LArSoft - Bug #11659

Investigate the failure in support of pandora visualization within LArSoft

02/05/2016 07:02 PM - Gianluca Petrillo

Status:	Closed	**Start date:**	02/05/2016
Priority:	Low	**Due date:**	
Assignee:	Gianluca Petrillo	**% Done:**	100%
Category:	External Packages	**Estimated time:**	0.00 hour
Target version:		**Spent time:**	4.50 hours
Occurs In:		**Co-Assignees:**	
Experiment:	LArSoft		

Description

John Marshall has mentioned that there have been reports of Pandora visualization not working under LArSoft.

From the description, it might be some race for graphic resources between different instances of ROOT TApplication... or not.

I haven't attempted to reproduce the problem, not I yet know how to try to.

History

#1 - 03/01/2016 11:07 AM - Gianluca Petrillo

- Status changed from Assigned to Feedback

Could any of you experts give a simple example to reproduce the problem?
We are not very versed in Pandorian yet.

#2 - 03/01/2016 03:09 PM - John Marshall

This would appear to be a TApplication issue. Something seems to, either implicitly or explicitly, have created a "default" TApplication (which has become the gApplication), before the request from PandoraMonitoring.

If it helps, we know that we have been able to use this ROOT functionality in the MARLIN linear collider framework (no issues), ATHENA for ATLAS (use a special interactive mode they have) and CMS-SW. It, of course, works without issue via a simple command-line application with full control of the TApplication.

-Approach: make local modification to PandoraSettings file and ensure that it is picked-up during MicroBooNE reco stage 1.

```
source /cvmfs/uboone.opensciencegrid.org/products/setup_uboone.sh

setup uboonecode v05_01_01 -q e9:prof

cd $MY_ARBITRARY_PATH
export FW_SEARCH_PATH=`pwd`:$FW_SEARCH_PATH

cp /cvmfs/fermilab.opensciencegrid.org/products/larsoft/larpandora/v05_00_03/scripts/PandoraSettings_MicroBooNE_Cosmic.xml ./MyPandoraSettings_MicroBooNE_Cosmic.xml
cp /cvmfs/uboone.opensciencegrid.org/products/uboonecode/v05_01_01/job/reco_uboone_stage_1.fcl ./myreco_uboone_stage_1.fcl

diff MyPandoraSettings_MicroBooNE_Cosmic.xml /cvmfs/fermilab.opensciencegrid.org/products/larsoft/larpandora/v05_00_03/scripts/PandoraSettings_MicroBooNE_Cosmic.xml

5c5
<     <IsMonitoringEnabled>true</IsMonitoringEnabled>
---
>     <IsMonitoringEnabled>false</IsMonitoringEnabled>

diff myreco_uboone_stage_1.fcl /cvmfs/uboone.opensciencegrid.org/products/uboonecode/v05_01_01/job/reco_uboone_stage_1.fcl

178c178
< physics.producers.pandoraCosmic.ConfigFile:                            "MyPandoraSettings_MicroBooNE_Cosmic.xml"
---
```

```
> physics.producers.pandoraCosmic.ConfigFile:                      "PandoraSettings_MicroBooNE_Cosmic.xml"
```

```
lar -c prod_piminus_0.1-2.0GeV_isotropic_uboone.fcl -n 1 # Somewhat arbitrary choice
lar -c standard_g4_uboone.fcl prod_piminus_0.1-2.0GeV_isotropic_uboone_20160301T181806_gen.root
lar -c standard_detsim_uboone.fcl prod_piminus_0.1-2.0GeV_isotropic_uboone_20160301T181806_gen_20160301T181832
_g4.root
lar -c myreco_uboone_stage_1.fcl prod_piminus_0.1-2.0GeV_isotropic_uboone_20160301T181806_gen_20160301T181832_
g4_20160301T181918_detsim.root
```

-Resulting screen output

```
<snip>
TimeModule> run: 1 subRun: 0 event: 1 cctrack CCTrackMaker 0.00292397
PandoraMonitoring, only able to use default TApplication (limited functionality).
PandoraMonitoring::InitializeEve(): DISPLAY environment set to :1044.0
PandoraMonitoring::InitializeEve(): Caught TEveException: TEveManager::Create ROOT is running in batch mode.
PandoraMonitoring::InitializeEve(): Attempt to release ROOT from batch mode.
PandoraMonitoring::InitializeEve(): Caught TEveException: TEveManager::Create window system not initialized.
Failure in algorithm 0x4f6d690, LArVisualMonitoring, unrecognized exception
PandoraMonitoring::InitializeEve(): DISPLAY environment set to :1044.0
PandoraMonitoring::InitializeEve(): Caught TEveException: TEveManager::Create window system not initialized.
PandoraMonitoring::InitializeEve(): Attempt to release ROOT from batch mode.
PandoraMonitoring::InitializeEve(): Caught TEveException: TEveManager::Create window system not initialized.
Failure in algorithm 0x4f910d0, LArVisualMonitoring, unrecognized exception
PandoraMonitoring::InitializeEve(): DISPLAY environment set to :1044.0
PandoraMonitoring::InitializeEve(): Caught TEveException: TEveManager::Create window system not initialized.
PandoraMonitoring::InitializeEve(): Attempt to release ROOT from batch mode.
PandoraMonitoring::InitializeEve(): Caught TEveException: TEveManager::Create window system not initialized.
Failure in algorithm 0x4f91300, LArVisualMonitoring, unrecognized exception
PandoraMonitoring::InitializeEve(): DISPLAY environment set to :1044.0
PandoraMonitoring::InitializeEve(): Caught TEveException: TEveManager::Create window system not initialized.
PandoraMonitoring::InitializeEve(): Attempt to release ROOT from batch mode.
PandoraMonitoring::InitializeEve(): Caught TEveException: TEveManager::Create window system not initialized.
Failure in algorithm 0x4f915e0, LArVisualMonitoring, unrecognized exception
TimeModule> run: 1 subRun: 0 event: 1 pandoraCosmic MicroBooNEPandora 8.53292
</snip>
```

-Relevant code snippets
https://github.com/PandoraPFA/PandoraMonitoring/blob/master/src/PandoraMonitoring.cc#L906
https://github.com/PandoraPFA/PandoraMonitoring/blob/master/src/PandoraMonitoring.cc#L1066

#3 - 03/01/2016 04:52 PM - Gianluca Petrillo

I can now reproduce the problem.

#4 - 03/03/2016 11:22 PM - Gianluca Petrillo

- % Done changed from 0 to 50

The graphics initialization code is sort of all-or-nothing, in that it expects that either everything is set up (interactive mode, graphics client, application, interpreter), or nothing is. The check whether the system is graphics-ready is performed on only one of these, and if that is set up already, it is assumed everything is.

Unfortunately, chances are that the set up is only partial, and this causes the problem we see.
One thing that triggers a partial setup appears to be the GDML parser used internally by ROOT when LArSoft's Geometry service reads the geometry.

Anyway, I have put in place a hack that attempts to make a complete initialization. This code is static code that gets executed when its library is loaded. I have also provided a service, RootGraphicsEnablingService, that drags that library in.
So in short, adding services.RootGraphicsEnablingService: {} will make *art* load the service, and therefore load the library and execute the hack code.
This code is in currently in branch feature/gp_PandoraVisualization of larpandora.

With this service and the change John described in Pandora's XML file, pandora does not complain in the console any more and I can see a window pop up. But I don't know how useful this is, and whether this is enough as a solution, or if the "interactive" mode the job is left in is still unusable.
I suspect this because I don't know how to resume regular *art* processing after Pandora monitoring window shows on screen.

Please advise...

#5 - 03/04/2016 07:17 AM - John Marshall

Thank you Gianluca. This is an excellent step forward, and goes most of the way towards solving the problem. I think that the RootGraphicsEnablingService is an elegant way to address the issue and it certainly ensures that all the TApplication functionality is available at the point at which PandoraMonitoring runs.

The ROOT TEve functionality is complete, so the Pandora EventDisplay and in-algorithm Visualisation needs are satisfied. The problem is then returning the thread to the underlying lar job.

PandoraMonitoring prompts the user to press return in the terminal to continue processing. In this case, however, pressing return simply leaves the user at the (somewhat unexpected) ROOT prompt. The TEve functionality remains accessible at this time. Pressing return a variable/undefined number of times at the ROOT prompt does allow the event processing to continue, but this is somewhat haphazard and inelegant. Using services of the gApplication also seems to (sometimes) allow processing to proceed.

I will see if I can make some additional progress, but a ROOT expert may be able to help find what is still missing. (For reference, in e.g. Marlin, Athena or a very simple application, pressing return cleanly allows the underlying job to continue, although the TEve functionality is actually inaccessible until the next visualisation trigger. When the thread is with TEve, the user can click between any past visualisation requests.)

I should emphasise again that this is a big step forwards and we would like to ensure this functionality is soon available to other users (hopefully with the minor issue above addressed). This functionality allows for development of Pandora algorithms directly within LArSoft (no longer any need to dump events to .pndr files and run in Pandora standalone). It also allows other users to see events exactly as Pandora does, if they are trying to understand the end-product Pandora output.

#6 - 03/04/2016 03:21 PM - John Marshall

- % Done changed from 50 to 90

The expected/desired behaviour is obtained if the RootGraphicsEnabler service creates a TApplication instance, instead of a TRint instance.

Pressing return in the relevant terminal then correctly continues the underlying lar job, with the focus only returned to ROOT TEve again when a Pandora algorithm next requests visualisation.

See suggested diff, below.

```
diff --git a/larpandora/RootGraphics/RootGraphicsEnabler.cxx b/larpandora/RootGraphics/RootGraphicsEnabler.cxx
index adbe54e..0fd4899 100644
--- a/larpandora/RootGraphics/RootGraphicsEnabler.cxx
+++ b/larpandora/RootGraphics/RootGraphicsEnabler.cxx
@@ -18,7 +18,6 @@
 // ROOT libraries
 #include "TROOT.h"
 #include "TApplication.h"
-#include "TRint.h"
 #include "TGClient.h"
 #include "TSystem.h"
 #include "TVirtualX.h"
@@ -89,10 +88,10 @@ namespace { // local namespace
       throw std::runtime_error("RootGraphicsEnabler: no ROOT global pointer");

     if (!app) {
-      if (out) (*out) << "  ==> create a TApplication (TRint)" << std::endl;
+      if (out) (*out) << "  ==> create a TApplication" << std::endl;
       int    argc = 0;
       char** argv = nullptr;
-      new TRint("TRintFromRootGraphicsEnabler", &argc, argv, 0, 0, kTRUE);
+      new TApplication("TApplicationFromRootGraphicsEnabler", &argc, argv);
     } // if no application

     if (out) (*out) << "  ==> set batch mode off (now it is " << (gROOT->IsBatch()? "on": "already off") << s
td::endl;
```

#7 - 03/04/2016 04:13 PM - Gianluca Petrillo

- Status changed from Feedback to Resolved

- % Done changed from 90 to 100

Excellent!
I had a vague feeling that TRint was to blame, but I was, for the rest, almost clueless.
I have pushed the suggested change into larpandora branch feature/gp_PandoraVisualization.

I could not see anything useful in the visualization in my test, but that's probably because my test is not complete enough.
I do see the visualization window and I can make the process follow after it.

#8 - 06/17/2016 04:04 PM - Gianluca Petrillo

- Status changed from Resolved to Closed