

## 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 #121

**Topic:** `posix_trace_trid_eventid_open()` **Relevant Sections:** Section 21.3.8.2 PASC

It is not at all clear that the function `posix_trace_trid_eventid_open()` need be provided ONLY if the Trace Event Filter option is supported (in addition to the Trace option). Is obtaining a trace event identifier for a trace event name only necessary (in the trace controller process) for purposes of specifying filters, or might the controller process need to perform this mapping for some other purpose? If the latter is true, this function should be provided by implementations that support the Trace option but not the Trace Event Filter option.

Either remove the dependency on the Trace Event Filter option for this function, or provide in the rationale why it is not necessary if this option is not supported.

### Interpretation Response

The standard is clear - the function `posix_trace_trid_eventid_open()` need be provided only by an implementation that supports the Trace Event Filter option. This is intentional, as the filtering of specified events is the only operation which requires that the trace controller process obtain trace event identifiers for a specified stream. The traced process needs these identifiers for many other operations, but may obtain such identifiers by using the `posix_trace_eventid_open()` function, which is not dependent on the Trace Event Filter option. The only references to `posix_trace_trid_eventid_open()` in the rationale are in the context of event filtering.

However, the following paragraph of rationale is misleading (P93, L847-849): "The function `posix_trace_trid_eventid_open()` allows the application to get the system trace event type identifier back from the system, given its well-known trace event name. One possible use of this function is to qualify a filter." Given that the only use of this function

is for a controller process to qualify a filter, forward this interpretation to the sponsor suggesting that this paragraph of rationale be changed to: "The function `posix_trace_trid_eventid_open()` allows the application to get the system trace event type identifier back from the system, given its well-known trace event name. This function is useful only when a controlling process needs to specify events to be filtered."

### **Rationale for Interpretation**

Francois Riche's response follows: There are two functions to create new trace event identifiers: - The `posix_trace_eventid_open()` usable only from the traced process without trace stream identifier because this process may be not traced at the time of the creation of the mapping between trace event type identifier and the trace event name. - The `posix_trace_trid_eventid_open()` usable only from the controller process, a trace stream identifier is requested to identify where this mapping should be memorized. If the controller process has no need to control the set of event types, the fact to get back the trace event type identifier for a given trace event name from the traced process is useless.

Therefore, there is no need to add the implementation of a useless function for an implementation not interested by event type set manipulation. Please have a look to two examples that demonstrates: - B.21.5.6.2, programming example for traced process, - B.21.5.4.2, programming example for controller process. On the other hand, I guess there is an indentation error from line 1047 to 1063, there is a bad one level of indentation for these lines. We have to suppress this indentation. The two other functions are required for the analyzer process as you can see at lines 437 (introduction).

### **Notes to the Project Editor (not part of this interpretation)**

AGR D5 @ page 3487 line 7903-7904 section B.2.11.5 objection Problem: Rationale conflicts with `posix_trace_trid_eventid_open()` being part of the Trace Event Filter option. It implies there are other uses for this function, which cannot be the case if this option is not supported. Action: Replace "One possible use of this function is to qualify a filter." with "This function is useful only when a controlling process needs to specify specific events to be filtered." Forwarded to Interpretations Group: 25 January 2001 Proposed resolution: 21 March 2001 Finalized: April 5 2001