

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 #36

Topic: ENAMETOOLONG **Relevant Sections:** 5.3.1.4 **Classification:** No Change required.

Section 5.3.1.4 open() Errors - ENAMETOOLONG addresses the cases where the path argument exceeds the PATH_MAX limit. If the path argument does not start with a slash, path name resolution (as defined in section 2.3.6) can generate a path name that exceeds the system limit. Is ENAMETOOLONG also required? (Paul Wanish IBM)

Interpretation Response

No, the ENAMETOOLONG error condition is not required for this case.

Rationale for Interpretation

The [ENAMETOOLONG] error condition refers to pathname arguments supplied to the implementation, not to any pathnames that might be internally generated by the implementation. Thus there is no requirement than an implementation fail if the length of such an implementation generated pathname exceeds PATH_MAX.