

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 #30

Topic: Max # of directory levels **Relevant Sections:** 2.3.6

I wish for an interpretation of ISO/IEC 9945-1 (IEEE Std 1003.1) regarding the maximum number of directory levels that a conforming implementation must support. The value {PATH_MAX} defines the maximum number of bytes in a pathname. The error [ENAMETOOLONG] is returned when the length of the path or file arguments exceeds {PATH_MAX}. All other references to {PATH_MAX} are to arguments or the result of the pathconf() function.

All the arguments that are checked against {PATH_MAX} go thru pathname resolution (2.3.6). However, the current working directory mechanism can be used to effectively by-pass any limits imposed by {PATH_MAX}. If the pathname is too long, change the current working directory to some directory on the pathname, delete up to and including that directory, and try again. {PATH_MAX} only controls how many directories can be described at any one call.

If this is the correct interpretation, it would prevent a conforming implementation on file systems that associates each file with an absolute pathname (with filename) stored in a limited size array. Such a file system is used on A Series from Unisys, and I suspect on many other operating systems. I hope that instead the correct interpretation is that the maximum number of directory levels that a conforming implementation must support is limited by {PATH_MAX} when expressed as an absolute pathname. Note that this would not restrict what an implementation may support, only what it must support.

Interpretation Response

No, {PATH_MAX} was not intended to constrain actual absolute pathnames. {PATH_MAX} specifies the length of the string that can be passed to any of a number of

POSIX.1 interfaces as an argument that represents a pathname. IEEE Std 1003.1-1990 does not state an upper limit or a lower limit for the number of levels in directory hierarchies that must be supported.

Since {PATH_MAX} is not the maximum length of an absolute path, the absolute path for a file can be much longer than {PATH_MAX} as long as the pathname used to refer to the file is not longer than {PATH_MAX}.

Rationale for Interpretation

None.