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All rights reserved as an unpublished work.

This is an unapproved and unpublished IEEE Standards Draft, subject to change.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 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.

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.