

art - Bug #3553

dlopen: cannot load any more object with static TLS

02/28/2013 11:06 AM - Andrei Gaponenko

Status:	Closed	Start date:	02/28/2013
Priority:	Immediate	Due date:	
Assignee:		% Done:	100%
Category:	Third Party	Estimated time:	0.00 hour
Target version:	1.08.00	Spent time:	0.00 hour
Occurs In:		Experiment:	-
Scope:	Internal	SSI Package:	-
Description			
Hello,			
When attempting to run a mu2e job on an SL6 system I got			
cet::exception caught in art			
---- Configuration BEGIN			
UnknownModule			
---- Configuration BEGIN			
Unable to load requested library /home/andr/local/mu2e/Offline/lib/libG4_module.so			
dlopen: cannot load any more object with static TLS			
---- Configuration END			
Module G4 with version v1_00_11 was not registered.			
Perhaps your module type is misspelled or is not a framework plugin.			
---- Configuration END			
This is using			
https://oink.fnal.gov/distro/art/art_suite-1.00.11-slf6-x86_64-mu2e-prof.tar.bz2			
https://oink.fnal.gov/distro/art/art_externals-0.04.03-slf6-x86_64-gcc46-prof.tar.gz			
https://oink.fnal.gov/distro/art/mu2e_extras-0.04.03-noarch.tar.gz			
https://oink.fnal.gov/distro/art/mu2e_extras-0.04.03-slf6-x86_64-gcc46-prof.tar.gz			
https://oink.fnal.gov/distro/relocatable-ups/ups-upd-4.9.7-slf6-x86_64.tar.bz2			
Andrei			

History

#1 - 03/04/2013 10:40 PM - Andrei Gaponenko

Hi,

I am able to run mu2e's g4test_03.fcl on the fermicloud057 machine, which is SLF6.3. So we have at least two SL6 system which do not work (positron.triumf.ca I used for the original report, and Krzysztof's workstation), and one that works.

Comparing fermicloud057 and positron, I see that glibc RPMs are identical (glibc-2.12-1.80.el6_3.5.x86_64). The working machine has an older kernel 2.6.32-279.1.1.el6.x86_64, vs 2.6.32-279.22.1.el6.x86_64. What other packages should we compare?

Andrei

#2 - 03/15/2013 11:36 AM - Jim Kowalkowski

- Category set to Third Party

- Status changed from New to Accepted

- Priority changed from Normal to Immediate

The problem has been identified. Our gcc 4.7.1 compiler configuration uses a static run-time option, where the run-time routines are linked statically into every executable. This configuration permits a simpler deployment and management of the compiler distribution.

Unfortunately it has the undesirable effect that it imposes some resources limits that cannot be adjusted - mainly the TLS area that is used up as shared objects are introduced into the running executable.

The solution is to use a different compiler configuration with a dynamic run-time. This will require changes in how the compiler is administered. With

the new version, each executable must be able to find the correct version of the compiler-specific run-time libraries. The art team will be working on this new configuration over the next few weeks (mostly likely Lynn and Paul).

#3 - 08/06/2013 04:20 PM - Christopher Green

- *Status changed from Accepted to Resolved*

- *Target version set to 1.08.00*

- *% Done changed from 0 to 100*

- *Scope set to Internal*

- *Experiment - added*

- *SSI Package - added*

This problem is tentatively resolved by release version 1.08.00, which uses a dynamically-built GCC 4.8.1. Please verify as soon as possible after your experiment switches to this release.

#4 - 10/03/2013 03:15 PM - Christopher Green

- *Status changed from Resolved to Closed*