

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 #49

Topic: XBD 7.3.1 LC_CTYPE **Relevant Sections:** XBD 7.3.1 Page: 128 Line: 4133

XBD contradicts the C standard.

XBD's Locale LC_CTYPE "space definition" (which is the basis for the isspace() interface): "space Define characters to be classified as white-space characters. In the POSIX locale, at a minimum, the , , , , and shall be included. [...]"

Compare this to ISO 9899:1999 (C99) says (7.4.1.10, p183 [pdf page 195]):

"The isspace function tests for any character that is a standard white-space character or is one of a locale-specific set of characters for which isalnum is false. The standard white-space characters are the following: space (' '), form feed ('\f'), new-line ('\n'), carriage return ('\r'), horizontal tab ('\t'), and vertical tab ('\v').

In the "C" locale, isspace returns true only for the standard white-space characters."

Note also that POSIX says (XBD p124, line 3953) "Conforming systems shall provide a POSIX locale, also known as the C locale."

This implies that C does not allow any other than the mentioned 6 characters in the "space" character class, while POSIX appears to allow extensions, at least that is how I'd interpret the "at a minimum" apposition. To me, this looks like an unintended ambiguity between C99 and POSIX, which should be resolved.

See the Austin-Group-L mailing list for some discussion that seems to consent that the

suggested action below is the only sensible solution. Thanks to Nick Stoughton for looking up page and line numbers in XBD and ISO 9899:1999.

Action:

Please remove the “at a minimum” apposition from the definition of the “space” character class (LC_TYPE) and replace it by “exactly” for alignment with C99, so that the text then reads:

“space Define characters to be classified as white-space characters. In the POSIX locale, exactly the , , , , , and shall be included. [...rest of paragraph unchanged...]”

Interpretation Response #49

The standards states the requirements for LC_CTYPE, and conforming implementations must conform to this. However, concerns have been raised about this which are being referred to the sponsor.

Rationale for Interpretation

None.