

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 #109

Topic: more command **Relevant Clauses:** 5.18

For each of the paragraphs below, I believe that the POSIX description is ambiguous or does not match historic practice.

1: Subclause 5.18.3, page 564, lines 2750-2756 The wording here is ambiguous, and could be read as the -c option can be ignored if the device can't clear to the end of the line OR can't clear to the end of the screen. It should also be noted that if the device doesn't support cursor motion commands, it's not going to work. Change lines 2754-2756 from: its first screen, clear the screen. This option may be ignored on devices that do not have the ability to clear to the end of a line or the end of a screen. To: its first screen, clear the screen. This option may be ignored on devices that have insufficient terminal capabilities.

2: Subclause 5.18.3, page 564, line 2757: The historic -e option is the reverse, i.e. DON'T exit after writing the last line.

3: Subclause 5.18.3, page 564, lines 2766-2778: Subclause 5.18.7, page 568, lines 2913-2915: The whole notion of "current position" in the POSIX specification is badly defined, and used repeatedly. For example, the current position in 5.18.7 is noted as the third line of the file, which isn't historic practice for the +command (which is then correctly specified in lines 2771-2778). Change lines 2768-2769 from: mand, such as a line number of a BRE search, set the current position (see 5.18.7) to represent the final results of the command, without writing any intermediate lines of the file. (For example, To: mand, such as a line number of a BRE search, the first line of the first screen displayed shall be the line moved to as a result of executing the command from the first

line of the file. If the command fails for any reason, the first line of the first screen displayed shall be the default (see 5.18.7) and an error message shall be displayed for the user. In addition, lines 2880-2881 claims that the boundary cases for the current line are described later, and, as far as I can tell, the later explanation doesn't exist. In addition, the specification uses the terms "scrolling", "moving" and "skipping", and much confusion results. For example, lines 2937-2939 list the "scrolling" commands, yet some of these commands don't have "scroll" in their descriptions and the <space> command, not listed here, does have "scroll" in its description. Or, in lines 2925-2932: does the use of the phrase "forward scrolling" at 2930 imply that these specifications are restricted to scrolling commands as listed in 2937- 2939, excluding <space> and <newline>? It seems to me that the "current position" needs to be defined as the 3rd line of the screen. Then, the special case of there being no lines before the "current position" (either because the file is short, or because the target line can't be put on the 3rd line of the screen for whatever reason) has to be described.

4: Subclause 5.18.3, page 564, line 2779: It would be possible to read this as requiring that files that are displayed be modified in place. Change line 2779 from: -s Replace consecutive empty lines with a single empty line. To: -s When acting as a filter, replace consecutive empty lines with a single empty line. When writing to a terminal, act as if consecutive empty lines were a single empty line.

5: Subclause 5.18.3, page 564, line 2781: Subclause 5.18.7, page 568, lines 2911-2912: The description of where the displayed line goes is wrong, historically. Historical practice was that it was the first line on the screen, the same as the +command option. Change line 2781 from: Write the screenful of the file containing the tag named by the To: If the tags file contained a line number, the first line of the first screen displayed shall be that line number. If the tags file contained a pattern, the first line of the first screen displayed shall be the line containing the first occurrence of the pattern in the file. If the specified line does not exist in the file, or the specified pattern is not found, the first line of the first screen displayed shall be the default (see 5.18.7) and an error message shall be displayed for the user.

6: Subclause 5.18.5.2, page 565, line 2806: There's no reason for the input files to be limited to text files. All historical implementations of more handled nul bytes, and it's not acceptable that future implementations be limited. It's probably okay to limit more to LINE_MAX characters, although some historical implementations had no line limits. Change line 2806 from: The input files being examined shall be text files. If standard output is a termi- To: The input files may be text or binary files. If standard output is a termi-

7: Subclause 5.18.6.2, page 567, line 2859-2860: Error messages are historically in the same place as the prompt. Change line 2859-2860 from: Used for diagnostic messages and user commands (see 5.18.5.2) and, if standard output is a terminal device, to write a prompting string. The prompting string To: Used for diagnostic messages, user commands (see 5.18.5.2), and, if standard output is a terminal device, to write a prompting

string. If standard output is a terminal device, the prompting string and diagnostic messages associated with non-fatal errors

8: Subclause 5.18.6.2 There's no specification of what happens if an error message or user input line is longer than a single logical line on the screen. This is clearly a problem, as most terminals will scroll the screen. My guess is that this should be left up to the implementation, but there needs to be some wording of some kind.

9: Subclause 5.18.6.2, page 567, line 2859-2860: Historically, the prompt came after the last line of the screen, so if the screen was smaller than the physical terminal, the prompt was after the last line displayed, not at the bottom of the terminal. This is a better user interface (particularly for small screens), and should be required.

10: Subclause 5.18.6.2, page 567, line 2864: Historically, the mark command also had a user input prompt. Change line 2864 from: prompt. It is unspecified if informational messages are written for other user To: prompt. It is unspecified if user input or informational messages are written for other user

11: Subclause 5.18.7, page 567, line 2871-2872: If the file is small, it may not be possible to write that many lines. Change lines 2871-2872 from: method yields a number, an unspecified number of lines shall be used. The actual number of lines written shall be one less than this number, as the last line of the To: method yields a number, an unspecified number of lines shall be used. The maximum number of lines displayed on the terminal at any one time shall be one less than this number, as the last line of the

12: Subclause 5.18.7, page 568, lines 2916-2917: There's no description of where or how the first screen is written. Change lines 2916-2917 from: If neither the -p or -t options were specified, more shall write the first screen of the first input file. To: If neither the -p or -t options were specified, the first line of the first screen displayed shall be the first line of the first file.

13: Subclause 5.18.7, page 568, line 2922: The -c option did not historically affect the normal scrolling of more, i.e. the 'j' command didn't cause the screen to redraw. The current specification would require that the j command clear and redraw the screen if the -c option was specified. Change line 2922 from: (unless the -c option is specified). If the modification of the screen results in a To: . If the modification of the screen results in a

14: Subclause 5.18.7, page 568, lines 2925-2932: The phrase "reaching end-of-file" has no meaning. In addition, there's no description of what happens after more prompts on intermediate files. Change lines 2926-2932 from: If the standard output is not a terminal device, more always shall exit when it reaches end-of-file on the last file in its argument list. Otherwise, for all files but the last, more shall prompt, with an indication that it has reached end-of-file, along with the name of the next file. For the last file specified, or for the standard input if no file is specified, more shall prompt, indicating end-of-file,

and accept additional commands. If the next command specifies forwards scrolling, more shall exit. If the -e option is specified, more shall exist immediately after writing the last line of the last file. To: If the -e option is specified or the standard output is not a terminal device, more shall exit immediately after it displays the last line of the last file in its argument list. Otherwise, for all files but the last, more shall prompt the user. The prompt shall contain an indication that more has reached the end of the current file and the name of the next file, but is otherwise unspecified. If the user enters a command that implies forward motion (e.g. j or f), more shall continue, displaying the first screen of the next file. For the last file specified (or for the standard input if no files were specified), more shall prompt as described above, indicating end-of-file, and accept additional commands. If the next command implies forward motion, more shall exit.

15: Subclause 5.18.7.1, page 569, lines 2953-2955: The description of reaching end-of-file has already been handled by 5.18.7, and noting it here for f and not for d makes it unclear when it takes effect. Subclause 5.18.7.4, page 569, lines 2967-2969: The description of reaching end-of-file has already been handled by 5.18.7, and noting it here for <newline> and not j or <space> implies that they don't have the same effect which isn't historical practice.

16: Subclause t.18.7.23, page 573, line 3080: The specification doesn't appear to permit more to switch files based on the tag command. Switching files is historic practice.

Interpretation Response

#1: The standard is unclear on this issue, and no conformance distinction can be made between alternative implementations based on this. This is being referred to the sponsor.

#2: The standard states the behavior for -e, and conforming implementations must conform to this. However, concerns have been raised about this which are being referred to the sponsor.

#3: The standard is unclear on this issue, and no conformance distinction can be made between alternative implementations based on this. This is being referred to the sponsor.

#4: The standard clearly states the behavior for -s, and conforming implementations must conform to this.

#5: The standard states the behavior for -t, and conforming implementations must conform to this. However, concerns have been raised about this which are being referred to the sponsor.

#6: The standard clearly states the behavior for input files, and conforming implementations must conform to this.

#7: The standard clearly states the behavior for standard error, and conforming implementations must conform to this.

#8: The standard does not speak to this issue, and as such no conformance distinction can be made between alternative implementations based on this. This is being referred to the sponsor.

#9: The standard states the behavior for the prompt string for standard error, and conforming implementations must conform to this. However, concerns have been raised about this which are being referred to the sponsor.

#10: The standard states the behavior for informational messages for standard error, and conforming implementations must conform to this. However, concerns have been raised about this which are being referred to the sponsor.

#11: The standard states the behavior for the number of lines written per “screen”, and conforming implementations must conform to this. However, concerns have been raised about this which are being referred to the sponsor.

#12: The standard states the behavior for writing the “screen”, and conforming implementations must conform to this. However, concerns have been raised about this which are being referred to the sponsor.

#13: The standard states the behavior for scrolling the “screen”, and conforming implementations must conform to this. However, concerns have been raised about this which are being referred to the sponsor.

#14: The standard states the behavior for exiting more, and conforming implementations must conform to this. However, concerns have been raised about this which are being referred to the sponsor.

#15: The standard states the behavior for reaching the end-of-file, and conforming implementations must conform to this. However, concerns have been raised about this which are being referred to the sponsor.

#16: The standard states the behavior for :t, 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.