artdaq - Idea #20815

Include RawEventHeader as art product

09/12/2018 01:31 PM - Eric Flumerfelt

Status:	Resolved	**Start date:**	09/12/2018
Priority:	Normal	**Due date:**	
Assignee:	Eric Flumerfelt	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
Experiment:	-		

Description

The RawEventHeader is used by the art input source to set some of the art event fields. However, there is some information in there that may not be available elsewhere. We should make sure that the RawEventHeader is added to the art Dictionaries and added to each event in the input source.

Related issues:

Related to artdaq - Feature #5958: Add the ability to flag various conditions...	**Closed**	**12/18/2012**

History

#1 - 09/13/2018 11:58 AM - Eric Flumerfelt

- Status changed from New to Resolved

- Assignee set to Eric Flumerfelt

There are two parts to this implementation: Adding the RawEventHeader to the ROOT dictionary, and putting the header into the art event in the input source.

1st part: artdaq_core:feature/RawEvent_HeaderInArtDictionary
2nd part: artdaq:feature/SharedMemoryReader_RawEventHeaderInArtEvent

Code has been tested using artdaqDriver and the version of EventDump_module found in the feature/SharedMemoryReader_RawEventHeaderInArtEvent branch.

#2 - 01/07/2019 12:56 PM - Kurt Biery

Eric,
This functionality is looking good, based on an initial couple of tests.

A few questions came up as I looked at things:

1. I noticed that the following pair of lines shows up twice in the SharedMemoryReader constructor. Should/can the second pair be removed?

```
                                incoming_events.reset(new SharedMemoryEventReceiver(ps.get<uint32_t>("shar
ed_memory_key", 0xBEE70000 + getppid()), ps.get<uint32_t>\
("broadcast_shared_memory_key", 0xCEE70000 + getppid())));
                                my_rank = incoming_events->GetRank();
```

2. In the event dump, the RawEventHeader fragment count and the word count are typically listed as zero, as shown below. I wonder if this will always** be true, and therefore we should suppress those fields from the event dump output. (** since the determination of the fragment count and word count are determined from the "fragments_" data member in RawEvent, and since it is only the RawEventHeader that is stored in the art/ROOT event, it seems like we will never be able to display these values.)

```
***** Start of EventDump for event 427 *****
Event Header: Run 237, Subrun 1, Event 427, SeqID 427, FragCount 0, WordCount 0, Complete? 1
```

3. I ran a test in which one of the fragments in an event had its 'missing data' flag set to true. This didn't get reflected in the RawEvent Complete? flag (which still showed "1"). This could be because RawEvent::Complete? is intended for a different purpose, or because we haven't yet added the code to properly set the RawEvent::Complete? flag when a fragment has some missing data. Do you know whether either of these might be true?

Thanks,
Kurt

#3 - 01/07/2019 03:36 PM - Eric Flumerfelt

1. Looks like a merge issue, develop only has it once
2. That looks to be correct. However, this is a change that would either have to be made in artdaq_core (in RawEvent.cc), or by adding a full member-wise dump in EventDump_module (and remembering anywhere else that we print out the RawEvent that the fragments_ vector may be empty).
3. I believe that the RawEvent::isComplete() flag is intended to indicate whether all expected Fragments are present. I agree that this could be misleading, and we should carefully thing about the language we want to use. I would vote for adding another field to RawEventHeader so we can say "Complete, missing data" or "Incomplete, not missing data", etc.

#4 - 01/09/2019 01:57 PM - Kurt Biery

1. I removed the extra lines from SharedMemoryReader.
2. It seemed like creating an operator<< function for RawEventHeader and using that in the EventDump_module was another option. I've committed some code along these lines. Please take a look and see if that makes sense.
3. I agree with (what I believe to be) the sentiment of what you are saying. That is, maybe we leave RawEventHeader::is_complete as providing feedback on whether the correct number of fragments are present, and we add new data members over time to indicate other conditions, such as whether one of those fragments is an Empty fragment or one of those fragments has some sort of internal inconsistency.

#5 - 10/18/2019 02:50 PM - Eric Flumerfelt

- Related to Feature #5958: Add the ability to flag various conditions in a RawEvent added