow to use single-process architecture. I'm not an expert in Web-UI. A huge Web-UI based project my employer was working on for many years has failed, although we had a lot of success with Flash-based vector GUI, which we implemented as a static library.

Qt browser library should use a third-party rendering engine. It is practically impossible to develop another decent in-house rendering engine and JavaScript engine.

It is much easier to create regular GUI app with UPP and convert it into Web-GUI using TURTLE (example: reference/WebWord) :)

Hope this helps.

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.

Subject: Re: Statically linked Web Browser
Posted by [Novo](#) on Thu, 13 Sep 2018 21:41:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

seasoned_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.

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.

Subject: Re: Statically linked Web Browser
Posted by [dolik.rce](#) on Fri, 14 Sep 2018 05:10:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

Novo: 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

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 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 customizable 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.

Subject: Re: Statically linked Web Browser
Posted by [Novo](#) on Fri, 14 Sep 2018 15:20:12 GMT
[View Forum Message](#) <> [Reply to Message](#)

dolik.rce wrote on Fri, 14 September 2018 01:10Novo: I'm sorry, but I have to slightly correct you :) Multi-process is not always the same thing as multi-executable...
I suspected something like this :)
Anyway, building of a static version of a Web Browser (or CEF) is a generic problem and it is not related to UPP. Just build a static version of each required library and pass them in a correct order to a linker. :)

Subject: Re: Statically linked Web Browser
Posted by [seasoned_geek](#) on Sun, 16 Sep 2018 13:26:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

Novo wrote on Fri, 14 September 2018 10:20dolik.rce wrote on Fri, 14 September 2018 01:10Novo: I'm sorry, but I have to slightly correct you :) Multi-process is not always the same thing as multi-executable...
I suspected something like this :)

Anyway, building of a static version of a Web Browser (or CEF) is a generic problem and it is not related to UPP. Just build a static version of each required library and pass them in a correct order to a linker. :)

Actually the entire gist of my original question makes it a UPP problem.

Qt has WebEngine, WebSockets, WebView and a ton of other classes built into the product. While it may be difficult to statically link such stuff (impossible for a token few things), it is all part of the package.

From the responses here, UPP has not rolled in any OpenSource C++ Web library. Oh, for sure, a few http and other calls to retrieve things, but nothing which could be used to display a local Website. I'm making that assumption by the responses I've gotten here making it sound like a Web browser is this foreign country I must find without a map.

That is fine. It does answer my question. UPP shouldn't be on the short list of things to research for this client.

Thanks again for all of the help! You saved me a lot of time.
