

artdaq - Idea #6256

Separate the existing artdaq code into multiple packages

05/13/2014 12:54 PM - Kurt Biery

Status:	Closed	Start date:	05/13/2014
Priority:	Normal	Due date:	07/31/2014
Assignee:	Kurt Biery	% Done:	90%
Category:		Estimated time:	80.00 hours
Target version:	v1_09_00		
Experiment:	-		

Description

We believe that there could be advantages to separating the existing artdaq code into two or more software packages. It's not totally clear what the best separation would be, but there are a couple of use cases that illustrate the issues:

1. *art* and a couple of classes from *artdaq* are used for the NOvA data-driven trigger. The current organization of the code means that when NOvA folks want a new *art* release, a new *artdaq* release needs to be cut and released in order for the new version of *art* to be made available to the NOvA DDT. If instead there were a separate package that had the input source (or whatever the DDT needs from *artdaq*), and this new package depended on *art*, but not *artdaq*, then the artificial strong coupling between the NOvA DDT and *artdaq* would be broken.
2. The uBooNE DAQ uses a couple of utility classes from *artdaq*, but it does not use *art* or ROOT. It would be nice to be able to use the needed classes from *artdaq* in the uBooNE DAQ code without bringing in a number of unnecessary heavy-weight packages.

History

#1 - 05/21/2014 03:41 PM - Kurt Biery

Some background information on the use of *artdaq* in the NOvA DDT...

Primary source files:

- <https://cdcvns.fnal.gov/redmine/projects/novaddt/repository/show/trunk/DDTproto>
- [novadaq-ctrl-master.fnal.gov:/nova/novaddt/novaddt/releases/development/DDTproto](https://cdcvns.fnal.gov/redmine/projects/novaddt/repository/show/trunk/DDTproto)

Files that are used from *artdaq*:

```
[novadaq@novadaq-ctrl-master DDTproto]$ findc artdaq | grep include | cut -d':' -f2 | sort -u
#include "artdaq/DAQdata/Debug.hh"
#include "artdaq/DAQdata/FragmentGenerator.hh"
#include "artdaq/DAQdata/Fragment.hh"
#include "artdaq/DAQdata/Fragments.hh"
#include "artdaq/DAQdata/GeneratorMacros.hh"
#include "artdaq/DAQdata/makeFragmentGenerator.hh"
#include "artdaq/DAQdata/RawEvent.hh"
#include "artdaq/DAQrate/GlobalQueue.hh"
include SoftRelTools/arch_spec_artdaq.mk
```

#2 - 06/12/2014 02:53 PM - Kurt Biery

- Status changed from New to Assigned

- Assignee set to Kurt Biery

- % Done changed from 0 to 10

I've started to summarize some ideas on how to proceed [here](#).

I've spent ~9 hours so far on this task through 14:45 today.

#3 - 07/28/2014 10:11 AM - Kurt Biery

- Target version changed from 576 to v1_09_00

#4 - 07/28/2014 11:14 AM - Kurt Biery

This is currently implemented as a separation into artdaq-core (which depends on art) and artdaq (which depends on artdaq-core, MPI, xmlrpc, etc.)

#5 - 07/28/2014 11:23 AM - Kurt Biery

The new *artdaq-core* repo is a secondary one in the *artdaq* Redmine project.

Here is an email that I sent on this subject:

After thinking about this a little more, I decided to split off only one package: *artdaq-core*. If we decide that we need multiple packages underneath *artdaq* later, we can create them then (with sufficient knowledge of why they are needed.)

On 7/21/2014 4:55 PM, Kurt Biery wrote:

Hi guys,

As we've discussed, we're splitting *artdaq* into *artdaq*, *artdaq-core*, and *artdaq-generators*. Those are git repository names. UPS products can't have hyphens (from what I can tell). This means that the associated UPS products will be *artdaq*, *artdaq_core*, and *artdaq_generators*.

When user code includes header files from these packages, that will look like `'#include "artdaq-core/Data/Fragments.hh"`.

So far, I've kept the "artdaq" namespace for all of the code in these three packages, and my sense is that we should use that model elsewhere. (E.g. namespace="lbne" for both *lbne-raw-data* and *lbne-artdaq*).

This model (hyphens in repos and underscores in UPS product names) seems to be used elsewhere. I just wanted to mention it in case any of you wanted to talk about this more before we get too far down this path.

Kurt

#6 - 07/28/2014 11:24 AM - Kurt Biery

- *Due date set to 07/31/2014*

- *Status changed from Assigned to Resolved*

- *% Done changed from 10 to 90*

I calculate that I spent another ~32 hours on this task. This brings the total effort up to 41 hours.

#7 - 07/28/2014 11:29 AM - Kurt Biery

- *Status changed from Resolved to Closed*