--
--

IEEE P1003.2 Draft 11.2 – September 1991

1.2 Normative References

The following standards contain provisions which, through references in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of this International Standard are encouraged to investigate the possibility of applying the most recent editions of the standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards.

{1} ISO/IEC 646: 1983,[2] *Information processing— ISO 7-bit coded character set for information interchange.*

{2} ISO 1539: 1980, *Programming languages— FORTRAN.*

{3} ISO 4217: 1987, *Codes for the representation of currencies and funds.*

{4} ISO 4873: 1986, *Information processing— ISO 8-bit code for information interchange—Structure and rule for implementation.*

{5} ISO 8859-1: 1987, *Information processing—8-bit single-byte coded graphic character sets—Part 1: Latin alphabet No. 1.*

{6} ISO 8859-2: 1987, *Information processing—8-bit single-byte coded graphic character sets—Part 2: Latin alphabet No. 2.*

{7} ISO/IEC 9899: 1990, *Information processing systems—Programming* 1
languages—C.

{8} ISO/IEC 9945-1: 1990, *Information technology—Portable Operating System Interface (POSIX)—Part 1: System Application Program Interface (API) [C Language]*

1.3 Conformance

1.3.1 Implementation Conformance

1.3.1.1 Requirements

A *conforming implementation* shall meet all of the following criteria:

---

2) Under revision. (This notation is meant to explicitly reference the 1990 Draft International Standard version of ISO/IEC 646.)

ISO/IEC documents can be obtained from the ISO office, 1, rue de Varembé, Case Postale 56, CH-1211, Genève 20, Switzerland/Suisse.

(1) The system shall support all required interfaces defined within this standard. These interfaces shall support the functional behavior described herein. The system shall provide the shell command language described in Section 3 and the utilities in Section 4.

(2) The system may provide one or more of the following: the Software Development Utilities Option, the C Language Bindings Option, the C Language Development Utilities Option, the FORTRAN Development Utilities Option, or the FORTRAN Runtime Utilities Option. When an implementation claims that an optional facility is provided, all of its constituent parts shall be provided.

(3) The system may provide additional or enhanced utilities, functions, or facilities not required by this standard. Nonstandard extensions should be identified as such in the system documentation. Nonstandard extensions, when used, may change the behavior of utilities, functions, or facilities defined by this standard. In such cases, the implementation's conformance document (see 2.2.1.2) shall define an execution environment (i.e., shall provide general operating instructions) in which an application can be run with the behavior specified by the standard. In no case shall such an environment require modification of a Strictly Conforming POSIX.2 Application.

1.3.1.2 Documentation

A conformance document with the following information shall be available for an implementation claiming conformance to this standard. The conformance document shall have the same structure as this standard, with the information presented in the appropriately numbered sections; sections that consist solely of subordinate section titles, with no other information, are not required.

The conformance document shall not contain information about extended facilities or capabilities outside the scope of this standard, unless those extensions affect the behavior of a Strictly Conforming POSIX.2 Application; in such cases, the documentation required by the previous subclause shall be included.

The conformance document shall contain a statement that indicates the full name, number, and date of the standard that applies. The conformance document may also list software standards approved by ISO/IEC or any ISO/IEC member body that are available for use by a Conforming POSIX.2 Application. It should indicate whether it is based on a fully-conformant POSIX.1 {8} system. Applicable characteristics where documentation is required by one of these standards, or by standards of government bodies, may also be included.

The conformance document shall describe the symbolic values found in 2.13.2, stating values, the conditions under which those values can change, and the limits of such variations, if any.

The conformance document shall describe the behavior of the implementation for all implementation-defined features defined in this standard. This requirement shall be met by listing these features and providing either a specific reference to the system documentation or providing full syntax and semantics of these features. When the value or behavior in the implementation is designed to be variable or customizable on each instantiation of the system, the implementation provider shall document the nature and permissible ranges of this variation. When information required by this standard is related to the underlying operating system and is already available in the POSIX.1 {8} conformance document, the implementation need not duplicate this information in the POSIX.2 conformance document, but may provide a cross-reference for this purpose.

The conformance document may specify the behavior of the implementation for those features where this standard states that implementations may vary or where features are identified as undefined or unspecified.

No specifications other than those described in this subclause (1.3.1.2) shall be present in the conformance document.

The phrase ‘‘shall be documented’’ in this standard means that documentation of the feature shall appear in the conformance document, as described previously, unless the system documentation is explicitly mentioned.

The system documentation should also contain the information found in the conformance document.

1.3.1.3 Conforming Implementation Options

The following symbolic constants, described in 2.13.2 reflect implementation options for this standard that could warrant requirement by Conforming POSIX.2 Applications, or in specifications of conforming systems, or both:

{POSIX2_SW_DEV}	The system supports the Software Development Utilities Option in Section 6.
{POSIX2_C_BIND}	The system supports the C Language Bindings Option in Annex B.
{POSIX2_C_DEV}	The system supports the C Language Development Utilities Option in Annex A.
{POSIX2_FORT_DEV}	The system supports the FORTRAN Development Utilities Option in Annex C.
{POSIX2_FORT_RUN}	The system supports the FORTRAN Runtime Utilities Option in Annex C.
{POSIX2_LOCALEDEF}	The system supports the creation of locales as described in 4.35.

Additional language bindings and development utility options may be provided in other related standards or in future revisions to this standard. In the former case, additional symbolic constants of the same general form as shown in this subclause should be defined by the related standard document and made available to the application, without requiring this POSIX.2 document to be updated.

1.3.2 Application Conformance

All applications claiming conformance to this standard fall within one of the following categories:

1.3.2.1 Strictly Conforming POSIX.2 Application

A Strictly Conforming POSIX.2 Application is an application that requires only the facilities described in this standard (including any required facilities of the underlying operating system; see 2.9.1). Such an application:

(1) shall accept any implementation behavior that results from actions it takes in areas described in this standard as *implementation-defined* or *unspecified*, or where the standard indicates that implementations may vary;

(2) shall not perform any actions that are described as producing *undefined* results;

(3) for symbolic constants, shall accept any value in the range permitted by this standard, but shall not rely on any value in the range being greater than the minimums listed in this standard;

(4) shall not use facilities designated as *obsolescent*;

(5) is required to tolerate, and is permitted to adapt to, the presence or absence of optional facilities whose availability is indicated by the constants in 2.13.1, or that are described using the verb *may*. However, an application requiring a high-level language binding option can only be considered at best a Conforming POSIX.2 Application; see 1.3.2.2.

Within this standard, any restrictions placed upon a Conforming POSIX.2 Application shall also restrict a Strictly Conforming POSIX.2 Application.

1.3.2.2 Conforming POSIX.2 Application

The term Conforming POSIX.2 Application is used to describe either of the two following application types.

1.3.2.2.1 ISO/IEC Conforming POSIX.2 Application

An ISO/IEC Conforming POSIX.2 Application is an application that uses only the facilities described in this standard (including the implied facilities of the underlying operating system; see 2.9.1) and approved conforming language bindings for any ISO/IEC standard. Such an application shall include a statement of conformance that documents all options and limit dependencies, and all other ISO/IEC standards used.

1.3.2.2.2 <National Body> Conforming POSIX.2 Application

A <National Body> Conforming POSIX.2 Application differs from an ISO/IEC Conforming POSIX.2 Application in that it also may use specific standards of a single ISO/IEC member body referred to here as ‘‘*<National Body>*.’’ Such an application shall include a statement of conformance that documents all options and limit dependencies, and all other *<National Body>* standards used.

1.3.2.3 Conforming POSIX.2 Application Using Extensions

A Conforming POSIX.2 Application Using Extensions is an application that differs from a Conforming POSIX.2 Application only in that it uses nonstandard facilities that are consistent with this standard. Such an application shall fully document its requirements for these extended facilities, in addition to the documentation required of a Conforming POSIX.2 Application. A Conforming POSIX.2 Application Using Extensions shall be either an ISO/IEC Conforming POSIX.2 Application Using Extensions or a <National Body> Conforming POSIX.2 Application Using Extensions (see 1.3.2.2.1 and 1.3.2.2.2).

1.3.3 Conformance Rationale. *(This subclause is not a part of P1003.2)*

These conformance definitions are closely related to those in POSIX.1 {8}.

The terms *Conforming POSIX.2 Application* and its variants were selected to parallel the terms used in POSIX.1 {8}.

The descriptions of the ISO/IEC and <National Body> Conforming POSIX.2 Applications are similar to the same descriptions in POSIX.1 {8}. This is not a duplication of effort, as this standard relies on only a portion of POSIX.1 {8}, as explained in 1.1 and 2.9.1. Therefore conformance to POSIX.2 has to be described separately from any conformance options or requirements in POSIX.1 {8}.

A reference to a Language-Independent System Services Option was removed from the list of optional features that may be provided by the conforming implementation. There is no conformance value provided by that section, except as a reference point for functions actually provided by a real language binding. Therefore, the language binding sections are the ones that remain in the optional list. The Draft 8 section Language-Dependent Services for the C Programming Language was removed, as this subject is adequately, and appropriately, covered in Annex A.

The documentation requirement for implementation extensions (‘‘shall define an execution environment’’) is simply meant to require that system-wide or per-user configuration options or environment variables that affect the operation of applications that use the standard utilities and functions be described in the conformance document. For example, if setting the (imaginary) **LC_TRUTH** variable causes changes in the exit status of `true`, the conformance document must describe this condition and how to avoid it—say, by unsetting the variable in the login script.

For further rationale on the types of conformance, see the POSIX.1 {8} Rationale.

Section 2: Terminology and General Requirements

2.1 Conventions

2.1.1 Editorial Conventions

This standard uses the following editorial and typographical conventions. A summary of typographical conventions is shown in Table 2-1.

Table 2-1 – Typographical Conventions

Reference	Example
C-Language Data Type	*long*
C-Language Function	*system* ()
C-Language Function Argument	*arg1*
C-Language Global External	*errno*
C-Language Header	`<sys/stat.h>`
C-Language Keyword	`#define`
Cross Reference: Annex	Annex A
Cross Reference: Clause	2.3
Cross Reference: Other Standard	ISO 9999-1 {*n*}
Cross Reference: Section	Section 2
Cross Reference: Subclause	2.3.4, 2.3.4.5, 2.3.4.5.6
Defined Term	(see text)
Environment Variable	**PATH**
Error Number	[EINTR]
Example Input	`echo foo`
Example Output	**`foo`**
Figure Reference	Figure 7
File Name	`/tmp`
Parameter	<*directory pathname* >
Special Character	`<newline>`
Symbolic Constant, Limit	{_POSIX_VDISABLE }, {LINE_MAX}
Table Reference	Table 6
Utility Name	`awk`
Utility Operand	*file_name*
Utility Option	`−c`
Utility Option with Option-Argument	`−w` *width*