

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 #85

Topic: ERE's **Relevant Clauses:** 2.8.4.1.2 **Subclauses:** 2.8.4.1.2, ERE Special Characters, lines 3069-3072 and B.5.3, Returns, line 424

Subclause 2.8.4.1.1 states, with respect to the repeat characters *, +, ?, and {, that "Any of the following uses produces undefined results: - If these characters appear first in an ERE, or immediately following a vertical line, circumflex, or left parenthesis." This implies that, for instance, the RE " *foo", has undefined results. In clause B.5.3, discussing the return codes from the regexexec and regcomp C API's, the table B-10 includes the error: " REG_BADRPT ?, *, or + not preceded by a valid RE" This text seems to overlap and contradict the previous text. If the repeater is at the beginning of a RE, then it is not preceded by a valid regular expression, which then results in the error.

This section implies that the same RE, " *foo", would result in the error REG_BADRPT, since the NULL character preceding the repeat character is not a valid RE. We would like to see clarification of these two points. Recommendation: It is requested that the implementation be allowed undefined results if the repeat character appears first in the regular expression. Historically, this condition would either be treated as an error, or the repeat character would not be treated specially, as is the case with BRE's. If the repeat character appears after a regular expression which is not a valid expression, this condition should trigger the error. So, the expression " *foo" will produce undefined results, while the expression " f+*oo" would cause a REG_BADRPT (or REG_BADPAT) error condition.

Interpretation Response

The standard does not require the implementation to detect any particular error, nor to return an error in any particular situation. It only requires that the listed errors only be

returned when the indicated error is detected by the implementation. So, `regcomp()` may return `REG_BADRPT` if given the pattern `"*foo"`, since the `*` certainly isn't preceded by a valid ERE specified by the standard. It may also do just about anything else, since the interpretation of this ERE is undefined.

The interpretation request is based on the conclusion that `regcomp (&preg, "*foo", 0);` could reasonably dump core, because the interpretation of `"*foo"` is undefined. The behavior of `regcomp()` with a pattern such as `*foo` produces undefined results. A conforming application shall not expect the return code `REG_BADRPT` from `regcomp()`, if it uses an ERE with a repeat character appearing first or following any of the characters mentioned in subclause 2.8.4.1.2. The standard clearly states behavior for regular expressions and conforming implementations must conform to this.

Rationale for Interpretation

None.