

# ***A White Paper from The Open Group***

## **The Single UNIX Specification, Version 3**

*The Open Group*

*Copyright © May 2003, The Open Group*

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the copyright owners.

A White Paper from The Open Group  
The Single UNIX Specification, Version 3

Published in the U.K. by The Open Group, May 2003.

Any comments relating to the material contained in this document may be submitted to:

The Open Group  
Apex Plaza  
Forbury Road  
Reading  
Berkshire, RG1 1AX  
United Kingdom

or by Electronic Mail to:

[OGSpecs@opengroup.org](mailto:OGSpecs@opengroup.org)

# The Single UNIX Specification, Version 3

This white paper gives background on the development of the Single UNIX Specification, Version 3 and an overview of the documents that comprise the different parts of the document and how they are organized. The Single UNIX Specification, Version 3 is made up of the Base Specifications, Issue 6 and X/Open Curses, Issue 4, Version 2. The Base Specifications, Issue 6 comprise Base Definitions, System Interfaces, Shell and Utilities, and Rationale.

## 1.1 The Development of the Specification

Work commenced in late 1998 on the Single UNIX Specification, Version 3. The core of the Single UNIX Specification, Version 3, collectively known as the *Base Specifications*, was developed, and is maintained, by a joint working group of members of the IEEE Portable Applications Standards Committee, members of The Open Group, and members of ISO/IEC Joint Technical Committee 1. This joint working group is known as the Austin Group.<sup>1</sup>

The Austin Group arose out of discussions amongst the parties which started in early 1998, leading to an initial meeting and formation of the group in September 1998. This represented a seachange in attitude regarding development of two related specifications by three development organizations, which to date had been developed separately, often with the same standards developers involved.

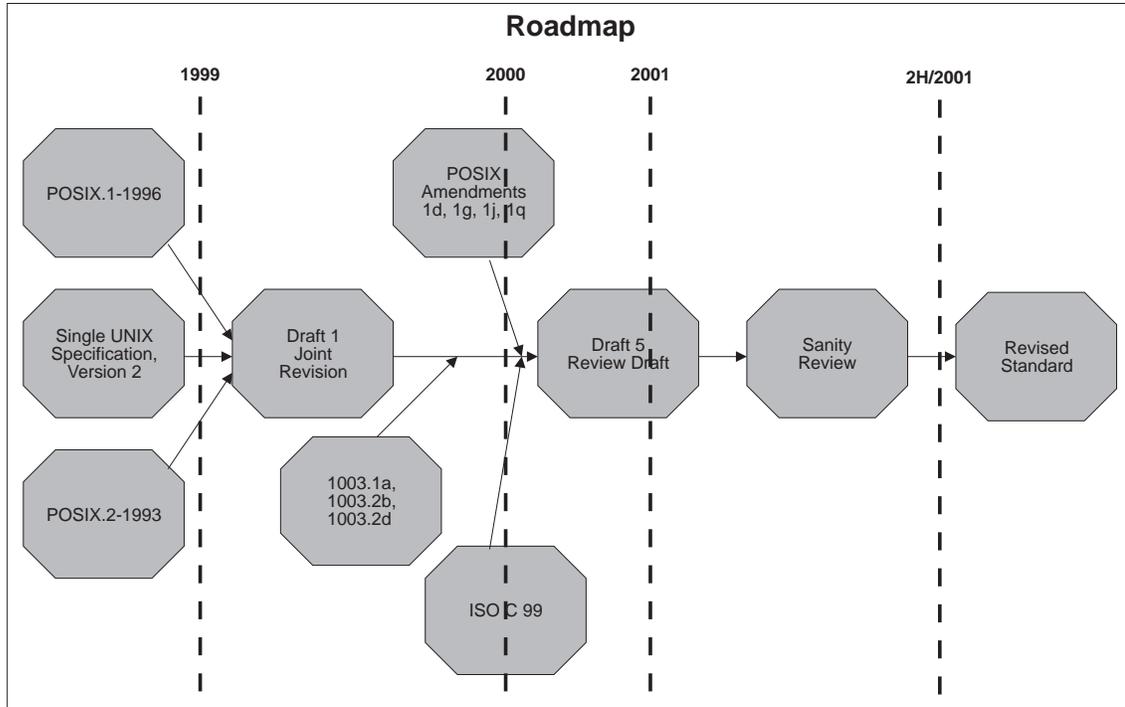
The purpose of the Austin Group has been to revise, combine, and update the following standards: ISO/IEC 9945-1, ISO/IEC 9945-2, IEEE Std 1003.1, IEEE Std 1003.2, and the Base Specifications of The Open Group Single UNIX Specification.

After two initial meetings, an agreement was signed in July 1999 between The Open Group and the Institute of Electrical and Electronics Engineers (IEEE), Inc. to formalize the project, with the first draft of the revised specifications being made available at the same time. Under this agreement, The Open Group and IEEE agreed to share joint copyright of the resulting work. The Open Group has provided the chair and secretariat for the Austin Group.

The following figure shows the development roadmap from base documents through to approved standard:

---

1. The Austin Group is named after the location of the inaugural meeting held at the IBM facility in Austin, Texas in September 1998.



This unique development has combined both the industry-led efforts and the formal standardization activities into a single initiative, and included a wide spectrum of participants including commercial, academia, government, and the open source communities.

The approach to specification development was one of “write once, adopt everywhere”, with the resulting set of specifications being approved as IEEE Std 1003.1-2001 (POSIX), The Open Group Base Specifications, Issue 6 and ISO/IEC 9945:2002. In late 2002, Technical Corrigendum 1 was produced, which corrected a number of defects in the 2001 publication. A 2003 edition of the Base Specifications, Issue 6 and IEEE Std 1003.1, was published on March 31st 2003.

The Base Specifications, Issue 6 consist of the following Technical Standards:

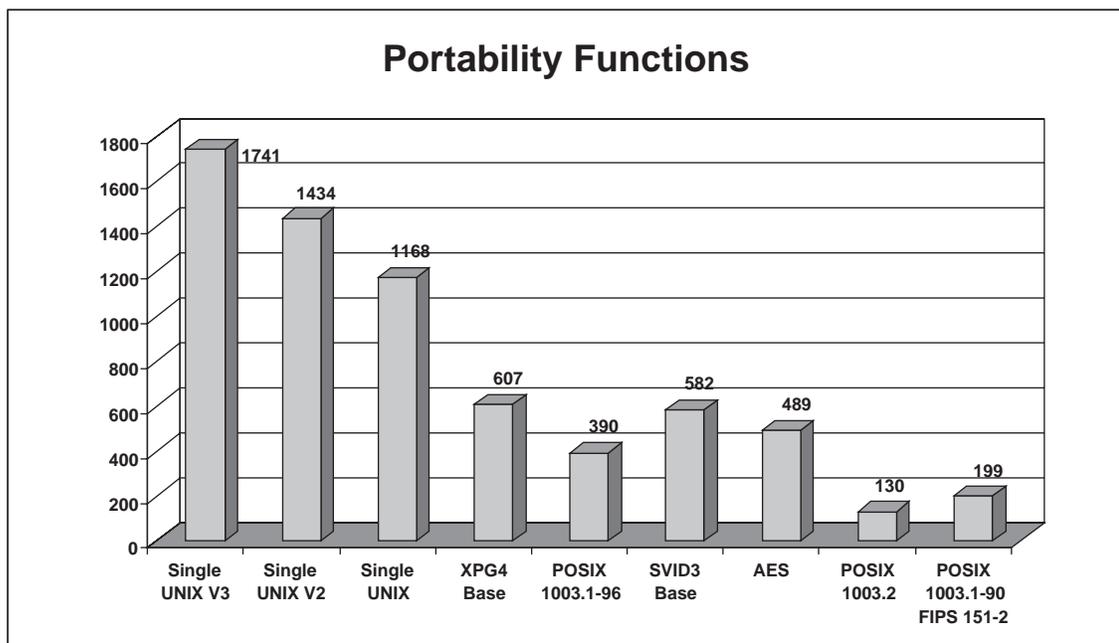
- Base Definitions, Issue 6 (XBD)
- Shell and Utilities, Issue 6 (XCU)
- System Interfaces, Issue 6 (XSH)
- Rationale (Informative)

The revision of the Base Specifications has tried to minimize the number of changes required to implementations which conform to the earlier versions of the approved standards to bring them into conformance with the current standard. Specifically, the scope of this work excluded doing any “new” work, but rather collecting into a single document what had been spread across a number of documents, and presenting it in what had been proven in practice to be a more effective way. Some changes to prior conforming implementations were unavoidable, primarily as a consequence of resolving conflicts found in prior revisions, or which became apparent when bringing the various pieces together.

However, since the revision now references the 1999 version of the ISO C standard, there are a number of unavoidable changes that have been made which will affect applications portability.

In addition to the Base Specifications, the Single UNIX Specification, Version 3 includes the X/Open Curses, Issue 4, Version 2 specification. Updates to X/Open Curses, Issue 4, Version 2 have been limited to production of a Corrigendum to allow it to exist in a Base Specifications, Issue 6 environment.

The following figure shows the interface counts for the different versions of the Single UNIX Specification and related documents.



### 1.1.1 IEEE Std 1003.1

The core of the Single UNIX Specification, Version 3 is also IEEE Std 1003.1. IEEE Std 1003.1 is a major revision and incorporates IEEE Std 1003.1-1990 (POSIX.1) and its subsequent amendments, and IEEE Std 1003.2-1992 (POSIX.2) and its subsequent amendments, combined with the core volumes of the Single UNIX Specification, Version 2. It is technically identical to The Open Group, Base Specifications, Issue 6; they are one and the same documents, the front cover having both designations. The final draft achieved 98% approval by the IEEE ballot group and was officially approved by the IEEE-SA Standards Board on December 6th 2001. During 2002, Technical Corrigendum 1 was developed. It was approved by the IEEE-SA Standards Board on December 11th 2002, and the 2003 Edition of IEEE Std 1003.1 is officially designated as including IEEE Std 1003.1-2001 and IEEE Std 1003.1-2001/Cor 1-2002 .

### 1.1.2 ISO/IEC 9945:2002

The core of the Single UNIX Specification, Version 3 is also ISO/IEC 9945:2002. ISO/IEC 9945:2002 is the third edition of ISO/IEC 9945 and incorporates the former ISO/IEC 9945-1:1996 (POSIX.1), and ISO/IEC 9945-2:1993 (POSIX.2) plus subsequent IEEE amendments, combined with the core volumes of the Single UNIX Specification, Version 2. It is technically identical to The Open Group, Base Specifications, Issue 6 and IEEE Std 1003.1-2001; they are one and the same documents, the front cover having all the designations. The final draft passed its final ballot on September 19 2002. It was published on November 15 2002.

ISO/IEC 9945 consists of the following parts, under the general title Information technology Portable Operating System Interface (POSIX B.):

- Part 1: Base Definitions
- Part 2: System Interfaces
- Part 3: Shell and Utilities
- Part 4: Rationale

At the time of writing Technical Corrigendum 1 to ISO/IEC 9945:2002 has been approved and a new edition of the standard is being prepared for publication.

### 1.1.3 The Austin Group

The Austin Group continues as the maintenance body for the Base Specifications.

Anyone wishing to participate in the Austin Group should contact the chair with their request. There are no fees for participation or membership. You may participate as an observer or as a contributor. You do not have to attend face-to-face meetings to participate; electronic participation is most welcome. For more information on the Austin Group and how to participate, see <http://www.opengroup.org/austin>.

## 1.2 Organization of the Specification

The following sections give an overview of the document that comprise the different parts of the Single UNIX Specification, Version 3.

### 1.2.1 Base Definitions (XBD)

The XBD document is part of the Base Specifications, Issue 6. XBD provides common definitions for the Base Specifications of the Single UNIX Specification; therefore readers should be familiar with it before using the other parts of the Single UNIX Specification. The presence of this document reduces duplication in the other related parts of the Single UNIX Specification and ensures consistent use of terminology.

This document is structured as follows:

- Chapter 1 is an introduction which includes the scope of the Base Specifications, and the scope of the changes made in this revision. Normative references, terminology, and portability codes used throughout the Base Specifications are included in this chapter.
- Chapter 2 defines the conformance requirements, both for implementation and application conformance. For implementation conformance, this includes documentation requirements, conformance definitions for the core POSIX subset, conformance definitions for systems conforming to the Single UNIX Specification (denoted as the XSI extension), and option groups (previously known as feature groups).
- Chapter 3 contains the general terms and definitions that apply throughout the Base Specifications.
- Chapter 4 describes general concepts that apply throughout the Base Specifications.
- Chapter 5 describes the notation used to specify file input and output formats in XBD and XCU.
- Chapter 6 describes the portable character set and the process of character set definition.

- Chapter 7 describes the syntax for defining internationalization locales as well as the POSIX locale provided on all systems.
- Chapter 8 describes the use of environment variables for internationalization and other purposes.
- Chapter 9 describes the syntax of pattern matching using regular expressions employed by many utilities and matched by the *regcomp()* and *regexexec()* functions. Both Basic Regular Expressions (BREs) and Extended Regular Expressions (EREs) are described in this chapter.
- Chapter 10 describes files and devices found on all systems and their semantics. For example, the device */dev/null* is an infinite data source and data sink.
- Chapter 11 describes the asynchronous terminal interface for many of the functions in XSH and the *stty* utility in XCU.
- Chapter 12 describes the policies for command line argument construction and parsing. It contains the utility argument syntax used throughout XCU, and also utility syntax guidelines for naming of utilities and the specification of their arguments and option-arguments and operands.
- Chapter 13 defines the contents of headers which declare constants, macros, and data structures that are needed by programs using the services provided by the system interfaces defined in XSH. These are in the form of reference pages and are organized alphabetically.

### 1.2.2 System Interfaces (XSH)

The XSH document is part of the Base Specifications, Issue 6. XSH describes a set of system interfaces offered to application programs by systems conformant to this part of the Single UNIX Specification. Readers are expected to be experienced C language programmers, and to be familiar with the XBD document.

This document is structured as follows:

- Chapter 1 explains the status of this document and its relationship to other formal standards. The scope, conformance, and definitions sections are pointers to the XBD document; the sections are here to meet ISO/IEC rules regarding required sections. The terminology and portability codes are identical to the section in XBD and repeated here for ease of reference.
- Chapter 2 contains important concepts, terms, and caveats relating to the rest of this document. This includes information on the compilation environment, the name space, definitions of error numbers, signal concepts, standard I/O streams, STREAMS, XSI IPC, realtime, threads, sockets, tracing, and data types.
- Chapter 3 defines the functional interfaces to systems conformant to this part of the Single UNIX Specification. These are in the form of reference pages and are organized alphabetically.

#### How to Read an XSH Reference Page

Each reference page in XSH has a common layout of sections describing the interface. (Function interface descriptions in X/Open Curses follow the same layout.) This layout is similar to the manual page or “man” page format shipped with most UNIX systems, and each interface has SYNOPSIS, DESCRIPTION, RETURN VALUE, and ERRORS sections. These are the four sections that relate to conformance.

Additional sections contain considerable extra information for the application developer. The EXAMPLES sections provide source code examples of how to use certain interfaces. The APPLICATION USAGE sections provide additional caveats, issues, and recommendations to the

developer. The SEE ALSO sections contain useful pointers to related interfaces and headers that a developer may wish to also read.

The FUTURE DIRECTIONS sections act as pointers to related work that may impact on the interface in the future, and often cautions the developer to architect the code to account for a change in this area. (A FUTURE DIRECTIONS section expresses current thinking and should not be considered a commitment to adopt the feature or interface in the future.)

A new RATIONALE section has been added to this version of the specification which includes historical information about an interface and why features were included or discarded in the definition.

The CHANGE HISTORY section describes when the interface was introduced, and how it has changed. This information can be useful when porting existing applications that may reflect earlier implementations of the interface.

Option Group labels in the reference page headers, and portability shading and margin marks are features already described in this document; they appear on the reference pages to guide an application developer when deciding how best an interface should be used. Refer to XSH, Section 1.8, Format of Entries for information on the exact layout.

### 1.2.3 Shell and Utilities (XCU)

The XCU<sup>2</sup> document is part of the Base Specifications, Issue 6. XCU describes the shell and utilities that are available to application programs on systems conformant to this part of the Single UNIX Specification. Readers are expected to be familiar with the XBD document.

This document is structured as follows:

- Chapter 1 explains the status of this document and its relationship to other formal standards, including the ISO C standard and also the XSH document. It also describes the utility limits, grammar conventions, defaults used by the utility descriptions, considerations for utilities in support of large files, and the list of required built-in utilities. The scope, conformance, and definitions sections are pointers to the XBD document; the sections are here to meet ISO/IEC rules regarding required sections. The terminology and portability codes are identical to the section in XBD and repeated here for ease of reference.
- Chapter 2 describes the command language—that is, the shell command language interpreter—used in systems conformant to the Single UNIX Specification.
- Chapter 3 describes a set of services and utilities that are implemented on systems supporting the Batch Environment option.
- Chapter 4 consists of reference pages for all utilities available on systems conforming to the Single UNIX Specification. These are in the form of reference pages and are organized alphabetically.

---

2. The acronym “XCU” derives from the previous version of the specification which was called “Commands and Utilities”.

### How to Read an XCU Reference Page

Each reference page in XCU has a common layout of sections describing the interface. This layout, while similar to the manual page or “man” page format shipped with most UNIX systems, offers a more detailed view of the utility’s description.

As well as the SYNOPSIS and DESCRIPTION sections, each interface has OPTIONS, OPERANDS, STDIN (standard input format), INPUT FILES, ENVIRONMENT VARIABLES, ASYNCHRONOUS EVENTS (what signals are caught and the consequence of receiving signals), STDOUT (standard output format), STDERR, and OUTPUT FILES sections.

An EXTENDED DESCRIPTION will be used if the utility has a particularly long description; for example, if it supports its own language (*awk*), or provides a considerable number of subcommands (*mailx*).

Utilities generally return 0 upon successful completion, and a failed status as greater than 0. The EXIT STATUS sections specify this, but will also describe if particular values are returned in certain circumstances. In general, an application should be written to test for successful completion, rather than specific error returns.

The CONSEQUENCE OF ERRORS section describes what happens to such items as open files, process state, and the environment, should errors occur.

As with the XSH reference pages, additional sections contain considerable extra information for the application developer, and include EXAMPLES, APPLICATION USAGE, FUTURE DIRECTIONS, RATIONALE, SEE ALSO, and CHANGE HISTORY sections. The defaults for these sections, and additional detail about what each section specifies, are covered in XCU, Section 1.11, Utility Description Defaults.

#### 1.2.4 Rationale (XRAT)

The XRAT document is part of the Base Specifications, Issue 6. The XRAT document has been published to assist in the process of review and understanding of the main text. It contains historical information concerning the contents of the Base Specifications, Issue 6 and why features were included or discarded by the standard developers. It also contains notes of interest to application programmers on recommended programming practices, emphasizing the consequences of some aspects that may not be immediately apparent.

This document is organized in parallel to the normative documents of the Base Specification, with a separate part (Parts A, B, and C) for each of the three normative documents. In addition, two additional parts are included: Part D, Portability Considerations and Part E Subprofiling Considerations. The Portability Considerations chapter includes a report on the perceived user requirements for the Base Specification and how the facilities provided satisfy those requirements, together with guidance to writers of profiles on how to use the configurable options, limits, and optional behavior. The Subprofiling Considerations chapter satisfies the requirement that the document address subprofiling. This contains an example set of subprofiling options.

### 1.2.5 X/Open Curses (XCURSES)

XCURSES is not part of the Base Specifications, Issue 6. XCURSES describes a set of interfaces providing a terminal-independent method of updating character screens that are available to application programs on systems conformant to this part of the Single UNIX Specification. This document should be read in conjunction with The Open Group Corrigendum U056.

This document is structured as follows:

- Chapter 1 introduces Curses, gives an overview of enhancements that have been made to this version, and lists specific interfaces marked TO BE WITHDRAWN. This chapter also defines the requirements for conformance to this document and shows the generic format followed by interface definitions in Chapter 4.
- Chapter 2 describes the relationship between Curses and the C language, the compilation environment, and the X/Open System Interface (XSI) operating system requirements. It also defines the effect of the interface on the name space for identifiers and introduces the major data types that the interfaces use.
- Chapter 3 gives an overview of Curses. It discusses the use of some of the key data types and gives general rules for important common concepts such as characters, renditions, and window properties. It contains general rules for the common Curses operations and operating modes. This information is implicitly referenced by the interface definitions in Chapter 4. The chapter explains the system of naming the Curses functions and presents a table of function families. Finally, the chapter contains notes regarding use of macros and restrictions on block-mode terminals.
- Chapter 4 defines the Curses functional interfaces.
- Chapter 5 defines the contents of headers which declare constants, macros, and data structures that are needed by programs using the services provided by Chapter 4.
- Chapter 6 discusses the *terminfo* database which Curses uses to describe terminals. The chapter specifies the source format of a *terminfo* entry using a formal grammar, an informal discussion, and an example. Boolean, numeric, and string capabilities are presented in tabular form.
- Appendix A discusses the use of these capabilities by the writer of a *terminfo* entry to describe the characteristics of the terminal in use.
- The chapters are followed by a glossary, which contains normative definitions of terms used in the document.

### **1.3 More Information**

For more information on the Single UNIX Specification, Version 3 see <http://www.unix.org/version3/>.

