

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 #131

Topic: shell signal handling **Relevant Sections:** 3.11

In section 3.11, page 149, lines 1302-11305, the standard states, "When a command is an asynchronous list, the shell shall prevent SIGQUIT and SIGINT signals from the keyboard from interrupting the command. Otherwise, signals shall have the values inherited by the shell process from the parent." This behavior differs from all historical shells. Historical shells prevent all SIGQUIT and SIGINT signals from interrupting the command by default. A command can enable these signals if they wish. The only way to implement this requirement on 1003.1 systems, is to run the background command in a separate process group; and even this won't work if the asynchronous list is brought into the foreground with a fg command. Thus, while the wording is clear, there doesn't seem to be a way to implement this and moreover, it differs from historical practice.

I recommend changing this clause to the following: "When a command is an asynchronous list, the shell shall prevent SIGQUIT and SIGINT signals from the keyboard from interrupting the command when the list is in the background. The shell may set SIGQUIT and SIGINT to ignore for all processes in the list. Otherwise, signals shall have the values inherited by the shell process from the parent."

Interpretation Response

The standard states the requirement for handling the SIGQUIT and SIGINT signals 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.