Subject: UPDATE - Folder restructuring in dev branch
Posted by Ralf Kliemt on Fri, 23 Feb 2018 15:30:17 GMT
View Forum Message <> Reply to Message

Dear developers.

I submitted a quite strong restructuring of our folder in Pandaroot - including adjusting those hard-coded paths. This was planned ahead already in Sep. 2017 in Novosibirsk and was scheduled to be close to our change to git.
Everybody who works on a fork right now should carefully rebase as soon as possible!

Please let us know of any inconsistencies or issues.

Cheers!
Ralf

Subject: Re: UPDATE - Folder restructuring in dev branch
Posted by Michael Papenbrock on Thu, 01 Mar 2018 14:45:42 GMT
View Forum Message <> Reply to Message

Hi Ralf,

are there any plans to have a consolidated include path as well, at least in the build directory? The path resolution for e.g. code completion / IntelliSense can be a bit tedious.

Michael

Subject: Re: UPDATE - Folder restructuring in dev branch
Posted by Ralf Kliemt on Thu, 01 Mar 2018 16:46:29 GMT
View Forum Message <> Reply to Message

Hi Michael,

Nice idea. I suggest you open an Issue on the website. Do you have already an idea how to implement that in CMake?

Cheers!
Ralf

Subject: Re: UPDATE - Folder restructuring in dev branch
Posted by Michael Papenbrock on Fri, 02 Mar 2018 11:12:44 GMT
View Forum Message <> Reply to Message

Hi Ralf,

the answer to that is a clear yes/no.

Yes in the sense that I have written something like that a long time ago with vanilla CMake. Have a look at the following code:

```
foreach(source ${Project_Sources})
  configure_file(${CMAKE_CURRENT_SOURCE_DIR}/${source}.h
${Project_BINARY_DIR}/include/${source}.h COPYONLY)
endforeach(source ${Project_Sources})
```

I put this into each of my subdirectories' CMakeLists.txt and it would copy the header files into an include directory in the build folder. It was a quick and dirty hack, but still a quality-of-life improvement.

No in the sense that we are using a more customized flavour of CMake by using macros inherited from FairRoot. Therefore, we pass sources slightly differently than in vanilla (including the file ending instead of just the base name). I don't know if that would really pose a problem because you probably can still tokenize them and get the base name that way.

A more sophisticated approach might be to define a "make install" target similar to FairSoft and FairRoot. I tried this earlier today and found that it seems to be already inherited from FairRoot, but only a few libraries and no headers were copied to the install directory. Here I'm not really sure what we would need to add on our side, i.e. just the vanilla CMake commands or something customised. Are you perhaps aware of any FairRoot documentation concerning their CMake macros?

Cheers,
Michael

Edit: Forgot to link the gitlab issue:
https://pandaatfair.githost.io/PandaRootGroup/PandaRoot/issues/134
