

X/Open CAE Specification

X/Open Curses, Issue 4, Version 2

X/Open Company Ltd.



© July 1996, X/Open Company Limited

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.

Portions of this document are derived from copyrighted material owned by Hewlett-Packard Company, International Business Machines Corporation, Novell Inc., The Open Software Foundation, and Sun Microsystems, Inc.

X/Open CAE Specification

X/Open Curses, Issue 4, Version 2

ISBN: 1-85912-171-3

X/Open Document Number: C610

Published by X/Open Company Ltd., U.K.

Any comments relating to the material contained in this document may be submitted to X/Open at:

X/Open Company Limited
Apex Plaza
Forbury Road
Reading
Berkshire, RG1 1AX
United Kingdom

or by Electronic Mail to:

XoSpecs@xopen.org

Contents

Chapter 1	Introduction.....	1
1.1	This Document	1
1.1.1	Relationship to Issue 3	1
1.1.2	New Features.....	1
1.1.3	To Be Withdrawn	2
1.1.4	Withdrawn.....	3
1.2	Conformance	3
1.2.1	Base Curses Conformance	3
1.2.2	Enhanced Curses Conformance.....	3
1.3	Terminology.....	4
1.3.1	Shaded Text.....	5
1.4	Format of Entries.....	6
 Chapter 2	 Use and Implementation of Interfaces.....	 7
2.1	C Language Definition.....	7
2.2	The Compilation Environment.....	8
2.2.1	The X/Open Name Space (ENHANCED CURSES)	9
2.2.2	Interfaces Implemented as Macros (ENHANCED CURSES).....	11
2.3	Relationship to the XSH Specification.....	11
2.3.1	Error Numbers.....	11
2.4	Data Types.....	12
 Chapter 3	 Interface Overview.....	 13
3.1	Components.....	13
3.2	Screens, Windows and Terminals	14
3.3	Characters.....	16
3.3.1	Character Storage Size	16
3.3.2	Multi-column Characters	16
3.3.3	Attributes.....	16
3.3.4	Rendition	17
3.3.5	Non-spacing Characters	17
3.3.6	Window Properties.....	18
3.4	Conceptual Operations.....	19
3.4.1	Screen Addressing.....	19
3.4.2	Basic Character Operations	19
3.4.3	Special Characters.....	21
3.4.4	Rendition of Characters Placed into a Window	22
3.5	Input Processing.....	23
3.5.1	Keypad Processing	23
3.5.2	Input Mode.....	24
3.5.3	Delay Mode.....	25
3.5.4	Echo Processing.....	25

3.6	The Set of Curses Functions	26
3.6.1	Function Name Conventions	26
3.6.2	Function Families Provided	27
3.7	Interfaces Implemented as Macros	29
3.8	Initialised Curses Environment	29
3.9	Synchronous and Networked Asynchronous Terminals	30
Chapter 4	Curses Interfaces	31
	<i>addch()</i>	32
	<i>addchstr()</i>	33
	<i>addnstr()</i>	34
	<i>addnwstr()</i>	35
	<i>add_wch()</i>	36
	<i>add_wchnstr()</i>	37
	<i>attroff()</i>	38
	<i>attr_get()</i>	39
	<i>baudrate()</i>	41
	<i>beep()</i>	42
	<i>bkgd()</i>	43
	<i>bkgrnd()</i>	44
	<i>border()</i>	45
	<i>border_set()</i>	46
	<i>box()</i>	47
	<i>box_set()</i>	48
	<i>can_change_color()</i>	49
	<i>cbreak()</i>	52
	<i>chgat()</i>	53
	<i>clear()</i>	54
	<i>clearok()</i>	55
	<i>clrtobot()</i>	57
	<i>clrtoeol()</i>	58
	<i>color_content()</i>	59
	<i>COLOR_PAIRS</i>	60
	<i>COLS</i>	61
	<i>copywin()</i>	62
	<i>curscr</i>	63
	<i>curs_set()</i>	64
	<i>cur_term</i>	65
	<i>def_prog_mode()</i>	66
	<i>delay_output()</i>	67
	<i>delch()</i>	68
	<i>del_curterm()</i>	69
	<i>deleteln()</i>	71
	<i>delscreen()</i>	72
	<i>delwin()</i>	73
	<i>derwin()</i>	74
	<i>doupdate()</i>	75
	<i>dupwin()</i>	76

<i>echo()</i>	77
<i>echochar()</i>	78
<i>echo_wchar()</i>	79
<i>endwin()</i>	80
<i>erase()</i>	81
<i>erasechar()</i>	82
<i>filter()</i>	83
<i>flash()</i>	84
<i>flushinp()</i>	85
<i>getbegyx()</i>	86
<i>getbkgd()</i>	88
<i>getbkgrnd()</i>	89
<i>getcchar()</i>	90
<i>getch()</i>	91
<i>getmaxyx()</i>	93
<i>getnstr()</i>	94
<i>getn_wstr()</i>	96
<i>getparyx()</i>	98
<i>getstr()</i>	99
<i>get_wch()</i>	100
<i>getwin()</i>	101
<i>get_wstr()</i>	102
<i>getyx()</i>	103
<i>halfdelay()</i>	104
<i>has_colors()</i>	105
<i>has_ic()</i>	106
<i>hline()</i>	107
<i>hline_set()</i>	108
<i>idcok()</i>	109
<i>idlok()</i>	110
<i>immedok()</i>	111
<i>inch()</i>	112
<i>inchnstr()</i>	113
<i>init_color()</i>	114
<i>initscr()</i>	115
<i>innstr()</i>	117
<i>innwstr()</i>	118
<i>insch()</i>	119
<i>insdelln()</i>	120
<i>insertln()</i>	121
<i>insnstr()</i>	122
<i>ins_nwstr()</i>	123
<i>insstr()</i>	124
<i>instr()</i>	125
<i>ins_wch()</i>	126
<i>ins_wstr()</i>	127
<i>intrflush()</i>	128
<i>in_wch()</i>	129

<i>in_wchnstr()</i>	130
<i>inwstr()</i>	131
<i>isendwin()</i>	132
<i>is_linetouched()</i>	133
<i>keyname()</i>	134
<i>keypad()</i>	135
<i>killchar()</i>	136
<i>leaveok()</i>	137
<i>LINES</i>	138
<i>longname()</i>	139
<i>meta()</i>	140
<i>move()</i>	141
<i>mv</i>	142
<i>mvcur()</i>	144
<i>mvderwin()</i>	145
<i>mvprintw()</i>	146
<i>mvscanw()</i>	147
<i>mvwin()</i>	148
<i>napms()</i>	149
<i>newpad()</i>	150
<i>newterm()</i>	152
<i>newwin()</i>	153
<i>nl()</i>	154
<i>no</i>	155
<i>nodelay()</i>	156
<i>noqiflush()</i>	157
<i>notimeout()</i>	158
<i>overlay()</i>	159
<i>pair_content()</i>	160
<i>pechochar()</i>	161
<i>pnoutrefresh()</i>	162
<i>printw()</i>	163
<i>putp()</i>	164
<i>putwin()</i>	165
<i>qiflush()</i>	166
<i>raw()</i>	167
<i>redrawwin()</i>	168
<i>refresh()</i>	169
<i>reset_prog_mode()</i>	170
<i>resetty()</i>	171
<i>restartterm()</i>	172
<i>ripoffline()</i>	173
<i>savetty()</i>	174
<i>scanw()</i>	175
<i>scr_dump()</i>	176
<i>scr1()</i>	177
<i>scrollok()</i>	178
<i>setcchar()</i>	179

	<i>set_curterm()</i>	180
	<i>setscrreg()</i>	181
	<i>set_term()</i>	182
	<i>setupterm()</i>	183
	<i>slk_attroff()</i>	184
	<i>standend()</i>	186
	<i>start_color()</i>	187
	<i>stdscr</i>	188
	<i>subpad()</i>	189
	<i>subwin()</i>	190
	<i>syncok()</i>	191
	<i>termattrs()</i>	192
	<i>termname()</i>	193
	<i>tgetent()</i>	194
	<i>tigetflag()</i>	196
	<i>timeout()</i>	198
	<i>touchline()</i>	199
	<i>tparm()</i>	200
	<i>tputs()</i>	201
	<i>typeahead()</i>	202
	<i>unctrl()</i>	203
	<i>ungetch()</i>	204
	<i>untouchwin()</i>	205
	<i>use_env()</i>	206
	<i>vidattr()</i>	207
	<i>vline()</i>	209
	<i>vline_set()</i>	210
	<i>vwprintw()</i>	211
	<i>vw_printw()</i>	212
	<i>vwscanw()</i>	213
	<i>vw_scanw()</i>	214
	<i>w</i>	215
	<i>wunctrl()</i>	218
Chapter 5	Headers.....	219
	<curses.h>	220
	<term.h>	235
	<unctrl.h>	236
Chapter 6	Terminfo Source Format (ENHANCED CURSES).....	237
6.1	Source File Syntax.....	238
6.1.1	Minimum Guaranteed Limits.....	239
6.1.2	Formal Grammar.....	239
6.1.3	Defined Capabilities.....	241
6.1.4	Sample Entry.....	251
6.1.5	Types of Capabilities in the Sample Entry.....	251

Appendix A	Application Usage	255
A.1	Device Capabilities	255
A.1.1	Basic Capabilities	255
A.1.2	Parameterised Strings	256
A.1.3	Cursor Motions	257
A.1.4	Area Clears	258
A.1.5	Insert/Delete Line.....	258
A.1.6	Insert/Delete Character.....	259
A.1.7	Highlighting, Underlining and Visible Bells.....	260
A.1.8	Keypad.....	262
A.1.9	Tabs and Initialisation.....	262
A.1.10	Delays.....	263
A.1.11	Status Lines	263
A.1.12	Line Graphics.....	264
A.1.13	Colour Manipulation	265
A.1.14	Miscellaneous	266
A.1.15	Special Cases.....	267
A.1.16	Similar Terminals	268
A.2	Printer Capabilities	268
A.2.1	Rounding Values	268
A.2.2	Printer Resolution	268
A.2.3	Specifying Printer Resolution.....	269
A.2.4	Capabilities that Cause Movement.....	271
A.2.5	Alternate Character Sets.....	275
A.2.6	Dot-Matrix Graphics	276
A.2.7	Effect of Changing Printing Resolution.....	277
A.2.8	Print Quality.....	278
A.2.9	Printing Rate and Buffer Size.....	278
A.3	Selecting a Terminal	279
A.4	Application Usage	279
A.4.1	Conventions for Device Aliases.....	279
A.4.2	Variations of Terminal Definitions	280
	Glossary	281
	Index.....	283



Preface

X/Open

X/Open is an independent, worldwide, open systems organisation supported by most of the world's largest information systems suppliers, user organisations and software companies. Its mission is to bring to users greater value from computing, through the practical implementation of open systems.

X/Open's strategy for achieving this goal is to combine existing and emerging standards into a comprehensive, integrated, high-value and usable open system environment, called the Common Applications Environment (CAE). This environment covers the standards, above the hardware level, that are needed to support open systems. It provides for portability and interoperability of applications, and so protects investment in existing software while enabling additions and enhancements. It also allows users to move between systems with a minimum of retraining.

X/Open defines this CAE in a set of specifications which include an evolving portfolio of application programming interfaces (APIs) which significantly enhance portability of application programs at the source code level, along with definitions of and references to protocols and protocol profiles which significantly enhance the interoperability of applications and systems.

The X/Open CAE is implemented in real products and recognised by a distinctive trade mark — the X/Open brand — that is licensed by X/Open and may be used on products which have demonstrated their conformance.

X/Open Technical Publications

X/Open publishes a wide range of technical literature, the main part of which is focussed on specification development, but which also includes Guides, Snapshots, Technical Studies, Branding/Testing documents, industry surveys, and business titles.

There are two types of X/Open specification:

- *CAE Specifications*

CAE (Common Applications Environment) specifications are the stable specifications that form the basis for X/Open-branded products. These specifications are intended to be used widely within the industry for product development and procurement purposes.

Anyone developing products that implement an X/Open CAE specification can enjoy the benefits of a single, widely supported standard. In addition, they can demonstrate compliance with the majority of X/Open CAE specifications once these specifications are referenced in an X/Open component or profile definition and included in the X/Open branding programme.

CAE specifications are published as soon as they are developed, not published to coincide with the launch of a particular X/Open brand. By making its specifications available in this way, X/Open makes it possible for conformant products to be developed as soon as is practicable, so enhancing the value of the X/Open brand as a procurement aid to users.

- *Preliminary Specifications*

These specifications, which often address an emerging area of technology and consequently are not yet supported by multiple sources of stable conformant implementations, are released in a controlled manner for the purpose of validation through implementation of products. A Preliminary specification is not a draft specification. In fact, it is as stable as X/Open can make it, and on publication has gone through the same rigorous X/Open development and review procedures as a CAE specification.

Preliminary specifications are analogous to the *trial-use* standards issued by formal standards organisations, and product development teams are encouraged to develop products on the basis of them. However, because of the nature of the technology that a Preliminary specification is addressing, it may be untried in multiple independent implementations, and may therefore change before being published as a CAE specification. There is always the intent to progress to a corresponding CAE specification, but the ability to do so depends on consensus among X/Open members. In all cases, any resulting CAE specification is made as upwards-compatible as possible. However, complete upwards-compatibility from the Preliminary to the CAE specification cannot be guaranteed.

In addition, X/Open publishes:

- *Guides*

These provide information that X/Open believes is useful in the evaluation, procurement, development or management of open systems, particularly those that are X/Open-compliant. X/Open Guides are advisory, not normative, and should not be referenced for purposes of specifying or claiming X/Open conformance.

- *Technical Studies*

X/Open Technical Studies present results of analyses performed by X/Open on subjects of interest in areas relevant to X/Open's Technical Programme. They are intended to communicate the findings to the outside world and, where appropriate, stimulate discussion and actions by other bodies and the industry in general.

- *Snapshots*

These provide a mechanism for X/Open to disseminate information on its current direction and thinking, in advance of possible development of a Specification, Guide or Technical Study. The intention is to stimulate industry debate and prototyping, and solicit feedback. A Snapshot represents the interim results of an X/Open technical activity. Although at the time of its publication, there may be an intention to progress the activity towards publication of a Specification, Guide or Technical Study, X/Open is a consensus organisation, and makes no commitment regarding future development and further publication. Similarly, a Snapshot does not represent any commitment by X/Open members to develop any specific products.

Versions and Issues of Specifications

As with all *live* documents, CAE Specifications require revision, in this case as the subject technology develops and to align with emerging associated international standards. X/Open makes a distinction between revised specifications which are fully backward compatible and those which are not:

- a new *Version* indicates that this publication includes all the same (unchanged) definitive information from the previous publication of that title, but also includes extensions or additional information. As such, it *replaces* the previous publication.

- a new *Issue* does include changes to the definitive information contained in the previous publication of that title (and may also include extensions or additional information). As such, X/Open maintains *both* the previous and new issue as current publications.

Corrigenda

Most X/Open publications deal with technology at the leading edge of open systems development. Feedback from implementation experience gained from using these publications occasionally uncovers errors or inconsistencies. Significant errors or recommended solutions to reported problems are communicated by means of Corrigenda.

The reader of this document is advised to check periodically if any Corrigenda apply to this publication. This may be done in any one of the following ways:

- anonymous ftp to ftp.xopen.org
- ftpmail (see below)
- reference to the Corrigenda list in the latest X/Open Publications Price List.

To request Corrigenda information using ftpmail, send a message to ftpmail@xopen.org with the following four lines in the body of the message:

```
open
cd pub/Corrigenda
get index
quit
```

This will return the index of publications for which Corrigenda exist. Use the same email address to request a copy of the full corrigendum information following the email instructions.

This Document

This specification is a CAE Specification (see above) that defines the X/Open Curses interface offered to application programs by X/Open Curses conformant systems. Readers are expected to be experienced C language programmers and to be familiar with the **XBD** specification. This specification 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 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 on page 237 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 on page 255 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. Comprehensive references are available in the index.

Typographical Conventions

The following typographical conventions are used throughout this document:

- **Bold** font is used in text for options to commands, filenames, keywords, type names, data structures and their members.
- *Italic* strings are used for emphasis or to identify the first instance of a word requiring definition. Italics in text also denote:
 - command operands, command option-arguments or variable names, for example, substitutable argument prototypes
 - environment variables, which are also shown in capitals
 - utility names
 - external variables, such as *errno*
 - functions; these are shown as follows: *name()*; names without parentheses are C external variables, C function family names, utility names, command operands or command option-arguments.
- Normal font is used for the names of constants and literals.
- The notation **<file.h>** indicates a header file.
- Names surrounded by braces, for example, {ARG_MAX}, represent symbolic limits or configuration values which may be declared in appropriate headers by means of the C **#define** construct.
- The notation [EABCD] is used to identify an error value EABCD.
- Syntax, code examples and user input in interactive examples are shown in fixed width font. Brackets shown in this font, [], are part of the syntax and do *not* indicate optional items. In syntax the | symbol is used to separate alternatives, and ellipses (. . .) are used to show that additional arguments are optional.
- **Bold fixed width** font is used to identify brackets that surround optional items in syntax, [], and to identify system output in interactive examples.
- Variables within syntax statements are shown in *italic fixed width font*.
- Ranges of values are indicated with parentheses or brackets as follows:
 - (a,b) means the range of all values from a to b, including neither a nor b
 - [a,b] means the range of all values from a to b, including a and b

- [a,b) means the range of all values from a to b, including a, but not b
- (a,b] means the range of all values from a to b, including b, but not a
- Shading is used to identify X/Open Enhanced Curses material, relating to interfaces included to provide enhanced capabilities for applications originally written to be compiled on systems based on the UNIX operating system. Therefore, the features described may not be present on systems that conform to XPG4 or to earlier XPG releases. The relevant reference manual pages may provide additional or more specific portability warnings about use of the material.

If an entire **SYNOPSIS** section is shaded and marked with one EC, all the functionality described in that entry is an extension.

The material on pages labelled ENHANCED CURSES and the material flagged with the EC margin legend is available only in cases where the `_XOPEN_CURSES` version test macro is defined.

Notes:

1. Symbolic limits are used in this document instead of fixed values for portability. The values of most of these constants are defined in `<limits.h>` or `<unistd.h>`.
2. The values of errors are defined in `<errno.h>`.

Trade Marks

AT&T[®] is a registered trade mark of AT&T in the U.S.A. and other countries.

HP[®] is a registered trade mark of Hewlett-Packard.

UNIX[®] is a registered trade mark in the United States and other countries, licensed exclusively through X/Open Company Limited.

X/Open[®] is a registered trade mark, and the “X” device is a trade mark, of X/Open Company Limited.

The names of terminals and of terminal manufacturers cited as examples in Chapter 6 and Appendix A may be trade marks, which are the property of their respective owners.

Acknowledgements

X/Open gratefully acknowledges:

- Novell, Inc. for permission to reproduce portions of its copyrighted System V Interface Definition (SVID) and material from the UNIX System V Release 4.2 documentation.
- Hewlett-Packard Company, International Business Machines Corporation, Novell Inc., The Open Software Foundation, and Sun Microsystems, Inc., for their work in developing the X/Open UNIX Extension and sponsoring it through the X/Open Direct Review (Fast-track) process.

Referenced Documents

The following documents are referenced in this specification:

ANSI C

American National Standard for Information Systems: Standard X3.159-1989, Programming Language C. TO BE COMPLETED.

ISO 8859-1

ISO 8859-1: 1987, Information Processing — 8-bit Single-byte Coded Graphic Character Sets — Part 1: Latin Alphabet No. 1.

ISO/IEC 646

ISO/IEC 646: 1991, Information Processing — ISO 7-bit Coded Character Set for Information Interchange.

ISO C

ISO/IEC 9899: 1990: Programming Languages — C, including:
Technical Corrigendum 1: 1994.

Amendment 1: 1994, Multibyte Support Extensions (MSE) for ISO C.

Amendment 1: 1995, C Integrity.

SVID Issue 2

System V Interface Definition (Spring 1986 - Issue 2).

SVID 3rd Edition

System Interface Definitions (1989 - 3rd Edition).

System V Release 2.0

— UNIX System V Release 2.0 Programmer's Reference Manual (April 1984 - Issue 2).

— UNIX System V Release 2.0 Programming Guide (April 1984 - Issue 2).

System V Release 4.2

Operating System API Reference, UNIX® SVR4.2 (1992) (ISBN: 0-13-017658-3).

The following X/Open documents are referenced in this specification.

Curses Interface, Issue 3

X/Open Specification, February 1992, Supplementary Definitions, Issue 3 (ISBN: 1-872630-38-3, C213), Chapters 9 to 14 inclusive, Curses Interface; this specification was formerly X/Open Portability Guide, Issue 3, Volume 3, January 1989, XSI Supplementary Definitions (ISBN: 0-13-685850-3, XO/XPG/89/004).

Curses Interface, Issue 4

X/Open CAE Specification, December 1994, X/Open Curses, Issue 4 (ISBN: 1-85912-077-6, C437).

Curses Interface, Issue 4, Version 2

X/Open CAE Specification, May 1996, X/Open Curses, Issue 4, Version 2 (ISBN: 1-85912-171-3, C610). (This document.)

Headers Interface

X/Open Specification, February 1992, Supplementary Definitions, Issue 3 (ISBN: 1-872630-38-3, C213), Chapter 19, Cpio and Tar Headers; this specification was formerly X/Open Portability Guide Issue 3, Volume 3, January 1989, XSI Supplementary

Definitions (ISBN: 0-13-685850-3, XO/XPG/89/004).

XBD, Issue 4, Version 2

X/Open CAE Specification, August 1994, System Interface Definitions, Issue 4, Version 2 (ISBN: 1-85912-036-9, C434).

XCU, Issue 4, Version 2

X/Open CAE Specification, August 1994, Commands and Utilities, Issue 4, Version 2 (ISBN: 1-85912-034-2, C436).

XSH, Issue 3

X/Open Specification, February 1992, System Interfaces and Headers, Issue 3 (ISBN: 1-872630-37-5, C212); this specification was formerly X/Open Portability Guide, Issue 3, Volume 2, January 1989, XSI System Interface and Headers (ISBN: 0-13-685843-0, XO/XPG/89/003).

XSH, Issue 4

X/Open CAE Specification, July 1992, System Interfaces and Headers, Issue 4 (ISBN: 1-872630-47-2, C202).

XSH, Issue 4, Version 2

X/Open CAE Specification, August 1994, System Interfaces and Headers, Issue 4, Version 2 (ISBN: 1-85912-037-7, C435).