

art - Bug #13820

Event display exits when reloading.

09/08/2016 09:45 PM - Tingjun Yang

Status:	Closed	Start date:	09/08/2016
Priority:	Normal	Due date:	
Assignee:	Kyle Knoepfel	% Done:	100%
Category:	Event Loop	Estimated time:	0.00 hour
Target version:	2.04.01	Spent time:	12.50 hours
Occurs In:		Experiment:	LArSoft
Scope:		SSI Package:	art

Description

Dear LArSoft Experts,

I am having a strange problem with event display. If I reconstruct one event, look at it in event display and hit reload button, the event display would exit.

Here is how to reproduce this problem on a dunegpvm machine.

```
source /grid/fermiapp/products/dune/setup_dune.sh
setup dunetpc v06_05_00 -q e10:prof
lar -c standard_reco_dune10kt_1x2x6.fcl /pnfs/dune/scratch/dunepro/v06_04_01/detsim/prod_muminus_0.1-5.0GeV_isotropic_dune10kt_1x2x6/13289364_0/prod_muminus_0.1-5.0GeV_isotropic_dune10kt_1x2x6_7_20160826T215719_gen_6b914fe0-09c3-469e-b1f0-056dad368alb_g4_detsim.root -n 1
lar -c evd_dunefd.fcl prod_muminus_0.1-5.0GeV_isotropic_dune10kt_1x2x6_7_20160826T215719_gen_6b914fe0-09c3-469e-b1f0-056dad368alb_g4_detsim_reco.root
```

click Reload, event display will exit.

However, if I run event display on the file with all reconstructed events (from production):

```
lar -c evd_dunefd.fcl /pnfs/dune/scratch/dunepro/v06_04_01/reco/prod_muminus_0.1-5.0GeV_isotropic_dune10kt_1x2x6/10111991_0/prod_muminus_0.1-5.0GeV_isotropic_dune10kt_1x2x6_44_20160826T215741_gen_d40d2d5c-caa1-4a3f-b716-4e62843057dd_g4_detsim_reco.root
```

Clicking Reload does not exit.

I am having the same problem with argoneut as well. This is annoying because every time I change event display configuration (e.g. track module label), the display would refresh and quite. I suspect this is an art problem and I started to seeing this problem since moving to art2 (larsoft v06).

Thanks,
Tingjun

History

#1 - 09/26/2016 10:09 AM - Lynn Garren

- Status changed from New to Assigned
- Assignee set to Gianluca Petrillo

#2 - 09/26/2016 05:07 PM - Gianluca Petrillo

- Description updated
- Occurs In v06_03_00 added
- Occurs In deleted (with_root6)
- Experiment LArSoft added
- Experiment deleted (DUNE)

I can reproduce the problem.
Bisection shows that the problem was introduced between LArSoft v06_02_00 and v06_03_00.

LArSoft event display relies on *nutools* for driving the event loop, via EventDisplay service. When the action Tingjun reports is performed, the service attempts a RootInput::seekToEvent() call with the same event as the current one.

The differences in dependencies are:

larsoft	v06_02_00	v06_03_00
nutools	v2_01_03	v2_03_00
art	v2_00_03	v2_02_02

Differences in lareventdisplay and in *nutools* pertaining event display are naught. There are quite some changes in art::RootInput though. Bouncing the ball to *art* experts.

#3 - 09/26/2016 05:11 PM - Gianluca Petrillo

Simplified steps to reproduce:

```
setup larsoft v06_03_00 -q e10:prof
lar -c prodsingle.fcl -n 1
lar -c evd.fcl single_gen.root
```

and then follow Tingjun instructions (either hit [Reload] button or edit a parameter, e.g. menu Edit | Configure Drawing | RawDrawinOptions, click on the "RawDataLabel" box that says daq, hit <Enter> and then click on [Apply] button).

#4 - 09/26/2016 05:25 PM - Lynn Garren

- Project changed from LArSoft to art
- Category deleted (Event Display)
- Status changed from Assigned to Accepted
- Assignee deleted (Gianluca Petrillo)
- Target version deleted (develop/HEAD)

#5 - 09/26/2016 08:12 PM - Kyle Knoepfel

- Status changed from Accepted to Assigned
- Assignee set to Kyle Knoepfel

#6 - 09/28/2016 02:47 PM - Kyle Knoepfel

The problem is understood.

[N.B. The following explanation is technical.]

In the context of the EventDisplay service, the seekToEvent function ultimately controls which state-machine events are emitted. In art [2.01.00](#) and later, event-processing (and calls to all postProcessEvent callbacks) occur as an exit action from the HandleEvents state (which, in the boost::statechart implementation, is a call to the state's destructor). At the point when the destructor is called, the next signal has already been emitted and posted. If there is only one event in the file, this means that by the time the user has a chance to (e.g.) press the Reload button on the EventDisplay window, the Stop signal has already been registered as the next state-machine event. Any attempt at adjusting the signal, by calling seekToEvent, will be too late.

There are several considerations in pursuing a solution:

1. Moving the nutools EventDisplay postProcessEvent functionality to become part of the preProcessEvent callback is a possibility, but it is not obvious to me what the side-effects would be.
2. Putting the event-processing step back in the NewEvent state will result in processing (but not writing) one-too-many events per output-file switch, if the new output-file switching is enabled.
3. Fundamentally, the state machine should probably be adjusted, like adding a ProcessEvent stage. However, it is not obvious to me to what extent that can be done cleanly. More analysis is required at this stage.

#7 - 09/28/2016 03:06 PM - Kyle Knoepfel

I had a thinko when mentioning option 1 above--of course, the preProcessEvent callbacks are also called in the HandleEvents destructor. So it seems that we're constrained to adjusting the state machine somehow.

#8 - 09/29/2016 10:16 AM - Kyle Knoepfel

- Category set to Event Loop

- % Done changed from 0 to 60

- SSI Package art added

- SSI Package deleted ()

The problem has been solved: by adding a ProcessEvent state to the state machine, I have been able to move the event-processing outside of the HandleEvents destructor. This allows the seekToEvent function call to guide the emission of the correct state-machine event before the nextItemType() function is called in the EventProcessor::runToCompletion event loop.

Incidentally, by adding the ProcessEvent state as a sub-state of the HandleEvents composite state, I have restored the symmetry of the state machine with respect to Runs and SubRuns.

What remains: some code cleanup, adjusted documentation of the state machine, and a suitable regression/integration test within art.

#9 - 09/30/2016 02:32 PM - Kyle Knoepfel

- Status changed from Assigned to Resolved

- % Done changed from 60 to 100

Code polished, and a test added to art that will catch this problem in the future. I have verified that the event display works correctly with the pushed changes.

Implemented with commit [art:7e7e8d2f](#).

#10 - 10/10/2016 10:46 AM - Kyle Knoepfel

- Status changed from Resolved to Closed

- Target version set to 2.04.01