

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 #104

Topic: `expr` | operator **Relevant Clauses:** 4.22.7

In IEEE Std 1003.2-1992 subclause 4.22.7 (Extended Description of the `expr` utility), P277, L4159- 4160, the description of the | and & operators is: `expr1 | expr2` Returns the evaluation of `expr1` if it is neither null nor zero; otherwise, it returns the evaluation of `expr2`. `expr1 & expr2` Returns the evaluation of `expr1` if neither expression evaluates to null or zero; otherwise, returns zero.

This requires that the command: `expr "" \| ""` produce an empty line as its output. In both BSD and System V historic practice, the | operator acted as a boolean arithmetic operator like the & operator. If the expressions evaluated to null strings, the result was `"0\n"`, not a null. The rationale in clause E.4.22, P904, L6209-6213 acknowledges that `expr` could be replaced by other shell constructs in the POSIX.2 shell, but says the utility was kept because of the many historical shell scripts that use it. The rationale also mentions that other changes in early drafts of the standard were backed out because they weren't historic practice (see L6229-6230). It seems strange that this change to historical practice was not documented if it was intentional. To match historic BSD and System V implementations, the description of the | operator on P277, L4159- 4169 should have been something like: `expr1 | expr2` Returns the evaluation of `expr1` if it is neither null nor zero; otherwise, the evaluation of `expr2` if it is not null; otherwise, returns zero. Was this change to historic practice intentional?

Interpretation Response

The standard states the behavior for the | operator for the `expr` cmd, 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.