

## IEEE Standards Interpretations for IEEE Std 1003.2™-1992 IEEE Standard for Information Technology--Portable Operating System Interfaces (POSIX®)-- Part 2: Shell and Utilities

Copyright © 1996 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 #110

**Topic:** shell - ENV **Relevant Clauses:** 3.5.3

In subclause 3.5.3, page 123, lines 242-252 the standard says, "This variable, when the shell is invoked, shall be ... This standard specifies the effects of this variable only for systems supporting the User Portability Utilities Option." It isn't clear whether or not, non-interactive shells, and shell scripts are required to expand and use the ENV file. Behavior varies from one implementation to another and overall system performance can be effected. At the time that the UPE was removed and put into a separately balloted standard, there was discussion that indicated that the shell used for system() didn't have to support UPE extensions even on UPE systems, since portable scripts could not rely on the UPE. However, if systems that support UPE require ENV to be expanded for each call to system(), then this dichotomy into two shells would not be possible unless the standard is changed to provide an explicit exception for this. More importantly, if systems that support UPE are required to process ENV files for non-interactive shells, then it is impossible to write scripts that have predictable behavior.

For example, if the ENV file does a set or shift, then the arguments seen by the script will differ than the ones given by the user. Unlike, aliases, functions, and options that are set in the ENV, there is no way for the script to work around these changes. In addition, the standard doesn't specify when a shell will be invoked. For example, if foo is a script, will running foo invoke a shell? Current implementations vary. Some implementations will always invoked a new shell. Others, only if the line, #! /bin/sh is the first line of the script; others, never. This leads to unpredictable behavior. I believe that the standard is unclear about when ENV is to be used on systems that support the UPE. The current wording is the result of editorial choice that occurred when merging two separately balloted standards. The wording in the standard should be changed to make it clear that for

systems that support UPE, ENV is only required to be used for interactive shells.

**Interpretation Response**

The standard states the behavior for the ENV environment variable, 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.