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

Interpretation Request #141
Topic: ed commands **Relevant Sections:** 4.20.7.3

Section 4.20.7.3, page 263, lines 3666-3676 give general rules for dealing with multiple commands on a line. In particular, it states that [A]ny command ... can be suffixed by the letter l, n, or p; in which case ... the command shall be executed and then the new current line shall be written as described below for the l, n, and p commands. ... Also, the g, G, v, and V commands shall take a command as a parameter. Section 4.20.7.3.8, page 266 describes the behaviour of the 'G' command: interactive global command. Section 4.20.7.3.26, page 270 describes the behaviour of the 'V' command: interactive global not- matched command. What is the behaviour of the G and V commands when supplied flags l, n, or p? For example : G/the/nl V/the/nl The description from section 4.20.7.3 would imply that the prompt produced by the 'G' and 'V' commands is unaffect- ed by the suffixed format commands, but that the new current line displayed at the end of the command execute is so affected.

However, historically, ed demonstrates the following behaviour: ed -p '*' *i The wiz-o- matic kitchen utensil will slice, dice, and make JULIENNE fries. . *G/u/nl kitchen uten- sil   <- G prompt line .=   <- user command 2   kitchen utensil$ <- flag applied 2   <- user command output and make JULIENNE fries. <- next G prompt line .= <- next user command 4   <- user command output

* Note in this example that the n and l flags appear to be disabled for second and subse- quent interactive commands (unless the interactive command explicitly specified flags of its own).

This behaviour seems, to us, to be a bug in the historical ed. Should a standard con-

forming ed behave as we described above, namely printing only the new current line according to the format specifiers given, or should it apply the format specifiers to the 'G' and 'V' prompt lines as well, which would satisfy the intent of the (buggy) historical behaviour?

That is, should ed behave as: a) The flags print the new current line after all mmatches have been processed: *G/u/nl kitchen utensil .= 2 and make JULIENNE fries. .= 4 4 and make JULIENNE fries.$ <- new current line printed by the flags as the last action of the G command before returning to the * prompt. * or as: b) Each displayed prompt line by G/V (to which the current lline is set) is displayed according to the flags: *G/u/nl 2 kitchen utensil$ <- prompt written according to flags s/c/X kitXhen utensil <- output of s command 4   and make JULIENNE fries.$ <- prompt written according to flags s/J/Z and make Zulian fries. <- output of s command * This latter behaviour would seem to hew more closely to the behaviour of the historical ed, and is also more useful to the user running the 'G' command.

Interpretation Response
The standard clearly states the requirements for the behavior of ed and conforming implementations must conform to this. The standard states that the flags given on the uppercase global commands causes the new current line after the command has been completed to be written as specified by the flags.

Rationale for Interpretation
None.

Note: (not part of the interpretation)
In the example ed -p '*' , i (insert) won't work as it is not valid at address zero.