

## IEEE Standards Interpretation for IEEE Std 1003.1™-2001 IEEE Standard Standard for Information Technology -- Portable Operating System Interface (POSIX®)

Copyright © 2006 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 #94

**Topic:** pathchk DESCRIPTION **Relevant Sections:** XCU pathchk

In reviewing the pathchk -P option proposed the XCU ERN 39 notes to the editor (see the minutes of the Oct 7 2004 Teleconference), I discovered another problem with pathchk: it is not required to reject empty pathnames. For example, the command "pathchk "" is not required to fail, and on the implementations that I have easy access to (Solaris 9, GNU/Linux) it does not fail. This problem is somewhat independent of the -p option, as the empty pathname is not only unportable: it guaranteed to be invalid. As a result of this problem, a portable application cannot reliably use the command "pathchk -- "\$f" to check the validity of the pathname \$f, because \$f might be empty.

This raises another question, related to both the empty-filename problem and the leading-hyphen problem. Is an implementation of "pathchk -p" allowed to reject filenames with leading hyphens? The XCU ERN 39 interpretation says that the standard does not require "pathchk -p -- -x" to fail, but on the other hand the standard does not clearly require that the command must succeed either. That is, the standard lists several tests that the pathname operand must pass, but it does not say whether the list is exhaustive.

This second question also applies to empty pathnames. Is a conforming implementation of "pathchk" allowed to reject empty pathnames? I.e., can "pathchk "" fail (even though it is not required to fail)?

If the answer to the second question is "yes", that suggests that there is no need for the proposed -P option. If the current standard allows commands like "pathchk "" and "pathchk -p -- -x" to fail, then a revision to the standard could require these commands to fail without affecting portable applications. It would be a win if we could avoid the

need for a new, somewhat-confusing option like -P.

Proposed interpretation response:

The standard is clear that "pathchk "" is not required to fail, and conforming applications must not assume that it fails. However, concerns have been raised about this which are being referred to the sponsor.

The standard is not clear whether commands like "pathchk -p "" , and "pathchk -p "" are allowed to fail in conforming implementations. Nor is it clear whether commands like "pathchk -p -- -x" are allowed to fail. Concerns have been raised about this which are being referred to the sponsor.

Notes to the Editor for a future revision (not part of the interpretation):

After XCU page 696 line 26977 (pathchk DESCRIPTION), add a new bullet:

- \* Is empty.

After XCU page 696 line 26993 (pathchk OPTIONS, under -p), add two new bullets:

- \* Contains any component whose first character is a hyphen.

- \* Is empty.

After XCU page 699 line 27124 (pathchk RATIONALE), add:

Previous editions of this standard did not require pathchk to fail when given an empty pathname, or when given the -p option and a pathname containing a component with a leading hyphen. Such pathnames are invalid (or in the case of -p, are not portable), so the current edition requires pathchk to fail in these cases as well.

### **Interpretation Response**

The standard does not speak to the issue of handling empty pathnames, and as such no conformance distinction can be made between alternative implementations based on this. This is being referred to the sponsor.

### **Rationale for Interpretation**

The usage

```
pathchk -p -- -x
```

is not required to fail since the "-p" has specific defined tests that shall be followed , the characters -x are valid in a component of pathname.

The standard does not address the null pathname. It was felt that for the same reasons as expressed in XCU ERN 39 that this be added to -P rather than -p.