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

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 #155
Topic: ex behaviour **Relevant Clauses:** 5.10.7.3 (Regular Expression) 2.8.3.2 (RE Bracket Expression)

The contents on page 535 lines 1793 - 1799 state that '~' should be expanded to the replacement part of the last substitute command. The section appears to exclude the case when '~' appears at the starting point or at the end point of a range within a bracket expression. The Clause also mentions that '~' loses its special meaning if it is preceded by a '\'. But in a bracket expression the '\' itself loses its special meaning (Ref: 2.8.3.2 page 79-80 lines 2873-2876). So if '~' appears in a range in a bracket expression then should '~' be expanded to the replacement part of the last substitute command?

Proposed Interpretation for 5.10.7.3
The interpretation is based on 5.10.7.3 of POSIX Standard which states (in part): 1793 Match the replacement part of the last substitute command. The 1794 tilde(~) character can be escaped in a BRE to become a normal character 1795 with no special meaning. As per the standard '~' should be expanded to the replacement part of the last substitute command. The '~' loses its special meaning if it is preceded by a '\'.

Proposed Rationale for Interpretation
As is pointed out in the interpretation request the standard is not clear in the case when '~' appears in a range in a bracket expression. If tilde(~) appears at the starting point or at the end point of a range in a bracket expression and if tilde(~) is expanded then the range gets changed or may become an invalid range. This behaviour confuses users. For example consider the following set of vi commands

1. If the user intends to substitute all occurances of "aa" by "bb", then he can use the

following substitute command :s/aa/bb
2. Now if the user intends to search characters which fall in the range ‘a’ to ‘~’, then he can use the command /[a-~] /*

Note: here ‘~’ appears in a range within a bracket expression */ Here vi treats ‘~’ as a metacharacter and expands it to the replacement text of the last substitute command. So in effect it searches for characters in the range ‘a’ to ‘bb’ which is not what the user intended. Moreover, according to line 1794 and 1795 tilde(~) can be escaped to become a normal character. But Section 2.8.3.2 (Special Character), lines 2873-2876 say: 2873 The special characters 2874 . * [\ 2875 (period, asterisk, left bracket, and backslash, respectively) shall 2876 loose their special meaning within a bracket expression. The standard description in lines 1794-1795 and in 2873-2876 are contradicting each other. The standard should clearly state such inconsistency when specific behaviour overrides general behaviour. According to standard, regular expression definition in vi/ex may be different from general regular expression definition. But there should be a standard definition of regular expression and it should not change depending on utility.

According to the section 5.10.7.3 to write the range expresson whose starting point is say ‘a’ and end point is ‘~’ actually [a-\~] should be written, but according to 2.8.3.2 back slash(\) does not have special meaning in bracketed expression and the meaning of above expression is not same as described in second line of this paragraph. If following substitution command is given in ex(1) then the actual strings getting deleted may be different from the intended ones. :%s/[-~]//g For informational purposes, our analysis of current vendor implementations like Sun’s Solaris 5.5.1 and IBM AIX version 2 shows that the historical behavior for this situation is that ‘~’ is not expanded when it appears in a range in a bracket expression. Is it the intention of the standard to diverge from historical practice in this case ? The tilde(~) should not be expanded when it appears in bracketed regular expressions, as the expansion confuses users.

Interpretation Response

The standard is unclear on this issue. The standard states (pg 535 ll 1793-1795) that the tilde character can be escaped in a BRE but does not describe the escape mechanism. As such no conformance distinction can be made between alternative implementations based on this. This is being referred to the sponsor.

Rationale for Interpretation

There are at least two levels of parsing, one in ex/vi before it is passed to the regular expression parsing routines, therefore saying it can be escaped in ex is not in conflict with the other statement. Most existing versions of ex/vi at time standard written had their own RE parsers and it was expected that existing practice would change.