

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 #94

Topic: macro processing in make **Relevant Clauses:** 6.2.7.4

It is not clear in POSIX.2-1992 (subclause 6.2.7.4) what the proper behaviour is between the evaluation of macros in target lines and macro assignments. POSIX.2 (subclause 6.2.7.4, lines 471-472) states:

“Macros in target lines shall be evaluated when the target line is read.”

Lines 482-492 describe the rules for macro assignments. Macro assignments shall be accepted from the sources in the following list, in the order shown. If a macro name already exists at the time it is being processed, the newer definition shall replace the existing definition. Macros defined in the inference rules built into make. The contents of the environment, including the variables with null values, in the order defined in the environment. Macros defined in the makefile(s), processed in the order specified. Macros specified on the command line. It is unspecified whether the internal macros defined in 6.2.7.7 are accepted from the command line.

If the -e option is specified, the order of processing sources (2) and (3) shall be reversed. Its unclear what the temporal relationship is between the “processing” of Macro assignments and “when the target line is read”. Are target lines read (and thus macros evaluated) at the same time that the processing of macro assignments are being performed in (3)? I am assuming the answer is yes. If so, then the way I interpret 6.2.7.4 is that macros in target lines shall be evaluated to the current value of the macro at the time the makefile is read, which means that specifying a value on the command line will not be in effect at this time and the target line will expand to an empty value.

The following 2 examples show that historical behaviour does not follow my interpretation.

Example 1: What is supposed to happen when the following command line is executed: `$ make MY_MACRO=command_line` where the makefile contains: `MY_MACRO=in_makefile $(MY_MACRO): echo $$@ = $@, $$MY_MACRO = $(MY_MACRO)` My interpretation of 6.2.7.4 should yield the following result: `$@ = in_makefile, $MY_MACRO = command_line` But, historically the answer is `$@ = command_line, $MY_MACRO = command_line`

Example 2: Using the same makefile as above, what is supposed to happen when the following command line is executed: `$ export MY_MACRO=in_enviro n $ make -e A I interpret subclause 6.2.7.4`, I expect the following result: `$@ = in_makefile, $MY_MACRO = in_enviro n` But, historically the result is: `$@ = in_enviro n, $MY_MACRO = in_enviro n` To rationalize the historical behaviour using the description in subclause 6.2.7.4, this would imply that there is a two-pass process whereby macro assignments are accepted from all of the sources listed in steps (1)-(4) *before* the target lines are read.

Interpretation Response

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.

Rationale for Interpretation

None.