

IEEE Standards Interpretation for IEEE Std 1003.1™-1990 IEEE Standard for Information Technology--Portable Operating System Interfaces (POSIX®)

Copyright © 2001 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 #119

Topic: sigtimedwait and EAGAIN error **Relevant Sections:** page 90, clause 3.3.84, ll 1295-1296 PASC

Austin Group XSHd1 ERN 345 stated the following problem (to be clear these are the words of the originator of the ERN): ETIMEDOUT is used for pthread_cond_timedwait() in exactly the same context. Normally EAGAIN is reserved for use when some resource limit temporarily causes failure of an API, while ETIMEDOUT is specified for a user-specified timeout. Also, all of the realtime amendments in progress use ETIMEDOUT for this. I believe there is either an interp request or an unresolved objection against .1a calling for this to be made consistent. Action: Consider changing this to ETIMEDOUT unless the potential for breakage is too great.

The standard is clear in its requirement for EAGAIN. The issue is whether the standard has a defect. In hindsight ETIMEDOUT would seem a more appropriate error, however would that break applications?

Interpretation Response

The standard clearly states the requirements ,and conforming implementations must conform to this. To change this would break backwards compatibility. No change is required.

Rationale for Interpretation

There is very little reason one would use sigtimedwait() without intending to check for timeout, so we would expect most users of this interface to be broken by the proposed change. Whilst consistency is desirable, there appears to be no way to make a change without breaking source compatibility. Forwarded to Interpretations group: 29 Nov 2000 Proposed resolution: 15 Feb 2001 Finalized: 15 Mar 2001