

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 #45

Topic: `creat()` - implementation restrictions **Relevant Sections:** 8.2.3

Chapter 8 of POSIX.1 lists the functions required from the C Standard, required as part of POSIX.1 with the C Language Binding. Section 8.1 lines 25-29 specify the particular input/output functions required. Section 8.2 lines 193-198 state that functions from the C standard which “`creat[e]` a file shall do so as if they called the `creat()` function” with specified flags. Section 8.2.3 lines 341-345 describe that each function operating on a stream has “underlying functions” which “share certain traits with the underlying functions, but [do] not require that there be any relation between the implementations of the stream function and its underlying functions.”

Although section 8.2.3 clearly allows for independent implementation of the function from its underlying functions, are restrictions placed on the underlying function implicitly applied to the function listed here? For example, if an additional standard (such as POSIX.6) imposes additional restrictions on `write()`, are they automatically imposed on functions which use `write()` as an underlying function (such as `fprintf()` or `fputs()`) by virtue of the requirement to “share certain traits”? Or must the additional standard explicitly spell out equivalent restrictions for the functions from the C standard? (Jeremy Epstein)

Interpretation Response

Please refer to interpretations #43 and #44 respectively.