

## 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 #84

**Topic:** time.h and namespace **Relevant Sections:** Clause 14.1, P309 & subclause 14.2.2.1, P312 (timer\_create())

Although the synopsis for timer\_create() (POSIX.1-1996, P312, L138-141) shows that <signal.h> must be included before <time.h> to pick up the definition of struct sigevent, there are lots of synopses in POSIX.1-1996 that just specify <time.h>. When an application using routines defined in <time.h> that don't also require <signal.h> (e.g. clock\_gettime()) is built, most compilers generate warnings about a dubious tag declaration for struct sigevent.

This is because the specification for <time.h> does not require (or even allow) struct sigevent (and the union sigval needed to define its sigev\_notify\_function element) to be defined and does not reserve namespace to define elements of struct sigevent and union sigval when <time.h> is included. Even though these are warnings, not errors, our customers complain when they use standard functions from standard headers and don't get "clean compiles". In other places where this problem could occur, the C Standard and POSIX standards usually reserve the namespace or require that the needed elements be in both headers. Examples of these are:

1. The description of <stdio.h> in the C Standard (ISO/IEC 9899-1990), P124, Section 7.9.1, 2nd and 3rd paragraphs requires that <stdio.h> define size\_t and NULL in <stdio.h> as well as in <stddef.h>.
2. The description of <mqueue.h> in POSIX.1-1996, P319, L2-4 says that including <mqueue.h> may essentially include <sys/types.h>, <fcntl.h>, <time.h>, and <signal.h> and L13-16 requires that struct sigevent be defined in <mqueue.h> as well as in <signal.h>. Was the intent here to require warnings about dubious structure tags for conforming applications using <time.h>, but not using timer\_create()? Or, was it an

oversight that the description of `<time.h>` did not specify that implementations were allowed to include `<signal.h>` as a side effect of including `<time.h>`?

Add to the end of the paragraph on POSIX.1-1996, P309, L2-3 (description of `<time.h>`): Inclusion of the `<time.h>` header may make visible symbols allowed by this part of ISO/IEC 9945 to be in the `<signal.h>` header. (This copies what appears on POSIX.1-1996, P319, L2-4 for `<mqueue.h>`.)

### **Interpretation Response**

The standard is unclear on this issue, and no conformance distinction can be made between alternative implementations based on this. This is being referred to the sponsor.

### **Rationale for Interpretation**

The standard is ambiguous in its namespace requirements and there appear to be some omissions in the specification. It is recommended that a future revision address this issue. Forwarded to Interpretations group: 26 Feb 1998 Finalized: 30 March 1998