

## 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 #14

**Topic:** format of directory entries, use of read/write **Relevant Sections:** 5.1.1 **Classification:** No change required

In 5.1.1 Format of Directory Entries, page 83, line 10:  
"The internal format of directories is unspecified."

In 5.3.1.4 Errors [Open a File], page 90, lines 256-257:  
"[EISDIR] The named file is a directory, and the oflag argument specifies write or read/write access."

Traditional implementations permitted the use of the read() function on directory files, so no error condition is defined in POSIX.1 for this case. May a conforming application open a directory file and read it using the open() and read() functions?

### Interpretation Response

Yes, a conforming application can invoke read() on a directory, however, the result of any such read() is unspecified, and may be an error return, including [EISDIR].

### Rationale for Interpretation

Nothing in the standard prevents an application from using the open() and read() functions on a directory, but an application which uses knowledge of a particular implementation's format for directories is using a non-portable extension. Implementations are free to return anything they want from a read() to a directory. For example, read() could always return zero, or -1 with some appropriate errno.