IEEE P1003.2a Draft 8 – December 1991

Information technology — Portable Operating System Interface (POSIX) — Part 2: Shell and Utilities — Amendment 1: User Portability Extension (UPE)

Section 1: Revisions to General

⇒ **1.1 Scope.** *Modify the contents of clause 1.1, Scope, as shown below.*

1.1 Scope

This standard defines a standard source code level interface to command interpretation, or ''shell,'' services and common utility programs for application programs. These services and programs are complementary to those specified by ISO/IEC 9945-1: 1990 {8}, hereinafter referred to as 'POSIX.1 {8}.'' When the User Portability Utilities Option is included, this standard has additional scope. The list of utilities and features is extended to provide a common interactive environment for system users and program developers.

The standard has been designed to be used by both application programmers and system implementors. When the User Portability Utilities Option is included, it is also to be used by system users and program developers. However, it is intended to be a reference document and not a tutorial on the use of the services, the utilities, or the interrelationships between the utilities.

The emphasis of this standard without the User Portability Utilities Option is on the shell and utility functionality required by application programs (including ''shell scripts'') and not on the direct interactive use of the shell command language or the utilities by humans. When the User Portability Utilities Option is included, the emphasis is extended to support terminal users in a consistent manner across all conforming systems. There are three constraining factors that limit this user portability scope:

(1) The *users* in this context are limited to the group of individuals who are familiar with the style of interaction characteristic of historically derived systems based on one of the UNIX operating systems. Typical users would include program developers, engineers, or general purpose time-sharing users.

(2) The *environment* to be supported is a multi-user time-sharing system supporting character-oriented display terminals. Alternatively, it is a collection of single-user systems interconnected via local area networks or telephone lines, but with similar user interfaces. This standard does not include support tailored for bit-mapped or graphics display terminals, although it is expected that such terminals could emulate the character orientation required by this environment. When facilities require cursor addressability from the terminal hardware, these are specifically identified in the standard.

(3) The facilities to be provided are based on the historical models of the following documents: the *System V Interface Definition* {B24}; the BSD User Manual {B28}; the *X/Open Portability Guide* {B31}; and, documentation for the KornShell {B25}. Emphasis will be placed on standardizing existing practice for existing users, with changes or additions limited to correcting deficiencies in the following areas:

   (a) Support for international character sets and other localization requirements, such as date formats, collation sequences, etc.;

   (b) Reconciliation of differences between the historical implementations;

   (c) Elimination of system or device dependencies;

   (d) Corrections of features that could reduce system or user security/integrity.

Portions of this standard comprise optional language bindings to system service interfaces. See, for example, the C Language Bindings Option in Annex B. This standard is intended to describe language interfaces and utilities in sufficient detail so that an application developer can understand the required interfaces without access to the source code of existing implementations on which they may be based. Therefore, it does not attempt to describe the source programming language or internal design of the utilities; they should be considered ''black boxes'' that exhibit the described functionality.

For language interfaces, or functions, this standard has been defined exclusively at the source code level. The objective is that a conforming portable application source program can be translated to execute on a conforming implementation. The standard assumes that the source program may need to be retranslated to produce target code for a new environment prior to execution in that environment.

There is no requirement that the base operating system supporting the shell and utilities be one that fully conforms to ISO/IEC 9945-1: 1990 {8}. (The base system could contain a subset of POSIX .1 {8} functionality, enough to support the requirements for this standard, as described in 2.9.1, but that could not claim full conformance to all of POSIX .1 {8}.) Furthermore, there is no requirement that the shell command interpreter or any of the standard utilities be written as POSIX .1 {8} conforming programs, or be written in any particular language.

Although not requiring a fully conforming POSIX .1 {8} base, this standard is based upon documentation and the knowledge of existing programs that assume an interface and architecture similar to that described by POSIX .1 {8}. Any questions regarding the definition of terms or the semantics of an underlying concept should be referred to POSIX .1 {8}.

1.1.1 Scope Rationale. *(This subclause is not a part of P1003.2a)*

The purpose of this supplement is to extend the benefits of the common shell and tools environment from the application program to the interactive system user, by defining optional facilities in a supplement to the original standard, a ‘‘User Portability Extension.’’ The original standard’s scope limited the selection of utilities and features to those useful in a noninteractive environment only. Historically, however, many of the tools described in the original standard have seen interactive use; this supplement is predicated on the need to offer a completely consistent interactive environment for the users of such tools.

The scope was carefully worded to exclude the development of interfaces for new technology environments, such as bit-mapped graphics windowing terminals, and ‘‘look-and-feel’’ considerations (icons, menus, mice, etc.) for novice or specialized system users. There are numerous other standards groups in this domain and POSIX. 2 did not want to overlap, but merely satisfy historical users.

The acknowledgment of four base documents placed the UPE effort squarely within guidelines established by the developers of the standard (the IEEE Technical Committee on Operating Systems and Operational Environments) to anchor proposed standards on industry practice and documentation. Since a primary purpose was to carry users familiar with historical practice along into a more portable open systems environment, the base documents were closely adhered to in terms of utility names, features, and behaviors. Only the four exception criteria called for any modifications to historical practice.

Selecting the KornShell documentation was a clear signal by the working group 8
that `ksh` extended scripting features and interactive command-line editing 8
features were major factors in selecting POSIX shell behavior. The System V shell 8
and the C-shell are also in the base documents, but numerous working group dis- 8
cussions resulted in the following guidelines used to prepare and ballot the drafts: 8

— Use the System V shell as the broad shell scripting base, but select Korn- 8
Shell features to correct historical deficiencies (such as the lack of easy 8
arithmetic and substring manipulation, the tilde expansion, and the prob- 8
lems of command substitution nesting). 8

— Use the KornShell model for interactive features such as command-line 8
editing. 8

— When required features appeared in both the C-shell and KornShell, 8
emphasize the KornShell interface unless it was clearly deficient. This 8
decision was predicated on the fact that it made less sense to select script- 8
ing features and interactive features from two very different shells. Histor- 8
ical C-shell users would find the POSIX shell to be rather different than 8
their experience and selecting isolated C-shell features would only serve to 8
confuse both C-shell and KornShell users. 8

1.3.1.3 Conforming Implementation Options

⇒ **1.3.1.3 Conforming Implementation Options.** *Modify the contents of subclause 1.3.1.3, to add the following table entries in the proper sorted order:*

{POSIX 2_CHAR _TERM }	The system supports at least one terminal type capable of all operations described in this standard. See 2.14.
{POSIX 2_UPE }	The system supports the User Portability Utilities Option in Section 5.

Conforming Implementation Options Rationale. *(This subclause is not a part of P1003.2a)*

See the rationale for 2.14.

Section 2: Revisions to Terminology and General Requirements

2.2.1 Terminology

⇒ **2.2.1 Terminology.** *Modify the contents of subclause 2.2.1, Terminology, to add the following definition in the correct sorted order [disregarding the subclause number shown here].*

2.2.1.1 application: When the User Portability Utilities Option is supported, requirements associated with the term *application* shall also be interpreted to include the actions of the user who is interacting with the system by entering shell command language statements from a terminal.

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

The term *application* is not defined in either POSIX.1 {8} or POSIX.2; it is assumed to be a part of general computer science terminology. The definition in this subclause is present to ensure readers understand that the general requirements for applications also apply to interactive users. If, for instance, POSIX.2 requires a Strictly Conforming POSIX.2 Application to use a specific option to a utility, an interactive user would also have to use that option to be considered strictly conforming.