

artdaq - Bug #22777

DAQInterface makes inappropriate assumptions about versions of software being used

06/20/2019 10:32 AM - Kurt Biery

Status:	Reviewed	Start date:	06/20/2019
Priority:	Normal	Due date:	
Assignee:	John Freeman	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:		Co-Assignees:	
Experiment:	-		
Description			
Just because a particular version of a software package is included in the "srcs" area of an MRB development area, it doesn't mean that that version was used in a particular run.			
As a simple example from protoDUNE... If I cd into "srcs/<packageName>" and "git checkout <some old version>", and then take a run without rebuilding (which would likely fail), DAQInterface reports that the older version was used, when in fact, it wasn't.			

History

#1 - 06/20/2019 10:34 AM - Kurt Biery

I should have said that I look in metadata.txt to determine the reported versions of software.

#2 - 06/20/2019 11:34 AM - John Freeman

It's true that the git commit hashes which are saved in the run record (metadata.txt) are literally just the most recent commit hashes found in the ./srcs area. To get these values to tell us for certain what code was used, we need to ensure that the built code matches up with these commit hashes. So, I can do the following:

- Have DAQInterface refuse to boot if it discovers that there are code edits which haven't been locally committed in ./srcs/<packagename>
- If the code edits ARE all committed, then have DAQInterface refuse to boot if it discovers via a diff that the code in the local ups area (or, in fact, whatever ups area gets set up by the setup script) doesn't match with the code in the ./srcs area

#3 - 06/20/2019 12:20 PM - Kurt Biery

I don't want any changes that make the system harder to use, so any options with "DAQInterface refuses to boot" in the description don't seem like good ones to me.

I like the idea of comparing what is in the UPS package "source" area with what is in the MRB sources area. If those match, then it seems like it should be safe to use some of the git commit information in what is reported as the version used. If not, then we probably just need to fall back to the version number that is declared in the UPS package.

As a reminder, the information that is stored in the raw data file by the BuildInfo module looks reasonable:

```
-----  
Package      |Version      |Timestamp  
artdaq-core  |v2019_05_28_for_dune-artdaq_A|29-May-2019 17:49:10 UTC  
artdaq       |v2019_06_19_for_dune-artdaq_A|19-Jun-2019 22:12:35 UTC  
dune-raw-data|v2019_05_29_for_dune-artdaq_A|20-Jun-2019 13:21:27 UTC  
dune-artdaq  |v2019_06_09_develop_June2019|20-Jun-2019 13:21:29 UTC  
-----
```

#4 - 06/20/2019 01:04 PM - John Freeman

After discussing this with Kurt, what we decided was the following: if a package has a git repo in the development area, then for that package DAQInterface will perform a diff between the git repo's most recent local commit hash (i.e., the commit which gets saved in metadata.txt) and the code that was found in ups; the number of lines which were different will get saved alongside the git commit hash. Only if the number of differing lines was 0 could we say for sure what the code was, otherwise we have the number of differing lines as a heuristic for figuring out how much (or little) information the commit conveys about the code during the run.

#5 - 07/09/2019 04:22 PM - John Freeman

- Status changed from Assigned to Work in progress

This post describes the behavior of commit a3bbc0746c0985bc1eda7f5c6583da181cda0a42 at the current head of feature/issue22777_more_accurate_record_of_code.

On the develop branch, when saving a package's code info to the run record, it will just save either (A) the most recent commit and the commit comment, if the git repo is available, or (B) the package version in ups, if the git repo's not available. However, on the feature/issue22777_more_accurate_record_of_code branch, if the git repo is available it will also record the number of lines added/removed via edits to the most recent commit. Also, regardless of whether a git repository is available, if DAQInterface finds that a file with the name <package>/BuildInfo/GetPackageBuildInfo.cc is available in the package's ups source/ subdirectory, it will pluck out the build time and version given in that source file and add them to the run record. For a concrete example, here's what I have in mu2edeq01:/home/jcfree/run_records/2683, the record of a run performed with this new functionality. Besides what's already been mentioned, notice the addition of the comments for the user's benefit; this is automatically generated by DAQInterface:

```
# Two possible sets of fields provided below for code info, depending on if a git repo was available:
# <git commit hash> <LoCs added on top of commit> <LoCs removed on top of commit> <git commit comment> <BuildI
nfo build time (if available)> <BuildInfo version (if available)>
# <package version> <BuildInfo build time (if available)> <BuildInfo version (if available)>

DAQInterface commit/version: a3bbc0746c0985bc1eda7f5c6583da181cda0a42 0 0 "JCF: allow get_build_info to handle
multiple packa..." "time from BuildInfo undetermined" "version from BuildInfo undetermined"
artdaq commit/version: b81e90096c8eae5f58601f4756193fd61ceeda2e 0 0 "Pull out code for handling multiple sends
with the..." "09-Jul-2019 17:38:28 UTC" "v3_04_01-9-gb81e9009"
artdaq-core commit/version: v3_04_11 "06-Mar-2019 02:02:06 UTC" "v3_04_11"
artdaq-core-demo commit/version: 9312e98aeb941308fb52210be92a2a97e277da7c 0 0 "Fix to doxygen generation for p
artial MRB areas " "time from BuildInfo undetermined" "version from BuildInfo undetermined"
artdaq-demo commit/version: a4112709c79787cdd039e3ff58f9793395a6bbeb 2 3 "Merge branch release/v3_04_01 " "tim
e from BuildInfo undetermined" "version from BuildInfo undetermined"
artdaq-utilities commit/version: v1_04_10 "time from BuildInfo undetermined" "version from BuildInfo undetermi
ned"
```

This output can be compared to what I get if I run DAQInterface on its develop branch:

```
DAQInterface commit/version: 51c9b8587330f51f945b062c2f6ec24701261e89 "JCF: check for a file containing commit
info when ..."
artdaq commit/version: b81e90096c8eae5f58601f4756193fd61ceeda2e "Pull out code for handling multiple sends wit
h the..."
artdaq-core commit/version: v3_04_11
artdaq-core-demo commit/version: 9312e98aeb941308fb52210be92a2a97e277da7c "Fix to doxygen generation for parti
al MRB areas "
artdaq-demo commit/version: a4112709c79787cdd039e3ff58f9793395a6bbeb "Merge branch release/v3_04_01 "
artdaq-utilities commit/version: v1_04_10
```

#6 - 07/10/2019 03:09 PM - John Freeman

- % Done changed from 0 to 100

- Status changed from Work in progress to Resolved

After showing Kurt the functionality described in the post above, he recommended that I add the time of the commit so it can be compared to the build time. This has been done. With the head of feature/issue22777_more_accurate_record_of_code, 0de54ea99c8e9dbbfd1099cbfa42615a6fc44c1e, I'm marking the issue resolved.

#7 - 07/10/2019 03:41 PM - John Freeman

- Status changed from Resolved to Work in progress

Removing the "resolved" and reverting to "work in progress", since Ron just had a good suggestion: that if a package isn't installed as a ups product, it still may have BuildInfo values in the mrb area's build_slf7.x86_64 subdirectory.

#8 - 07/10/2019 05:14 PM - John Freeman

- Status changed from Work in progress to Resolved

Marking this issue as resolved again. With commit 532d45c0ae881472e259badc2fc7265dda2bf0dc, if DAQInterface sees that a ups product for the package in question isn't set up by the DAQ setup script, instead of giving up on finding a PackageBuildInfo.cc file, it'll look for the file in the local build subdirectory of the mrb area.

#9 - 07/17/2019 03:55 PM - Kurt Biery

I have verified the expected behavior in an artdaq-demo system on the mu2edeq cluster.

Here are the differences between the newer and older style, for one package:

```
< artdaq commit/version: 271f3fea4a6018c6d23fcabe2fb71df6f03471d6 4 4 "un-comment the optional packages in product_deps t..." "Wed Jul 17
14:12:19 2019 0500" "17-Jul-2019 20:17:14 UTC" "v2019_07_17_protodFQ_pduncHacks"
```

--

artdaq commit/version: df18c8cc1acb4cbb0a53acbf8b0553bf0e5a76ad "Merge remote-tracking branch origin/bugfix/22267_R..."

One thing that we'll probably want to remember is that the commit time includes Daylight Savings, whereas UTC does not.

#10 - 07/17/2019 05:50 PM - John Freeman

- *Status changed from Resolved to Reviewed*