

Concerning the Windows build we had the following issue:

The appveyor build of the plugin takes at least 40 Minutes. There is an opportunity to build the dll-file on its own (without the whole Wireshark), which is much less time consuming.

I tested this in Sprint #2 locally. However this needs one initial build including most of the Wireshark application. Since we have no persistence by using appveyor, this is not possible for us.

One approach to bypass this obstacle would be to offer the whole Wireshark-build as a download so that msbuild would recognize this and would only build the manipulated dll part.

This would be rather similiar to the local scenario I described above. But with this approach we would need to build the whole Wireshark anyway from time to time, because we need the guarantee that our Wireshark version is up-to-date. This means pulling from the wireshark repository and build it with the most recent source files. Besides this downloading this Wireshark-Build for every new building process of the dll would not be much faster than our conventional way of building it via Appveyor. Especially if we would use a premium account for appveyor, the whole Building process would be even faster. So that's why we would rather keep our conventional building-way.

When we want to keep on with this method there are two issues:

1. Since we are doing CI, every push on github would result in a new build. This could be a real mess if more than one of us push at the same time for instance. So what we want to do is making use of "suitable-for-windows-build"-tags in github, so that only commits with an appropriate tag would start an appveyor-build. This can be achieved with the APPVEYOR_REPO_TAG environment variable.

2. As I told you, the actual build takes about 40mins, but can be hastened through appveyor-premium-accounts. There is a possibility to get a free trial premium account for two weeks. So we probably need to create several of them