

SciSoft - Feature #22642

Would like an initial version of a HighFive UPS product for DUNE use

05/28/2019 03:41 PM - Thomas Junk

Status:	Assigned	Start date:	05/28/2019
Priority:	Normal	Due date:	
Assignee:	Lynn Garren	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:		Spent time:	0.00 hour
Description			
<p>Jeremy Hewes asked me if we can distribute HighFive along with the DUNE software stack. HighFive is a C++ header-only external package that provides user-friendly interfaces to HDF5, a product we already set up as a dependency of caffe, and which exists in the larsoft CVMFS repository.</p> <p>Lynn writes this:</p> <p>So the situation is that DUNE would like to use HighFive. You should make an official request for a ups product. File a redmine issue. I suspect that we would help you sort everything out the first time around. But long term maintenance will definitely fall to DUNE.</p> <p>So the location of HighFive will have to be in /cvmfs/dune.opensciencegrid.org/products if DUNE is to maintain it.</p> <p>The source of HighFive can be found at:</p> <p>https://github.com/BlueBrain/HighFive</p> <p>It is a header-only package that Jeremy has been using in his own private test release on his own computer. I expect it therefore to be null-flavored. It should set up hdf5 as a dependency.</p> <p>We also need to be able to make a tarball for distribution on SciSoft, and adjust the dunetpc manifest so PullProducts works. Some of this work may fall to DUNE software coordinators (i.e. me).</p>			
Related issues:			
Related to SciSoft - Support #23098: H5CPP does not compile with art's compil...		Assigned	08/12/2019

History

#1 - 06/03/2019 10:30 AM - Kyle Knoepfel

- Assignee set to Lynn Garren
- Status changed from New to Assigned

We will setup a meeting including Marc P. and Chris G.

#2 - 06/03/2019 10:48 AM - Thomas Junk

Yes, we brought this up at the DUNE computing meeting just now and came to the conclusion that further discussion among experts is needed. Brett Viren should be invited as well. And Jeremy.

#3 - 06/11/2019 04:17 PM - Brett Viren

Just a few points from me (maybe in lieu of me making some slides):

- I want to add HDF5 support to Wire-Cell Toolkit and want a "good" C++ API to the HDF5 C library.
- Only HighFive and h5cpp looked good to me. h5cpp's code generator looked interesting but not required. I am happy to go along with others on a choice between these two options.
- WCT's HDF5 support will be segregated to a git submodule. Building that will be a compile time option (like it is now for the ROOT dependency) and HDF5 will not required for any "core" functionality

(just the I/O).

- Packages using HighFive could take care not to expose any HighFive headers via their own headers. This would make HighFive only a compile-time dependency and thus a HighFive UPS product would not strictly be needed. Of course, HDF5's lib is still needed, but it's already there, so no problem. I can commit to not exposing HighFive headers when adding the HDF5 dependency to WCT. In this case, it may be easiest to "vendor" the headers into the package (ie, into the future wire-cell-hdf5 git submodule).

#4 - 06/11/2019 05:09 PM - Thomas Junk

A comment on the last of Brett's bullet points regarding skipping the HighFive UPS product. People debugging a program set up from CVMFS will need access to the source code, which is provided in the UPS products for our other dependencies. We'd also still have to provide HighFive in scisoft so that people compiling on their laptops with a distribution downloaded from SciSoft will be able to work.

#5 - 06/17/2019 11:08 AM - Lynn Garren

A meeting is scheduled for 1 p.m. June 18 in the Quarium (WH8X).

#6 - 06/18/2019 01:39 PM - Lynn Garren

The proposal is to provide hdf5 and [h5cpp](#) for testing. If those products work for everyone, then they will be added to the larsoft distribution.

#7 - 07/03/2019 02:08 PM - Lynn Garren

- % Done changed from 0 to 100

- Status changed from Assigned to Resolved

Apologies for the delay.

h5cpp v1_10_4_1 is now available on cvmfs. Please let us know how it works for you.

Note that this is a null flavored product. You will need to setup h5cpp v1_10_4_1 and also setup hdf5 v1_10_5 (with appropriate qualifiers).

#8 - 07/22/2019 03:42 PM - Lynn Garren

- Status changed from Resolved to Feedback

Has anyone been able to test these products yet? Recall that the proposal is to include these products in the larsoft distribution if there are no problems. That step is waiting for feedback from users.

#9 - 07/29/2019 08:13 AM - Jeremy Hewes

- File build.log added

Sorry for the delay on this – I've put together a module that uses H5CPP instead of HighFive, and it appears H5CPP is not compliant with Art's strict compilation requirements. I see many errors due to unused variables, comparison between signed and unsigned integers, enums not handled in switch statements, etc. I've attached a log file containing the output when I try to compile.

#10 - 08/05/2019 02:03 PM - Lynn Garren

Jeremy, can you provide instructions to reproduce the problem? Much thanks.

#11 - 08/07/2019 11:31 AM - Jeremy Hewes

Sure – I committed the module dune/tpc/dune/CVN/art/CVNH5CPP_module.cc to a new feature branch (feature/jhewes15_h5cpp). If you try building that branch, you should be able to reproduce the errors I see.

#12 - 08/12/2019 10:21 AM - Kyle Knoepfel

- Related to Support #23098: H5CPP does not compile with art's compilation flags added

#13 - 09/10/2019 01:49 PM - Jeremy Hewes

I know Chris Green and Marc Paterno at the lab have an HDF5 interface called ntuple which is designed specifically for HEP applications. The hep_hpc external is available in the NOvA UPS environment; would it be possible to have a version of this also made available in the DUNE environment? It looks like a viable alternative to C++ interfaces like H5CPP and HighFive, and hopefully shouldn't have any issues interfacing with Art.

#14 - 09/10/2019 01:49 PM - Jeremy Hewes

Jeremy Hewes wrote:

I know Chris Green and Marc Paterno at the lab have an HDF5 interface called ntuple which is designed specifically for HEP applications. The hep_hpc external is available in the NOvA UPS environment; would it be possible to have a version of this also made available in the DUNE environment? It looks like a viable alternative to C++ interfaces like H5CPP and HighFive, and hopefully shouldn't have any issues interfacing with Art.

<https://www.hdfgroup.org/wp-content/uploads/2019/01/HDF5Ntuple-final.pdf>

#15 - 09/16/2019 10:22 AM - Kyle Knoepfel

- Status changed from Feedback to Assigned

Files

build.log	62.9 KB	07/29/2019	Jeremy Hewes
-----------	---------	------------	--------------