

IEEE Standards Interpretations for IEEE Std 1003.1c™-1995 IEEE Standard for Information Technology--Portable Operating System Interface (POSIX(R)) - System Application Program Interface (API) Amendment 2: Threads Extension (C Language)

Copyright © 1996 by the Institute of Electrical and Electronics Engineers, Inc. 3 Park Avenue New York, New York 10016-5997 USA All Rights Reserved.

Interpretations are issued to explain and clarify the intent of a standard and **do not** constitute an alteration to the original standard. In addition, interpretations are not intended to supply consulting information. Permission is hereby granted to download and print one copy of this document. Individuals seeking permission to reproduce and/or distribute this document in its entirety or portions of this document must contact the IEEE Standards Department for the appropriate license. Use of the information contained in this document is at your own risk.

IEEE Standards Department, Copyrights and Permissions, 445 Hoes Lane, Piscataway, New Jersey 08855-1331, USA

Interpretation Request #44

Topic: Threads Options **Relevant Clauses:** 1.3.1.3, 2.9.3

The POSIX spec lists in 1.3.1.3, page 4, the symbolic constants that define the implementation options required by conforming implementations. Only these four symbols are listed for threads: `_POSIX_THREADS` Threads option (in 2.9.3) `_POSIX_THREAD_SAFE_FUNCTIONS` Thread-Safe Functions option (in 2.9.3) `_POSIX_THREAD_Prio_INHERIT` Priority inheritance option (in 2.9.3) `_POSIX_THREAD_Priority_SCHEDULING` Thread Execution Scheduling option (in 2.9.3) It says that the remaining constants defined in 2.9.3 and 2.9.4 are useful for testing but are not required by a conforming implementation. Functions which are not implemented must be callable with the defined syntax, but can do nothing except return an error.

The following symbolic constants are not listed in 1.3.1.3 or 2.9.3 as required, but are included in the list of options in table 2.10, page 54. `_POSIX_THREAD_ATTR_STACKADDR`: thread stack address attribute functions `_POSIX_THREAD_ATTR_STACKSIZE`: thread stack size attribute functions `_POSIX_THREAD_Prio_PROTECT`: thread priority protection functions `_POSIX_THREAD_PROCESS_SHARED`: process shared synchronization functions I interpret this to mean we can exclude implementing the 12 functions below (providing only a stub that returns the error `ENOSYS`) and still have a conforming implementation. `_POSIX_THREAD_ATTR_STACKADDR` `pthread_attr_getstackaddr()` `pthread_attr_setstackaddr()` `_POSIX_THREAD_ATTR_STACKSIZE` `pthread_attr_getstacksize()` `pthread_attr_setstacksize()` `_POSIX_THREAD_Prio_PROTECT` `pthread_mutexattr_setprioceiling()` `pthread_mutexattr_getprioceiling()` `pthread_mutex_getprioceiling()` `pthread_mutex_setprioceiling()` `_POSIX_THREAD_PROCESS_SHARED` `pthread_mutexattr_getpshared()` `pthread_mutexattr_setpshared()` `pthread_condattr_getpshared()`

`pthread_condattr_setpshared()` One point of confusion is the functions in the `_POSIX_THREAD_PRIORITY_SCHEDULING` option.

Subclause 1.3.1.3 indicates these are required for a conforming implementation; however, each of the functions is allowed to return the error `ENOSYS` if they are not implemented (Clause 13.5). All other functions which I've interpreted as being required cannot return this error. We would like to know the minimum set of options which are required for a POSIX Threads implementation to be conforming.

Interpretation Response

The interpretation request asks two questions. Responses are given for each. 1) (first line) What is the minimum functionality required to be conforming to IEEE 1003.1 1996: The standard is clear on page 3 lines 67 to 81 that a 9945-1 1996 conforming implementation shall support all the non-optional interfaces. 2) (second from last line) What is the minimum set of options which are required for POSIX Threads implementation to be conforming. The standard is clear in defining a set of options in the area of multithreading for implementations to possibly support. It does not define the notion of a 9945-1 1996 'threads' conforming implementation other than as the implementation chooses to support one or more of the thread option as required by conforming applications. Application Environment Profiles (AEPs) are the vehicle for defining further conformance sets required for specific applications.

Rationale for Interpretation

None.