ds50daq - Bug #4084

Runs with no events cause the EventBuilders to crash

06/12/2013 04:09 PM - Kurt Biery

Status:	Assigned	**Start date:**	06/12/2013
Priority:	Normal	**Due date:**	07/19/2013
Assignee:	Michael Wang	**% Done:**	0%
Category:		**Estimated time:**	12.00 hours
Target version:			

Description

In testing of the latest ds50daq code at LNGS, I find that if I start a run, no triggers are received, and then I end the run, the EventBuilders all crash.

Here is the tail end of the PMT log:

```
Wed Jun 12 23:06:54 +0200 2013: 12-Jun-2013 23:06:54 CEST MF-online
Wed Jun 12 23:06:54 +0200 2013: Done flushing stale events from the EventStore.
Wed Jun 12 23:06:54 +0200 2013: %MSG
Wed Jun 12 23:06:54 +0200 2013: %MSG-d CommandableInterface:  EventBuilder-5451 12-Jun-2013 23:06:54 CEST
MF-online
Wed Jun 12 23:06:54 +0200 2013: InRunExit called.
Wed Jun 12 23:06:54 +0200 2013: %MSG
Wed Jun 12 23:06:54 +0200 2013: %MSG-d CommandableInterface:  EventBuilder-5451 12-Jun-2013 23:06:54 CEST
MF-online
Wed Jun 12 23:06:54 +0200 2013: States before and after a stop transition: InRunMap::Running and InitializedMap::Ready
Wed Jun 12 23:06:54 +0200 2013: %MSG
Wed Jun 12 23:06:54 +0200 2013: %MSG-d CommandableInterface:  EventBuilder-5452 12-Jun-2013 23:06:54 CEST
MF-online
Wed Jun 12 23:06:54 +0200 2013: InRunExit called.
Wed Jun 12 23:06:54 +0200 2013: %MSG
Wed Jun 12 23:06:54 +0200 2013: %MSG-d CommandableInterface:  EventBuilder-5452 12-Jun-2013 23:06:54 CEST
MF-online
Wed Jun 12 23:06:54 +0200 2013: States before and after a stop transition: InRunMap::Running and InitializedMap::Ready
Wed Jun 12 23:06:54 +0200 2013: %MSG
Wed Jun 12 23:06:54 +0200 2013: %MSG-d CommandableInterface:  EventBuilder-5453 12-Jun-2013 23:06:54 CEST
MF-online
Wed Jun 12 23:06:54 +0200 2013: InRunExit called.
Wed Jun 12 23:06:54 +0200 2013: %MSG
Wed Jun 12 23:06:54 +0200 2013: %MSG-d CommandableInterface:  EventBuilder-5453 12-Jun-2013 23:06:54 CEST
MF-online
Wed Jun 12 23:06:54 +0200 2013: States before and after a stop transition: InRunMap::Running and InitializedMap::Ready
Wed Jun 12 23:06:54 +0200 2013: %MSG
Wed Jun 12 23:06:54 +0200 2013: EXITING:dseb1:139:eventbuilder:5450
eventbuilder on dseb1 is exiting.
Wed Jun 12 23:06:54 +0200 2013: /home/daq/system-Aggregator-Kurt/profile/bin/mpi_wrapper.sh: line 6: 51114 Segmentation
fault      $1 -p ${:2}
Wed Jun 12 23:06:54 +0200 2013: EXITING:dseb2:139:eventbuilder:5451
eventbuilder on dseb2 is exiting.
Wed Jun 12 23:06:54 +0200 2013: /home/daq/system-Aggregator-Kurt/profile/bin/mpi_wrapper.sh: line 6: 22529 Segmentation
fault      $1 -p ${:2}
Wed Jun 12 23:06:54 +0200 2013: EXITING:dseb4:139:eventbuilder:5452
eventbuilder on dseb4 is exiting.
Wed Jun 12 23:06:54 +0200 2013: /home/daq/system-Aggregator-Kurt/profile/bin/mpi_wrapper.sh: line 6: 25899 Segmentation
fault      $1 -p ${:2}
Wed Jun 12 23:06:54 +0200 2013: EXITING:dseb5:139:eventbuilder:5453
eventbuilder on dseb5 is exiting.
Wed Jun 12 23:06:54 +0200 2013: /home/daq/system-Aggregator-Kurt/profile/bin/mpi_wrapper.sh: line 6: 25622 Segmentation
fault      $1 -p ${:2}
^CCleaning up.  Please wait for PMT to exit...
[2013-06-12 23:07:06] INFO  going to shutdown ...
[2013-06-12 23:07:06] INFO  WEBrick::HTTPServer#start done.
```

History

#1 - 06/12/2013 04:11 PM - Kurt Biery

The command that I'm using to configure the system in such a way that it generates no triggers is the following:

```
manageCommPhase1System.sh -g 400 -t 18 -l 0 initConfiguration
```

#2 - 06/12/2013 04:13 PM - Kurt Biery

Here is a stack trace:

```
Program received signal SIGSEGV, Segmentation fault.
[Switching to Thread 0x7f0c37c3f700 (LWP 24152)]
artdaq::detail::RawEventQueueReader::readNext (this=0x7f0c38fbf040, inR=@0x7f0c37c3cb00, inSR=@0x7f0c37c3cb08, outR=@0x7f0c37c3cad0,
outSR=@0x7f0c37c3cae0, outE=@0x7f0c37c3caf0) at /home/daq/artdaq/artdaq/artdaq/ArtModules/detail/RawEventQueueReader.cc:121
121         art::EventID const evid(art::EventID::flushEvent(inR->id()));
Missing separate debuginfos, use: debuginfo-install libgcc-4.4.6-4.el6.x86_64
(gdb) where
#0  artdaq::detail::RawEventQueueReader::readNext (this=0x7f0c38fbf040, inR=@0x7f0c37c3cb00, inSR=@0x7f0c37c3cb08,
outR=@0x7f0c37c3cad0, outSR=@0x7f0c37c3cae0, outE=@0x7f0c37c3caf0) at
/home/daq/artdaq/artdaq/artdaq/ArtModules/detail/RawEventQueueReader.cc:121
#1  0x00007f0c33ffd610 in art::Source<artdaq::detail::RawEventQueueReader>::readNext_ (this=0x7f0c38fbefc0) at
/products/art/v1_07_01/include/art/Framework/IO/Sources/Source.h:404
#2  0x00007f0c33ffd8af in art::Source<artdaq::detail::RawEventQueueReader>::readNextAndRequireRun_ (this=0x7f0c38fbefc0) at
/products/art/v1_07_01/include/art/Framework/IO/Sources/Source.h:507
#3  0x00007f0c33ffdb6a in art::Source<artdaq::detail::RawEventQueueReader>::nextItemType (this=0x7f0c38fbefc0) at
/products/art/v1_07_01/include/art/Framework/IO/Sources/Source.h:478
#4  0x00007f0c43a5b25d in art::EventProcessor::runCommon_ (this=0x7f0c38fb2500) at
/home/garren/shared-scratch/p/art_suite/v1_07_01/source/art/art/Framework/Core/EventProcessor.cc:395
#5  0x00007f0c43a5c10b in art::EventProcessor::runToCompletion (this=0x7f0c38fb2500) at
/home/garren/shared-scratch/p/art_suite/v1_07_01/source/art/art/Framework/Core/EventProcessor.cc:357
#6  0x00007f0c4c067eb0 in art::run_art_common_ (main_pset=...) at
/home/garren/shared-scratch/p/art_suite/v1_07_01/source/art/art/Framework/Art/run_art.cc:280
#7  0x00007f0c4c068436 in art::run_art_string_config (config_string=...) at
/home/garren/shared-scratch/p/art_suite/v1_07_01/source/art/art/Framework/Art/run_art.cc:203
#8  0x00007f0c4da6f2bf in operator() (this=0x1e334d0) at
/products/gcc/v4_7_1/Linux64bit+2.6-2.12/bin/../lib/gcc/x86_64-unknown-linux-gnu/4.7.1/../../../../include/c++/4.7.1/functional:2311
#9  std::__future_base::_Task_setter<std::unique_ptr<std::__future_base::_Result<int>, std::__future_base::_Result_base::_Deleter>, int>::operator()
(this=0x1e334d0) at /products/gcc/v4_7_1/Linux64bit+2.6-2.12/bin/../lib/gcc/x86_64-unknown-linux-gnu/4.7.1/../../../../include/c++/4.7.1/future:1217
#10 0x00007f0c4da6f368 in std::_Function_handler<std::unique_ptr<std::__future_base::_Result_base,
std::__future_base::_Result_base::_Deleter>(), std::__future_base::_Task_setter<std::unique_ptr<std::__future_base::_Result<int>,
std::__future_base::_Result_base::_Deleter>, int> >::_M_invoke(const std::_Any_data &) (_functor=Unhandled dwarf expression opcode 0xf3
) at /products/gcc/v4_7_1/Linux64bit+2.6-2.12/bin/../lib/gcc/x86_64-unknown-linux-gnu/4.7.1/../../../../include/c++/4.7.1/functional:1912
#11 0x00007f0c4dccb10b in operator() (this=0xc16438, __f=Unhandled dwarf expression opcode 0xf3
) at /products/gcc/v4_7_1/Linux64bit+2.6-2.12/bin/../lib/gcc/x86_64-unknown-linux-gnu/4.7.1/../../../../include/c++/4.7.1/functional:2311
#12 std::_future_base::_State_base::_M_do_set(std::function<std::unique_ptr<std::__future_base::_Result_base,
std::__future_base::_Result_base::_Deleter>()> &, bool &) (this=0xc16438, _f=Unhandled dwarf expression opcode 0xf3
) at /products/gcc/v4_7_1/Linux64bit+2.6-2.12/bin/../lib/gcc/x86_64-unknown-linux-gnu/4.7.1/../../../../include/c++/4.7.1/future:473
#13 0x00007f0c4d7e79b9 in __once_proxy () from /products/fhiclcpp/v2_17_02/slf6.x86_64.e2.prof/lib/libfhiclcpp.so
#14 0x000000357f00cb23 in pthread_once () from /lib64/libpthread.so.0
#15 0x00007f0c4da6cdfd in __gthread_once (this=0xc16438, __res=Unhandled dwarf expression opcode 0xf3
) at
/products/gcc/v4_7_1/Linux64bit+2.6-2.12/bin/../lib/gcc/x86_64-unknown-linux-gnu/4.7.1/../../../../include/c++/4.7.1/x86_64-unknown-linux-gnu/bits/gth
r-default.h:718
#16 call_once<void (std::_future_base::_State_base::*)(std::function<std::unique_ptr<std::__future_base::_Result_base,
std::__future_base::_Result_base::_Deleter>()>&, bool&), std::__future_base::_State_base* const,
std::reference_wrapper<std::function<std::unique_ptr<std::__future_base::_Result_base, std::__future_base::_Result_base::_Deleter>()> >,
std::reference_wrapper<bool> > (this=0xc16438, _res=Unhandled dwarf expression opcode 0xf3
) at /products/gcc/v4_7_1/Linux64bit+2.6-2.12/bin/../lib/gcc/x86_64-unknown-linux-gnu/4.7.1/../../../../include/c++/4.7.1/mutex:819
#17 std::_future_base::_State_base::_M_set_result(std::function<std::unique_ptr<std::__future_base::_Result_base,
std::__future_base::_Result_base::_Deleter>()>, bool) (this=0xc16438, _res=Unhandled dwarf expression opcode 0xf3
) at /products/gcc/v4_7_1/Linux64bit+2.6-2.12/bin/../lib/gcc/x86_64-unknown-linux-gnu/4.7.1/../../../../include/c++/4.7.1/future:362
#18 0x00007f0c4da6f0f9 in std::_future_base::_Async_state_impl<std::_Bind_simple<int (()(std::basic_string<char, std::char_traits<char>,
std::allocator<char> >))(std::basic_string<char, std::char_traits<char>, std::allocator<char> > const&)>,
int>::_Async_state_impl(std::_Bind_simple<int (()(std::basic_string<char, std::char_traits<char>, std::allocator<char> >))(std::basic_string<char,
std::char_traits<char>, std::allocator<char> > const&)>&&)::{lambda()#1}::operator()() const () at
/products/gcc/v4_7_1/Linux64bit+2.6-2.12/bin/../lib/gcc/x86_64-unknown-linux-gnu/4.7.1/../../../../include/c++/4.7.1/future:1454
#19 0x00007f0c4d7e835c in execute_native_thread_routine () from /products/fhiclcpp/v2_17_02/slf6.x86_64.e2.prof/lib/libfhiclcpp.so
#20 0x000000357f007851 in start_thread () from /lib64/libpthread.so.0
#21 0x000000357e8e811d in clone () from /lib64/libc.so.6
```

(gdb)

#3 - 07/11/2013 11:59 AM - Kurt Biery

- Due date changed from 06/13/2013 to 07/19/2013

- Assignee changed from Kurt Biery to Michael Wang

- Estimated time changed from 4.00 h to 12.00 h

#4 - 07/19/2013 09:46 AM - Michael Wang

This may be a separate issue but I have observed these CAEN bus errors when I try to reproduce the no events EB crash:

Wed Jul 17 16:53:16 ~~0500 2013: %MSG i InputFailure: PostOpenFile BeforeEvents~~
~~Wed Jul 17 16:53:16 0500 2013: Reading timed out in RawEventQueueReader::readNext()~~
~~Wed Jul 17 16:53:16 -0500 2013: %MSG~~
~~Wed Jul 17 16:53:32 -0500 2013: %MSG-e XMLRPC_Commander: BoardReader-6660 MF-online~~
~~Wed Jul 17 16:53:32 0500 2013: Exception when trying to stop the run: ---~~ caen_io BEGIN
Wed Jul 17 16:53:32 ~~0500 2013: CAENVME_ReadCycle(0,0x1001018): cvBusError~~
~~Wed Jul 17 16:53:32 -0500 2013: ---~~ caen_io END
Wed Jul 17 16:53:32 -0500 2013: %MSG
Wed Jul 17 16:53:32 -0500 2013: %MSG-w V1495: BoardReader-6660 MF-online
Wed Jul 17 16:53:32 -0500 2013: no data to read at event 1, read size = 0
Wed Jul 17 16:53:32 -0500 2013: %MSG
Wed Jul 17 16:53:33 -0500 2013: EXITING:dseb8:139:eventbuilder:6670
Wed Jul 17 16:53:33 -0500 2013: /home/mwang/ds50daqBuild/bin/mpi_wrapper.sh: line 6: 38112 Segmentation fault $1 -p ${@:2}
Wed Jul 17 16:53:33 -0500 2013: %MSG-i Aggregator: Aggregator-6680 MF-online
Wed Jul 17 16:53:33 -0500 2013: Stopping run 101 after 0 events.
Wed Jul 17 16:53:33 -0500 2013: %MSG
Wed Jul 17 16:53:54 -0500 2013: %MSG-e Aggregator: Aggregator-6680 MF-online
Wed Jul 17 16:53:54 -0500 2013: Timeout receiving fragments after stop, but no endSubRun message available to send to art.
Wed Jul 17 16:53:54 -0500 2013: %MSG

Mike

#5 - 08/27/2013 10:03 AM - Michael Wang

We have looked extensively at this issue.

Initially, we found that the immediate issue causing the segmentation fault was an uninitialized pointer in RawEventQueReader.cc in artdaq. This was causing problems when the code detected a single fragment event that signified an end-of-subrun and tried to create a flushEvent using the pointer to pass the run number. Fixing this prevented the segmentation fault but other issues arose.

We now found out that two empty subruns next to each other would create two end-of-subrun flushEvents with identical Run numbers and "flush" subruns. But since RawEvenQueReader would return a new subrun principal signifying a new subrun, art would complain that the new subrun had the same number as the previous one. The problem was also no longer confined to the EventBuilder, complications also caused the Aggregator to crash.

The solution we came up with is detailed in the following message sent to Chris Green:

> Hi Chris,
>
> In art/Persistency/Provenance/EventID.h, we currently have:
>
> static EventID flushEvent();
> static EventID flushEvent(RunID rID);
>
> Would it be possible to add the following:
>
> static EventID flushEvent(RunID rID, SubRunID srID);
>
> I need this in order to solve the issue we have with the ds50daq aggregator crashing when there are runs/subruns with no events. I talked to you recently about this.
>
> When an end-of-subrun fragment is received, the RawEventQueReader uses flushEvent(RunID rID) to create a new event with a valid Run, flush Subrun, and flush Event. If there are no events in the subrun, RawEventQueReader will return new Run and SubRun principals. Because the SubRun is "flush", a new SubRun is never started and after the end-of-run fragment is received, no subrun is ever ended. Since no subrun is ended, the NetMonOutputModule never calls writeSubRun to propagate the end-of-subrun message from the eventbuilder to the aggregator side. This causes problems for the aggregator which checks for an end-of-subrun message when it receives a stop command.
>
> So my solution to this is to create a flushEvent with a valid Run AND SubRun but having a "flush" event number. Even in the absence of events, at the end of a subrun, such an event, because it has a valid subrun, will trigger a beginSubrun and a subsequent event with a different subrun (whether and end-of-run flush event or an actual data event) will trigger an endSubRun and therefore a writeSubRun that propagates the end-of-subrun message from the eventbuilder to the aggregator which causes subruns to terminate properly.

In the case when there are events, I let RawEventQueReader do things as usual using the flushEvent(RunID rID) form, i.e. upon the receipt of an end-of-subrun fragment, it generates a flush event with a valid Run , flush Subrun, and flush event. This is fine since art is already in a subrun started when the data events were received and a transition from a valid subrun in the data event to the flush Subrun in an end-of-subrun event triggers a endSubRun.

I have tested my proposed solution to the "no events" issue and verified that everything works properly with any sequence of start/stop/pause/resume commands whether events are present or not. Files are also created for each subrun even if they are empty.

Mike

We have tested the above proposal in a feature branch of art. Chris is looking at this and we expect this to be incorporated in an updated version of art.

Mike

#6 - 08/27/2013 10:17 AM - Michael Wang

- File debug-noevts.pdf added

I am attaching a pdf file with debug messages I inserted into the code when I was trying to understand how the art state machine worked and how the new run, subrun, and event principals returned by RawEventQueReader triggered various tasks in the output modules. Since messages from different parts of the code are interleaved and can be very confusing to understand, I have color coded the text using a color to represent related messages. I found this color coded output useful in educating myself about art behavior and include it here for future reference.

Files

debug-noevts.pdf	72.1 KB	08/27/2013	Michael Wang