art - Support #4865

art v1_08_09 attempts to load unwanted modules

10/29/2013 04:24 PM - Andrei Gaponenko

Status:	Closed		**Start date:**	10/29/2013
Priority:	Normal		**Due date:**	
Assignee:	Christopher Green		**% Done:**	100%
Category:	Application		**Estimated time:**	9.00 hours
Target version:			**Spent time:**	1.00 hour
Scope:	Internal		**SSI Package:**	art
Experiment:	Mu2e			

Description

An upgrade from art v1_06_00 to v1_08_09 caused a number of pre-defined jobs to stop working. The reason is that the new version attempts to construct modules even if they are not used in any paths. To reproduce: get Mu2e Offline v4_0_6 and run

$ mu2e -c ExtinctionMonitorFNAL/test/g4detectorSingleProtons.fcl

I get: Library specificaton "EMFCPrint": does not correspond to any library in LD_LIBRARY_PATH of type "module"

The only module of this type, "vdprint", is defined, but not used anywhere.

Andrei

History

#1 - 10/31/2013 11:52 AM - Christopher Green

- Category set to Application

- Status changed from New to Feedback

- Assignee set to Christopher Green

- % Done changed from 0 to 10

- Estimated time set to 9.00 h

- SSI Package art added

- SSI Package deleted ()

Upon looking at your full configuration and investigating further, it seems that your configuration has fallen afoul of a new set of checks implemented in the spirit of Rob's standing request to, "check early, check hard" as far as possible.

To record the actual issue for posterity, you have a module which is specified in physics.producers but is not configured in a path:

```
...
vdprint: { inputInstanceName: "virtualdetector"
         inputModuleLabel: "g4run"
         module_label: "vdprint"
         module_type: "EMFCPrint"
       }
...
```

The library corresponding to the module you specify is not built. As part of the path configuration processing, we examine every configured module to verify that its position in the configuration (producer, analyzer, filter or output) matches its definition in code according to the module's inheritance. In order to do this, each module library is loaded (but an instance is **not** constructed). Errors encountered during this check are accumulated and then posted as an error after all modules have been checked in this way.

We *could* provide an option to store the results of this check and only throw an error for that subset of erroneously configured module entries that are actually mentioned in a used path, but to be honest that is not my preference.

Please talk to Rob and have him present the position of Mu2e at the next stakeholders meeting, where we can discuss it with the other stakeholders. If people think this is a reasonable feature to have, we can do that. I have updated the estimated time for this issue to reflect the time expected to be required to implement this feature.

Thanks,
Chris.

#2 - 10/31/2013 12:09 PM - Andrei Gaponenko

Hi Chris,

I did not know this was an intended change, and the breakage after the upgrade was a surprise.
"Check early, check hard" is a good thing, and I am fine with the current behaviour.
Please close this issue.

Andrei

#3 - 10/31/2013 02:15 PM - Christopher Green

- Tracker changed from Bug to Support

- Status changed from Feedback to Closed

- % Done changed from 10 to 100

Closing per Andrei's request.

In our initial conversation about this problem, I had failed to consider (read: forgot) some of the extra checks I had implemented, and their implications with respect to the need to load libraries for modules configured but not selected.