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

IEEE Standards Department, Copyrights and Permissions, 445 Hoes Lane, Piscataway, New Jersey 08855-1331, USA

Interpretation Request #89
Topic: getconf **Relevant Clauses:** 4.26

Clause 4.26 of POSIX.2 specifies the semantics of the “getconf” utility, and in particular subclause 4.26.6.1 describes the standard output of “getconf”. This clause says: If the specified variable is defined on the system and its value is described to be available from the function in 7.8.1 [confstr() in the C binding], its value shall be written in the following format: “%s\n”, <value> Otherwise, if the specified variable is defined on the system, its value shall be written in the following format: “%d\n”, <value> If the specified variable is valid, but is undefined on the system, getconf shall write using the following format: “undefined\n” If the variable name is invalid or an error occurs, nothing shall be written to standard output.

My question has to do with the meaning of the phrases “the specified variable is defined on this system” and “the specified variable is valid, but is undefined on the system”, when the variable in question is one of those listed in Table 2-19 in subclause 2.13.2 of POSIX.2. If an implementation wishes to indicate that it does not support the facilities associated with one of the symbols listed in this table, then must a call to getconf <symbol> write “undefined\n” to standard output, or is it conforming for such an implementation to write “-1\n” to standard output? Relevant text in subclause 2.13.2 reads: Each of these symbols shall be considered valid names by the implementation. Each shall be defined on the system with a value of 1 if the corresponding option is supported; otherwise the symbol shall be undefined. One can interpret this to mean that the only valid outputs from such a call to “getconf” are “undefined” or “- 1”, or one can interpret it to mean that “getconf” can write anything, and that anything other than “1” is to be interpreted as indicating that the symbol is undefined. Which of these interpretations is correct?

Interpretation Response

In the description of getconf it says that if the symbol is defined on the system, it prints that value. If it is valid, but undefined getconf shall write “undefined” as described in subclause 4.26.6.1. For the value from table 2-19, subclause 2.13.2 specifies that the corresponding value shall be 1 if the option is supported. Otherwise, the symbol is undefined. Putting these two together, getconf shall either print the value 1 or the string “undefined” for the symbols listed in table 2-19. The standard clearly states behavior for getconf and conforming implementations must conform to this.

Rationale for Interpretation

None.