messagefacility - Feature #17900

Allow configuration of MessageLoggerScribe's plugin factory search paths

10/13/2017 08:03 AM - Ben Morgan

Status:	Assigned	**Start date:**	10/13/2017
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	2.00 hours
Target version:		**Spent time:**	0.00 hour

Description

With #11678 available for a while, I've had some time to look at upstream usage of LibraryManager and PluginFactory. The first main ones look to be here, in source:messagefacility/MessageLogger/MessageLoggerScribe.h#L63. On macOS (Sierra, Xcode 9, SIP enabled) the tests of MF plugins fail as expected. Hacking the two plugin factory definitions to:

```
cet::BasicPluginFactory pluginFactory_{cet::search_path{"MF_PLUGIN_PATH"},"mfPlugin"};
cet::BasicPluginFactory pluginFactory_{cet::search_path{"MF_PLUGIN_PATH"},"mfStatsPlugin"};
```

and setting MF_PLUGIN_PATH to the build time location of the plugins when running the tests makes all tests pass, but of course that's just a proof of principle.

I'm happy to provide patches to support making the MF plugin search paths configurable, but obviously any changes need to be considered for compatibility/configuration changes, so wanted advance from the developers before going ahead with things.

I guess the initial question would be where such a configuration option should come in - as an environment variable, or via FHiCL, or both - and how to modify the interfaces needed in the correct way?

History

#1 - 10/16/2017 11:42 AM - Marc Paterno

- Status changed from New to Feedback

We would be happy to receive patches that implement the support for a SIP-enabled system (although we are not directly supporting that ourselves at this time).

We agree with your proposed use of an environment variable (different from DYLD_LIBRARY_PATH) to use as the search path for the MF plugins. Perhaps MF_PLUGIN_PATH could be set in the UPS table files, as is DYLD_LIBRARY_PATH currently.

We are concerned with difficulty in maintenance, and so wish to avoid conditional compilation. And, as you noted, any changes must not break existing use (on a SIP-disabled machine, and on Linux).

#2 - 10/28/2017 03:13 AM - Ben Morgan

Marc Paterno wrote:

> We would be happy to receive patches that implement the support for a SIP-enabled system (although we are not directly supporting that ourselves at this time).
>
> We agree with your proposed use of an environment variable (different from DYLD_LIBRARY_PATH) to use as the search path for the MF plugins. Perhaps MF_PLUGIN_PATH could be set in the UPS table files, as is DYLD_LIBRARY_PATH currently.
>
> We are concerned with difficulty in maintenance, and so wish to avoid conditional compilation. And, as you noted, any changes must not break existing use (on a SIP-disabled machine, and on Linux).

Thanks Marc, I don't think conditional compilation will be needed, so I'll look at implementing a first iteration to

1. Use MF_PLUGIN_PATH if it is set in the environment
2. Fallback to (DY)LD_LIBRARY_PATH otherwise

Should I consider the case of both being set (combine, prefer one)?

#3 - 10/30/2017 11:41 AM - Kyle Knoepfel

The behavior you describe seems sensible to us. If both are set, prefer MF_PLUGIN_PATH.

#4 - 11/20/2017 09:38 AM - Ben Morgan

- File 0001-Use-MF_PLUGIN_PATH-env-var-for-plugins-is-present.patch added

I've attached a WIP patch for review (based on tag 2_01_01- also viewable online at : https://github.com/drbenmorgan/fnal-messagefacility/commit/1b8ea88abfa58291d7f170ff9228daec2a636ced

The basic behaviour is implemented as discussed earlier

1. Construct the two BasicPluginFactory instances with a cet::search_path that uses
   1. Environment variable MF_PLUGIN_PATH if it is set in the environment
   2. OS-specific dynamic loader path if not

Other than basic structure/style issues, I wasn't sure what to do on checking the exception category here on L67 It should be "getenv" but not sure on CET policy for checking this.

It's also not clang-formatted yet, so I can rebase onto the latest tag if needed.

Tests now pass on both macOS (SIP enabled, using non-UPS build) if MF_PLUGIN_PATH is set manually, and on Linux (using cetbuildtools) using LD_LIBRARY_PATH set by UPS

I wasn't sure how you'd prefer to test these cases, so have left source:messagefacility/test/CMakeLists.txt as-is. To test use of MF_PLUGIN_PATH would be a simple case of adding the line

```
# - MF_PLUGIN_PATH
cet_test_env("MF_PLUGIN_PATH=$<TARGET_FILE_DIR:MF_Utilities>")
```

to that file ((DY)LD_LIBRARY_PATH) can be set in a similar way if required).

#5 - 11/20/2017 11:22 AM - Christopher Green

- Status changed from Feedback to Assigned

- Assignee set to Paul Russo

- Estimated time set to 2.00 h

Thanks for this, Ben. We will review the patch and apply as appropriate.

#6 - 12/03/2019 12:41 PM - Kyle Knoepfel

- Assignee deleted (Paul Russo)

Files

0001-Use-MF_PLUGIN_PATH-env-var-for-plugins-is-present.patch	3.37 KB	11/20/2017	Ben Morgan