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All rights reserved as an unpublished work.

This is an unapproved and unpublished IEEE Standards Draft, subject to change.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.1.1.1 POSIX Symbols Rationale. *(This subclause is not a part of P1003.2)*

When the application defines the _POSIX_C_SOURCE feature test macro with value 2, it must be aware that all of the name space from POSIX.1 {8} and POSIX.2 has been reserved. This does not imply that a POSIX.2 implementation must support POSIX.1 {8}, just that the application must not conflict with an implementation that does. The application can check _POSIX_VERSION and _POSIX2_C_VERSION at compile time to see which standards are supported, if that is necessary. This is primarily an issue for the headers `<stdio.h>`, `<limits.h>`, `<locale.h>`, and `<unistd.h>`, since other POSIX.1 {8} names appear in other headers not mentioned in POSIX.2.

It is expected that C bindings to future POSIX standards and revisions will define new values for _POSIX_C_SOURCE, with each new value reserving the name space for that new standard or revision, plus all earlier POSIX standards. Using a single feature test macro for all standards rather than a separate macro for each standard furthers the goal of eventually combining all of the C bindings into one standard, which will be included in an international standard that refers to a language-independent ISO/IEC 9945-1 {8}.

### B.1.2 Headers and Function Prototypes

Implementations shall declare function prototypes for all functions. Each function prototype shall appear in the header included in the synopsis of the function.

### B.1.3 Error Numbers

Some of the functions in this annex use the variable *errno* to report errors. Such usage is documented in Errors in each specification. The usage of *errno* and the meanings of the symbolic names shall be as defined in POSIX.1 {8} B.1.3.

#### B.1.4 C Language Definitions Rationale. *(This subclause is not a part of P1003.2)*

This clause clarifies the interface to the C Standard {7}. The description was taken from POSIX.1, with one important modification. Since POSIX.1 {8} and the C Standard {7} were being developed and approved at about the same time, POSIX.1 {8} allowed ''Common Usage C'' implementations to give system vendors time to develop Standard C interfaces. Since Standard C compilers are now commonly available, POSIX.2 does not explicitly describe the binding to Common Usage C. However, such a binding would be straightforward, as long as the rules for Common Usage C in POSIX.1 are followed.

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

## B.2 C Numerical Limits

The following subclauses list the names of macros that C language applications can use to obtain minimum and current values for limits defined in 2.13.1.

### B.2.0.1 C Numerical Limits Rationale. *(This subclause is not a part of P1003.2)*

This subclause was added in Draft 9 to give C applications access to limits at compile time. Applications can use the values from the macros without resorting to *sysconf* (). The descriptions very closely follow the descriptions of macros and limits in POSIX .1 {8}.

This definition of the limits is specific to the C language. Other language bindings might use different interfaces or names to provide equivalent information to the application.

Note that there are no C bindings or interfaces that change based on the macros in Table B-5. These macro only advertise the availability of the associated utilities.

### B.2.1 C Macros for Symbolic Limits

The macros in Table B-3 shall be defined in the header `<limits.h>`. They specify values for the symbolic limits defined in 2.13.1.

**Table B-3 – C Macros for Symbolic Limits**

| Symbolic Limit | Minimum Allowed by POSIX.2 | Minimum for this Implementation |
|---|---|---|
| {BC_BASE _MAX} | _POSIX 2_BC_BASE _MAX | BC_BASE _MAX |
| {BC_DIM _MAX} | _POSIX 2_BC_DIM _MAX | BC_DIM _MAX |
| {BC_SCALE _MAX} | _POSIX 2_BC_SCALE _MAX | BC_SCALE _MAX |
| {BC_STRING _MAX} | _POSIX 2_BC_STRING _MAX | BC_STRING _MAX |
| {COLL_WEIGHTS _MAX} | _POSIX 2_COLL _WEIGHTS _MAX | COLL _WEIGHTS _MAX |
| {EXPR_NEST _MAX} | _POSIX 2_EXPR _NEST _MAX | EXPR _NEST _MAX |
| {LINE _MAX} | _POSIX 2_LINE _MAX | LINE _MAX |
| {RE_DUP _MAX} | _POSIX 2_RE _DUP _MAX | RE _DUP _MAX |

The names in the first column of Table B-3 are symbolic limits as defined in 2.13.1. The names in the second column are C macros that define the smallest values permitted for the symbolic limits on any POSIX. 2 implementation; they shall be defined as constant expressions with the most restrictive values specified in 2.13.1. The names in the third column are C macros that define less restrictive values provided by the implementation; each shall be defined as a constant that

— is not smaller than the associated macro in column 2, and

— is not larger than the smallest value that will be returned by *sysconf* () when the application is executed.

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