cetlib - Feature #22594

readdir_r deprecated in newer glibc

05/15/2019 02:24 PM - Eric Flumerfelt

Status:	Closed	**Start date:**	05/15/2019
Priority:	Normal	**Due date:**	
Assignee:	Paul Russo	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	3.03.00	**Spent time:**	8.00 hours

Description

Attempting to build cetlib on RHEL 8 produces the following error:

```
/home/eflumerf/products/cetlib/v3_07_02/src/cetlib/search_path.cc: In member function ‘size_t cet:
:search_path::find_files(const string&, std::vector<std::__cxx11::basic_string<char> >&) const’:
/home/eflumerf/products/cetlib/v3_07_02/src/cetlib/search_path.cc:136:55: error: ‘int readdir_r(DI
R*, dirent*, dirent**)’ is deprecated [-Werror=deprecated-declarations]
     while (!(err = readdir_r(dd.get(), &entry, &result)) && result != nullptr) {
                                                       ^
In file included from /home/eflumerf/products/cetlib/v3_07_02/src/cetlib/search_path.cc:12:
/usr/include/dirent.h:183:12: note: declared here
 extern int readdir_r (DIR *__restrict __dirp,
            ^~~~~~~~~
/home/eflumerf/products/cetlib/v3_07_02/src/cetlib/search_path.cc:136:55: error: ‘int readdir_r(DI
R*, dirent*, dirent**)’ is deprecated [-Werror=deprecated-declarations]
     while (!(err = readdir_r(dd.get(), &entry, &result)) && result != nullptr) {
                                                       ^
In file included from /home/eflumerf/products/cetlib/v3_07_02/src/cetlib/search_path.cc:12:
/usr/include/dirent.h:183:12: note: declared here
 extern int readdir_r (DIR *__restrict __dirp,
            ^~~~~~~~~
```

More information is here

I am curious; Ubuntu 18's UPS flavor is 4.15-2.27, which should have seen this issue as well...was there a workaround in place?

History

#1 - 05/20/2019 10:31 AM - Kyle Knoepfel

- Status changed from New to Under Discussion

- Tracker changed from Support to Feature

Things have a way of changing between the centOS beta (or not even) and an actual OS release. We have to wait for the official Fermilab supported CentOS 8 release before we can address these issues.

As far as Ubuntu LTS 18 goes, we have not yet attempted an art build, but we will try one.

#2 - 05/20/2019 03:46 PM - Lynn Garren

- Assignee set to Paul Russo

- Status changed from Under Discussion to Assigned

We confirm that this is a problem for Ubuntu LTS 18. In order to preserve thread safety, the code will need to determine which release of glibc is being used.

#3 - 05/29/2019 04:01 PM - Paul Russo

- Status changed from Assigned to Resolved

The readdir(3) documentation for glibc versions greater than or equal to 2.24 (2.29 is the current version) have greatly improved documentation for readdir(3) and readdir_r(3). They explain in detail why readdir_r(3) has been deprecated, and explain how to use readdir(3) properly in multi-threaded code. The summary is that so long as each thread has its own directory stream, then the function is thread-safe. If you must have multiple threads sharing the same directory stream, then you must use locks around your readdir(3) usage. Fortunately our usage naturally has a separate

direcectory stream per thread so I was able to fix the problem by changing the code to use readdir(3) instead (not a trivial change, the function signature and return values are different). This solution has the advantage that it works for both older glibc and newer glibc, no version testing required.

Fixed by commit: bc1af17bb74c6583abd5e4d4df0a2985b33deb30

#4 - 05/29/2019 04:03 PM - Paul Russo

I should note that this change is required for Ubuntu 18.04 LTS and above, and almost certainly for CentOS 8 when it comes out. Also needed for Fedora 30 if anyone cares.

#5 - 06/26/2019 09:04 AM - Kyle Knoepfel

- % Done changed from 0 to 100

- Target version set to 3.03.00

- Status changed from Resolved to Closed