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 #121
Topic: shell **Relevant Clauses:** 3.14.8

The first issue has to do with the output of readonly and export. In subclause 3.14.8, page 155, lines 1549-1551, and again in subclause 4.14.9, page 156, lines 1564-1566, the standard states, "The shell shall format the output, including the proper use of quoting, so that it is suitable for re-input to the shell as commands that achieve the same ... results." In subclause 3.14.8, page 155 line 1548, it lists the output format as "export %s=%s\n", <name>,<value> and in subclause 3.14.9, page 156 line 1563, it lists the output "readonly %s=%s\n", <name>,<value> What should the output of a variable that has the readonly or export attribute, but is unset be? For, example with unset foo export foo export will the line for foo be export foo= or export foo The former follows the explicit output format but violates the rule that on input it will achieve the same result since the former sets the value to the null string rather than leaving the value unset. I believe that the correct result is the latter since the purpose of the -p option was to save and restore shell environments. Therefore, the standard should be amended to allow this format when variable is unset.

The second issue is related to the trap builtin. In subclause 3.14.13, on page 160, lines 1733-1734, the standard states "... the argument actions shall be read and executed by the shell when one of the corresponding conditions arise". It isn't clear when these conditions arise. In particular, historical shells have treated ALRM, SEGV, and CHLD specially. The alarm signal is used internally in historical shells (used to time retry on fork() failures) so that ALRM signals sent by other processes are silently ignored. On some historical shells, the shell increases the size of memory when it receives a SEGV signal rathe than executing the SEGV trap. The CHLD signal is used by some historical shells to keep track of its child processes, some of which may be invisible to the application.

The standard should add words that make it unspecified or implementation defined as to when and if the ALRM, SEGV, or CHLD conditions ever arise.

Interpretation Response
For the first issue: The text on page 156, lines 1564-1566, page 155, lines 1549-1551 are in conflict and as such 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. For the second issue: The standard states the behavior for the trap builtin in the shell and conforming implementations must conform to this. However, we believe it is a defect, and does not match historic practice. It is not obvious that a shell could be made to work given the requirements specified by the standard at this time. Therefore, the concerns raised about this are being referred to the sponsor.

Rationale for Interpretation
None.