IEEE P1003.2 Draft 11.2 – September 1991

Copyright © 1991 by the
Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street
New York, NY 10017, USA
All rights reserved as an unpublished work.

This is an unapproved and unpublished IEEE Standards Draft, subject to change. The publication, distribution, or copying of this draft, as well as all derivative works based on this draft, is expressly prohibited except as set forth below.

Permission is hereby granted for IEEE Standards Committee participants to reproduce this document for purposes of IEEE standardization activities only, and subject to the restrictions contained herein.

Permission is hereby also granted for member bodies and technical committees of ISO and IEC to reproduce this document for purposes of developing a national position, subject to the restrictions contained herein.

Permission is hereby also granted to the preceding entities to make limited copies of this document in an electronic form only for the stated activities.

The following restrictions apply to reproducing or transmitting the document in any form: 1) all copies or portions thereof must identify the document's IEEE project number and draft number, and must be accompanied by this entire notice in a prominent location; 2) no portion of this document may be redistributed in any modified or abridged form without the prior approval of the IEEE Standards Department.

Other entities seeking permission to reproduce this document, or any portion thereof, for standardization or other activities, must contact the IEEE Standards Department for the appropriate license.

Use of information contained in this unapproved draft is at your own risk.

IEEE Standards Department
Copyright and Permissions
445 Hoes Lane, P.O. Box 1331
Piscataway, NJ 08855-1331, USA
+1 (908) 562-3800
+1 (908) 562-1571 [FAX]

B.2.1.1 C Macros for Symbolic Limits Rationale. *(This subclause is not a part of P1003.2)*

The macros in column 3 of Table B-3 are required to be constant expressions.

If the C binding is to be used with POSIX.2 implementations over which the implementor of the binding has no control, the column-3 values must be the same as column-2. If the implementation of the C binding is intended to be used with a POSIX.2 implementation that always supports a larger value than one in column 2, that implementation of the binding may use the larger value for the column-3 macro. If an application compiled with that binding is then used with a different POSIX.2 implementation, it is the user's fault that the application is being run in an environment in which it was not intended.

The application can assume, for example, that the stream created by `popen("mailx user","w")` will accept lines of length {LINE_MAX}, even if this is larger than {_POSIX2_LINE_MAX}. However, if the application is creating a data file that might be processed on another implementation, it should use the values in column 2.

B.2.2 Compile-Time Symbolic Constants for Portability Specifications

The macros in Table B-4 shall be defined in the header `<unistd.h>`. These macros can be used by the application, at compile time, to determine which optional facilities are present and what actions shall be taken by the implementation.

Table B-4 – C Compile-Time Symbolic Constants

Macro Name	Description
_POSIX2_C_VERSION	The integer value 199???L. This value indicates the version of the interfaces in this annex that are provided by the implementation. It will change with each published version of this standard to indicate the 4-digit year and 2-digit month that the standard was approved by the IEEE Standards Board.

B.2.2.1 Compile-Time Symbolic Constants for Portability Specifications Rationale. *(This subclause is not a part of P1003.2)*

The symbolic constant _POSIX2_C_VERSION is analogous to _POSIX_VERSION, defined in POSIX.1 {8}. It indicates the version of the C interfaces that are supplied by the compiler and runtime library.

The version of the utilities is given by the {POSIX2_VERSION} limit (see 2.13.1), whose value can be obtained at runtime using *sysconf* () (see B.10.2).

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

B.2.3 Execution-Time Symbolic Constants for Portability Specifications

The macros in Table B-5 can be used by the application at execution time to determine which optional facilities are present. If a macro is defined to have the value −1 in the header <unistd.h>, the implementation shall never provide that feature when the application runs under that implementation. If a macro is defined to have a value other than −1, the implementation shall always provide that feature. If the macro is undefined, then the *sysconf* () function (see B.10.2) can be used to determine if the feature is provided for a particular invocation of the application.

Table B-5 − C Execution-Time Symbolic Constants

Macro Name	Description
_POSIX 2_C_DEV	The system supports the C Language Development Utilities Option (see Annex A)
_POSIX 2_FORT _DEV	The system supports the FORTRAN Development Utilities Option (see Annex C)
_POSIX 2_FORT _RUN	The system supports the FORTRAN Runtime Utilities Option (see Annex C)
_POSIX 2_LOCALEDEF	The system supports the creation of locales as described in 4.35.
_POSIX 2_SW_DEV	The system supports the Software Development Utilities Option (see Section 6)

B.2.4 POSIX. 1 C Numerical Limits

The macros specified in POSIX .1 {8} to provide compile-time values for the configurable variables in Table 7-1 (see 7.8.2) shall also be visible in a POSIX. 2 system. Other macros required by POSIX .1 {8} 2.9 (Numerical Limits) and 2.10 (Symbolic Constants) may also be visible in a POSIX. 2 system.

B.2.4.1 POSIX. 1 C Numerical Limits Rationale. *(This subclause is not a part of P1003.2)*

Subclause 7.8.2 requires that certain POSIX .1 {8} configurable variables be visible in POSIX. 2. Subclause B.2.4 ensures that POSIX. 2 C applications can obtain these variables using the same macros as POSIX .1 {8} C applications. It also allows an implementation to make all of the POSIX .1 {8} macros available even if _POSIX _SOURCE is not set. It also allows an implementation to make all of the POSIX .1 {8} symbols available even if it does not support all of POSIX .1 {8}.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.