

canvas - Bug #17788

"openssl/md5.h" file not found

09/27/2017 01:19 PM - Will Foreman

Status:	Closed	Start date:	09/27/2017
Priority:	Normal	Due date:	
Assignee:	Kyle Knoepfel	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	2.09.00	Spent time:	20.00 hours
Scope:	Internal	SSI Package:	art
Experiment:	LArIAT		

Description

For many jobs I run on the grid (using lariatsoft v06_50_00), I am getting this error in the lar*.out log files:

```
-----  
Graphics systems deleted.  
Visualization Manager deleting...  
MSG-s ArtException: PostEndJob 26-Sep-2017 15:48:39 UTC ModuleEndJob  
cet::exception caught in art  
---- OtherArt BEGIN  
---- FatalRootError BEGIN  
Fatal Root Error: @SUB=  
! (prop&kIsClass) &x%x "Impossible code path" violated at line 445 of  
`/scratch/workspace/canvas-products/v3_00_01_rc2/e14/SLF6/prof/build/root/v6_10_04d/source/root-6.10.04/io/io/src/TGenCollectio  
nProxy.cxx'  
---- FatalRootError END  
---- OtherArt END  
%MSG  
Art has completed and will exit with status 1.  
-----
```

Looking into the corresponding lar*.err file, I see this:

```
-----  
In file included from libcanvas_Persistency_Provenance_dict dictionary payload:13:  
In file included from /grid/fermiapp/products/larsoft/canvas/v3_00_02/include/canvas/Persistency/Provenance/EventAuxiliary.h:7:  
In file included from /grid/fermiapp/products/larsoft/canvas/v3_00_02/include/canvas/Persistency/Provenance/ProcessHistoryID.h:5:  
In file included from /grid/fermiapp/products/larsoft/canvas/v3_00_02/include/canvas/Persistency/Provenance/Hash.h:19:  
/grid/fermiapp/products/larsoft/cetlib/v3_01_01/slf6.x86_64.e14.prof/include/cetlib/MD5Digest.h:11:10: fatal error: 'openssl/md5.h' file  
not found  
#include <openssl/md5.h>  
^  
-----
```

From looking around, it appears that cetlib v3_01_01 does **not** have the openssl libraries which are needed -- specifically, openssl/md5.h. Would it be possible to install the development libraries for openssl onto v3_01_01?

History

#1 - 09/27/2017 04:09 PM - Kyle Knoepfel

- Status changed from New to Feedback

We suspect the openssl error could be a red herring. What is telling is the "Impossible code path" you are seeing from ROOT, which occurs whenever a dictionary is missing for an STL container (e.g. std::vector<YourType>). Is it possible to run this job interactively to reproduce the error? This is the only way we could sensibly debug the problem.

#2 - 09/27/2017 05:09 PM - Will Foreman

Hi Kyle. I'm able to run this job interactively with no errors in my local area (/lariat/app, on any of the lariatgpvms). The error only crops up on the grid.

#3 - 09/28/2017 10:39 AM - Kyle Knoepfel

- Status changed from Feedback to Accepted

We believe that we can diagnose what's going on interactively even if the interactive job does not fail. Could you please send us instructions for running the problematic job interactively. Also include the normal setup commands we would need to execute upon logging on to a lariatgpmv machine.

#4 - 09/28/2017 01:47 PM - Will Foreman

Ah! Yes, sorry. Here are the commands to reproduce the job locally (just 10 events):

(after logging onto lariatgpmv04)

```
source /grid/fermiapp/lariat/setup_lariat.sh
setup lariatsoft v06_50_00 -q e14:prof
cp /lariat/app/users/wforeman/lariat_michels/job/Simulation/test/prodlist_cosmicmuons_1.txt .
lar -c /lariat/app/users/wforeman/lariat_michels/job/Simulation/test/prodtext_lariat_cosmicmuons_1.fcl -n 10
```

#5 - 09/28/2017 04:26 PM - Kyle Knoepfel

Will, this appears to not be the correct job. In the description above, the print out includes "Graphics systems deleted.", which we do not see whenever we run the job. Based on a google search, it looks like that is something that is emitted whenever Geant is run, which does not appear to be run in the job you sent us. Should the largeant module be included in one of the trigger paths?

#6 - 09/28/2017 05:15 PM - Will Foreman

You're right that this isn't the exact script as before, but a simpler script with only the first module in the reco chain (TextFileGen), which still produces the same error on the grid. You can see for yourself in the grid output here:

/pnfs/lariat/scratch/users/wforeman/lariat_michels/mctest4/gen/91503_0

Specifically, at the end of larStage0.out there is this error immediately following simulation of the last particle:

```
...
%MSG-i Root_Information: PostEndRun TInterpreter::AutoParse() 26-Sep-2017 18:50:04 UTC run: 1
Error parsing payload code for class art::ProductID with content:

#line 1 "libcanvas_Persistency_Provenance_dict dictionary payload"

#ifdef G__VECTOR_HAS_CLASS_ITERATOR
  #define G__VECTOR_HAS_CLASS_ITERATOR 1
#endif
#ifdef NDEBUG
  #define NDEBUG 1
#endif

#define _BACKWARD_BACKWARD_WARNING_H
#include "canvas/Persistency/Provenance/BranchChildren.h"
#include "canvas/Persistency/Provenance/BranchID.h"
#include "canvas/Persistency/Provenance/DictionaryChecker.h"
#include "canvas/Persistency/Provenance/EventAuxiliary.h"
#include "canvas/Persistency/Provenance/FileFormatVersion.h"
#include "canvas/Persistency/Provenance/FileIndex.h"
#include "canvas/Persistency/Provenance/History.h"
#include "canvas/Persistency/Provenance/ParameterSetMap.h"
#include "canvas/Persistency/Provenance/Parentage.h"
#include "canvas/Persistency/Provenance/ProcessConfiguration.h"
#include "canvas/Persistency/Provenance/ProcessConfigurationID.h"
#include "canvas/Persistency/Provenance/ProcessHistory.h"
#include "canvas/Persistency/Provenance/ProductID.h"
#include "canvas/Persistency/Provenance/ProductProvenance.h"
#include "canvas/Persistency/Provenance/ProductRegistry.h"
#include "canvas/Persistency/Provenance/ResultsAuxiliary.h"
#include "canvas/Persistency/Provenance/RunAuxiliary.h"
#include "canvas/Persistency/Provenance/SubRunAuxiliary.h"
#include "canvas/Persistency/Provenance/TypeTools.h"
#include "canvas/Utilities/WrappedClassName.h"

#undef _BACKWARD_BACKWARD_WARNING_H

%MSG
%MSG-i NuRandomService: RootOutput:out1@EndJob 26-Sep-2017 18:50:04 UTC ModuleEndJob
```

Summary of seeds computed by the NuRandomService

```
Random policy: 'random'
  master seed: 861620668
  seed within: [ 1 ; 900000000 ]
```

%MSG

```
TrigReport ----- Event Summary -----
TrigReport Events total = 100 passed = 100 failed = 0
```

```
TrigReport ----- Modules in End-Path: end_path -----
TrigReport Trig Bit#      Run      Success      Error Name
TrigReport      0      0      100      100      0 out1
```

```
TimeReport ----- Time Summary ---[sec]-----
TimeReport CPU = 1.687636 Real = 2.598369
```

```
MemReport ----- Memory Summary ---[base-10 MB]-----
MemReport VmPeak = 946.209 VmHWM = 249.213
```

%MSG-s ArtException: PostEndJob 26-Sep-2017 18:50:05 UTC ModuleEndJob

cet::exception caught in art

---- OtherArt BEGIN

---- FatalRootError BEGIN

Fatal Root Error: @SUB=

! (prop&kIsClass) && "Impossible code path" violated at line 445 of `~/scratch/workspace/canvas-products/v3_00_01_rc2/e14/SLF6/prof/build/root/v6_10_04d/source/root-6.10.04/io/io/src/TGenCollectionProxy.cxx'

---- FatalRootError END

---- OtherArt END

%MSG

Art has completed and will exit with status 1.

Additionally, larStage0.err has the same error message mentioned previously:

In file included from libcanvas_Persistency_Provenance_dict dictionary payload:13:

In file included from /grid/fermiapp/products/larsoft/canvas/v3_00_02/include/canvas/Persistency/Provenance/EventAuxiliary.h:7:

In file included from /grid/fermiapp/products/larsoft/canvas/v3_00_02/include/canvas/Persistency/Provenance/ProcessHistoryID.h:5:

In file included from /grid/fermiapp/products/larsoft/canvas/v3_00_02/include/canvas/Persistency/Provenance/Hash.h:19:

/grid/fermiapp/products/larsoft/cetlib/v3_01_01/slf6.x86_64.e14.prof/include/cetlib/MD5Digest.h:11:10: fatal error: 'openssl/md5.h' file not found

#include <openssl/md5.h>

^

#7 - 09/28/2017 07:24 PM - Kyle Knoepfel

Thank you, Will. The full error you post is very helpful--I suspect I know what is happening, but I would like to do some further analysis with Paul Russo (ROOT expert) tomorrow. If I am correct, this is an art bug that is easily fixable.

#8 - 10/02/2017 07:38 AM - Will Foreman

Hi Kyle. While this bug is being looked into, are there any work-arounds I can do to bypass it and start running these jobs on the grid again? Thank you.

#9 - 10/02/2017 08:11 AM - Kyle Knoepfel

- Target version deleted (2.08.03)

Hi Will, unfortunately no, there is no workaround. The only way to have a successful job is if openssl is installed on the grid node. We have a solution to this problem, but it will require an upgrade to a new art version, which has not yet been released--at the earliest, it could be available at the end of this week or early next week.

If a release with the fix is urgent, please let us know.

#10 - 10/02/2017 10:31 AM - Kyle Knoepfel

- Category set to I/O

- Status changed from Accepted to Resolved

- Assignee set to Kyle Knoepfel

- % Done changed from 0 to 100

- SSI Package art added

This was a tricky issue to wade through. The summary is that art failed to define the dictionaries for two different types:

- `std::vector<art::ProductID>`
- `std::pair<art::ProductID, std::set<art::ProductID>>`

When this situation occurs, ROOT has the capability of interrogating header files to find a definition from which a dictionary can be created. For the two types above, ROOT consulted the list of header files corresponding to the `art::ProductID` dictionary, and it then proceeded to open each header file. Opening each header file induces a recursive opening of all dependent header files, one of which was `<openssl/md5.h>`. Not all of the grid nodes have OpenSSL installed, hence the job failures you encountered.

The solution to the problem was to include the types above in the `classes_def.xml` selections file, used to generate the ROOT dictionaries. I have verified that doing so suppresses any openssl file opens.

The fixes to this bug have been included in multiple commits in various *art* suite packages. Although it is simple to incorporate these fixes into a new release of *art*, the *art* team will discuss today how to proceed on a reasonable timescale.

If getting these fixes is more urgent than the timescale of a new *art* release, we can consider providing a separate set of dictionary files that you can use alongside the dictionaries provided by art [2.08.03](#). If you are interested in this proposal, and if you are at Fermilab, please stop by WH9SW so we can discuss.

#11 - 10/02/2017 10:53 AM - Will Foreman

Hi Kyle. Thank you for looking into and fixing this! It looks like the Distributed Computing Support team has added the `openssl-devel` package (which provides `md5.h`) to their containers earlier this morning, so my jobs do in fact run successfully on grid nodes now. So, getting access to the new *art* release is no longer urgent.

#12 - 10/18/2017 01:44 PM - Kyle Knoepfel

- Target version set to 2.09.00

#13 - 10/18/2017 01:45 PM - Kyle Knoepfel

- Status changed from Resolved to Closed

#14 - 10/19/2017 09:38 AM - Kyle Knoepfel

- Project changed from *art* to *canvas*

- Category deleted (I/O)