ISO/IEC JTC 1/SC 22/OWG Linux N002

Date: 19 September 2017

ISO/IEC 23360-1-2

Edition 1

ISO/IEC JTC 1/SC 22/OWG LSB

Secretariat: ANSI

Information Technology — Operating systems – Linux Standard Base (LSB) core specification generic part

Document type: International standard

Document subtype: if applicable

Document stage: (10) development stage

Document language: E

*Élément introductif — Élément principal — Partie n: Titre de la partie*

Warning

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

**Copyright notice**

This ISO document is a working draft or committee draft and is copyright-protected by ISO. While the reproduction of working drafts or committee drafts in any form for use by participants in the ISO standards development process is permitted without prior permission from ISO, neither this document nor any extract from it may be reproduced, stored or transmitted in any form for any other purpose without prior written permission from ISO.

Requests for permission to reproduce this document for the purpose of selling it should be addressed as shown below or to ISO’s member body in the country of the requester:

*ISO copyright office*

*Case postale 56, CH-1211 Geneva 20*

*Tel. + 41 22 749 01 11*

*Fax + 41 22 749 09 47*

*E-mail copyright@iso.org*

*Web www.iso.org*

Reproduction for sales purposes may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

**Linux Foundation Copyright**

This specification is published under the terms of the GNU Free Documentation License, Version 1.1, March 2000

Copyright (C) 2000 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

See Annex A for the complete GNU Free Documentation License. GNU

**Linux Standard Base Core Specification, Generic Part**

LSB Core - Generic 5.0

Copyright © 2015 Linux Foundation

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Portions of the text may be copyrighted by the following parties:

• The Regents of the University of California

• Free Software Foundation

• Ian F. Darwin

• Paul Vixie

• BSDI (now Wind River)

• Jean-loup Gailly and Mark Adler

• Massachusetts Institute of Technology

• Apple Inc.

• Easy Software Products

• artofcode LLC

• Till Kamppeter

• Manfred Wassman

• Python Software Foundation

These excerpts are being used in accordance with their respective licenses.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

UNIX is a registered trademark of The Open Group.

LSB is a trademark of the Linux Foundation in the United States and other countries.

AMD is a trademark of Advanced Micro Devices, Inc.

Intel and Itanium are registered trademarks and Intel386 is a trademark of Intel Corporation.

PowerPC is a registered trademark and PowerPC Architecture is a trademark of the IBM Corporation.

S/390 is a registered trademark of the IBM Corporation.

OpenGL is a registered trademark of Silicon Graphics, Inc.

PAM documentation is Copyright (C) Andrew G. Morgan 1996-9. All rights reserved. Used under the following conditions:

1. Redistributions of source code must retain the above copyright notice, and the entire permission notice in its entirety, including the disclaimer of warranties.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission

## **Contents**

[**Contents** iv](#_Toc494808119)

[**List of Figures** xviii](#_Toc494808120)

[**Foreword** xx](#_Toc494808121)

[**Status of this Document** xxi](#_Toc494808122)

[**Introduction** xxii](#_Toc494808123)

[**I Introductory Elements** 1](#_Toc494808124)

[**1 Scope** 1](#_Toc494808125)

[**1.1 General** 1](#_Toc494808126)

[**1.2 Module Specific Scope** 2](#_Toc494808127)

[**2 References** 3](#_Toc494808128)

[**2.1 Normative References** 3](#_Toc494808129)

[**2.2 Informative References/Bibliography** 4](#_Toc494808130)

[**3 Requirements** 7](#_Toc494808131)

[**3.1 Relevant Libraries** 7](#_Toc494808132)

[**3.2 LSB Implementation Conformance** 7](#_Toc494808133)

[**3.3 LSB Application Conformance** 8](#_Toc494808134)

[**4 Terms and Definitions** 10](#_Toc494808135)

[**5 Documentation Conventions** 11](#_Toc494808136)

[**6 Relationship To ISO/IEC 9945 POSIX** 12](#_Toc494808137)

[**7 Relationship To Other Linux Foundation Specifications** 13](#_Toc494808138)

[**II Executable And Linking Format (ELF)** 14](#_Toc494808139)

[**8 Introduction** 15](#_Toc494808140)

[**9 Low Level System Information** 16](#_Toc494808141)

[**9.1 Operating System Interface** 16](#_Toc494808142)

[**9.2 Machine Interface** 16](#_Toc494808143)

[**9.2.1 Data Representation** 16](#_Toc494808144)

[**10 Object Format** 17](#_Toc494808145)

[**10.1 Object Files** 17](#_Toc494808146)

[**10.2 Sections** 17](#_Toc494808147)

[**10.2.1 Introduction** 17](#_Toc494808148)

[**10.2.2 Sections Types** 17](#_Toc494808149)

[**10.3 Special Sections** 20](#_Toc494808150)

[**10.3.1 Special Sections** 20](#_Toc494808151)

[**10.4 Symbol Mapping** 25](#_Toc494808152)

[**10.4.1 Introduction** 25](#_Toc494808153)

[**10.5 DWARF Extensions** 25](#_Toc494808154)

[**10.5.2 DWARF Exception Header Encoding** 25](#_Toc494808155)

[**10.5.3 DWARF CFI Extensions** 27](#_Toc494808156)

[**10.6 Exception Frames** 27](#_Toc494808157)

[**10.6.2 The .eh\_frame section** 27](#_Toc494808158)

[**10.6.2 The .eh\_frame\_hdr section** 31](#_Toc494808159)

[**10.7 Symbol Versioning** 32](#_Toc494808160)

[**10.7.1 Introduction** 32](#_Toc494808161)

[**10.7.2 Symbol Version Table** 32](#_Toc494808162)

[**10.7.3 Version Definitions** 32](#_Toc494808163)

[**10.7.4 Version Requirements** 33](#_Toc494808164)

[**10.7.5 Startup Sequence** 35](#_Toc494808165)

[**10.7.6 Symbol Resolution** 35](#_Toc494808166)

[**10.8 ABI note tag** 36](#_Toc494808167)

[**11 Dynamic Linking** 37](#_Toc494808168)

[**11.1 Program Loading and Dynamic Linking** 37](#_Toc494808169)

[**11.2 Program Header** 37](#_Toc494808170)

[**11.3 Dynamic Entries** 37](#_Toc494808171)

[**11.3.1 Introduction** 37](#_Toc494808172)

[**11.3.2 Dynamic Entries** 38](#_Toc494808173)

[**12 C++ Class Representations** 42](#_Toc494808174)

[**12.1 C++ Data Representation** 42](#_Toc494808175)

[**12.1.1 Class Representation** 42](#_Toc494808176)

[**13 Symbol Mapping** 46](#_Toc494808177)

[**13.1 Symbol Mapping** 46](#_Toc494808178)

[**13.1.1 C++ Language** 46](#_Toc494808179)

[**III Base Libraries** 47](#_Toc494808180)

[**14 Base Libraries** 48](#_Toc494808181)

[**14.1 Introduction** 48](#_Toc494808182)

[**14.2 Program Interpreter** 48](#_Toc494808183)

[**14.3 Interfaces for libc** 48](#_Toc494808184)

[**14.3.1 RPC** 49](#_Toc494808185)

[**14.3.2 Epoll** 50](#_Toc494808186)

[**14.3.3 System Calls** 50](#_Toc494808187)

[**14.3.4 Standard I/O** 52](#_Toc494808188)

[**14.3.5 Signal Handling** 54](#_Toc494808189)

[**14.3.6 Localization Functions** 54](#_Toc494808190)

[**14.3.7 Posix Spawn Option** 55](#_Toc494808191)

[**14.3.8 Posix Advisory Option** 56](#_Toc494808192)

[**14.3.9 Socket Interface** 56](#_Toc494808193)

[**14.3.10 Wide Characters** 56](#_Toc494808194)

[**14.3.11 String Functions** 58](#_Toc494808195)

[**14.3.12 IPC Functions** 59](#_Toc494808196)

[**14.3.13 Regular Expressions** 60](#_Toc494808197)

[**14.3.14 Character Type Functions** 60](#_Toc494808198)

[**14.3.15 Time Manipulation** 61](#_Toc494808199)

[**14.3.16 Terminal Interface Functions** 61](#_Toc494808200)

[**14.3.17 System Database Interface** 61](#_Toc494808201)

[**14.3.18 Language Support** 62](#_Toc494808202)

[**14.3.19 Large File Support** 63](#_Toc494808203)

[**14.3.20 Inotify** 63](#_Toc494808204)

[**14.3.21 Standard Library** 64](#_Toc494808205)

[**14.3.22 GNU Extensions for libc** 66](#_Toc494808206)

[**14.4 Data Definitions for libc** 66](#_Toc494808207)

[**14.4.1 argz.h** 67](#_Toc494808208)

[**14.4.2 arpa/inet.h** 67](#_Toc494808209)

[**14.4.3 assert.h** 67](#_Toc494808210)

[**14.4.4 cpio.h** 68](#_Toc494808211)

[**14.4.5 ctype.h** 68](#_Toc494808212)

[**14.4.6 dirent.h** 69](#_Toc494808213)

[**14.4.7 elf.h** 69](#_Toc494808214)

[**14.4.8 endian.h** 83](#_Toc494808215)

[**14.4.9 envz.h** 84](#_Toc494808216)

[**14.4.10 err.h** 84](#_Toc494808217)

[**14.4.11 errno.h** 84](#_Toc494808218)

[**14.4.12 error.h** 86](#_Toc494808219)

[**14.4.13 execinfo.h** 87](#_Toc494808220)

[**14.4.14 fcntl.h** 87](#_Toc494808221)

[**14.4.15 fmtmsg.h** 88](#_Toc494808222)

[**14.4.16 fnmatch.h** 89](#_Toc494808223)

[**14.4.17 ftw.h** 89](#_Toc494808224)

[**14.4.18 getopt.h** 90](#_Toc494808225)

[**14.4.19 glob.h** 91](#_Toc494808226)

[**14.4.20 gnu/libc-version.h** 91](#_Toc494808227)

[**14.4.21 grp.h** 91](#_Toc494808228)

[**14.4.22 iconv.h** 92](#_Toc494808229)

[**14.4.23 ifaddrs.h** 92](#_Toc494808230)

[**14.4.24 inttypes.h** 92](#_Toc494808231)

[**14.4.25 langinfo.h** 95](#_Toc494808232)

[**14.4.26 libgen.h** 96](#_Toc494808233)

[**14.4.27 libintl.h** 96](#_Toc494808234)

[**14.4.28 limits.h** 97](#_Toc494808235)

[**14.4.29 link.h** 98](#_Toc494808236)

[**14.4.30 locale.h** 98](#_Toc494808237)

[**14.4.31 lsb/time.h** 100](#_Toc494808238)

[**14.4.32 lsb/types.h** 100](#_Toc494808239)

[**14.4.33 lsb/wchar.h** 100](#_Toc494808240)

[**14.4.34 monetary.h** 100](#_Toc494808241)

[**14.4.35 net/if.h** 100](#_Toc494808242)

[**14.4.36 netdb.h** 102](#_Toc494808243)

[**14.4.37 netinet/icmp6.h** 104](#_Toc494808244)

[**14.4.38 netinet/igmp.h** 107](#_Toc494808245)

[**14.4.39 netinet/in.h** 108](#_Toc494808246)

[**14.4.40 netinet/in\_systm.h** 110](#_Toc494808247)

[**14.4.41 netinet/ip.h** 110](#_Toc494808248)

[**14.4.42 netinet/ip6.h** 111](#_Toc494808249)

[**14.4.43 netinet/ip\_icmp.h** 112](#_Toc494808250)

[**14.4.44 netinet/tcp.h** 115](#_Toc494808251)

[**14.4.45 netinet/udp.h** 116](#_Toc494808252)

[**14.4.46 nl\_types.h** 116](#_Toc494808253)

[**14.4.47 poll.h** 117](#_Toc494808254)

[**14.4.48 pwd.h** 117](#_Toc494808255)

[**14.4.49 regex.h** 117](#_Toc494808256)

[**14.4.50 rpc/auth.h** 118](#_Toc494808257)

[**14.4.51 rpc/clnt.h** 119](#_Toc494808258)

[**14.4.52 rpc/pmap\_clnt.h** 121](#_Toc494808259)

[**14.4.53 rpc/rpc\_msg.h** 121](#_Toc494808260)

[**14.4.54 rpc/svc.h** 123](#_Toc494808261)

[**14.4.55 rpc/types.h** 124](#_Toc494808262)

[**14.4.56 rpc/xdr.h** 124](#_Toc494808263)

[**14.4.57 sched.h** 126](#_Toc494808264)

[**14.4.58 search.h** 128](#_Toc494808265)

[**14.4.59 setjmp.h** 128](#_Toc494808266)

[**14.4.60 signal.h** 129](#_Toc494808267)

[**14.4.61 spawn.h** 133](#_Toc494808268)

[**14.4.62 stddef.h** 135](#_Toc494808269)

[**14.4.63 stdint.h** 135](#_Toc494808270)

[**14.4.64 stdio.h** 136](#_Toc494808271)

[**14.4.65 stdlib.h** 138](#_Toc494808272)

[**14.4.66 string.h** 141](#_Toc494808273)

[**14.4.67 strings.h** 142](#_Toc494808274)

[**14.4.68 sys/epoll.h** 142](#_Toc494808275)

[**14.4.69 sys/file.h** 143](#_Toc494808276)

[**14.4.70 sys/inotify.h** 143](#_Toc494808277)

[**14.4.71 sys/ioctl.h** 144](#_Toc494808278)

[**14.4.72 sys/ipc.h** 145](#_Toc494808279)

[**14.4.73 sys/mman.h** 145](#_Toc494808280)

[**14.4.74 sys/msg.h** 145](#_Toc494808281)

[**14.4.75 sys/param.h** 146](#_Toc494808282)

[**14.4.76 sys/poll.h** 146](#_Toc494808283)

[**14.4.77 sys/ptrace.h** 146](#_Toc494808284)

[**14.4.78 sys/resource.h** 146](#_Toc494808285)

[**14.4.79 sys/select.h** 148](#_Toc494808286)

[**14.4.80 sys/sem.h** 148](#_Toc494808287)

[**14.4.81 sys/sendfile.h** 149](#_Toc494808288)

[**14.4.82 sys/shm.h** 149](#_Toc494808289)

[**14.4.83 sys/socket.h** 149](#_Toc494808290)

[**14.4.84 sys/stat.h** 152](#_Toc494808291)

[**14.4.85 sys/statfs.h** 153](#_Toc494808292)

[**14.4.86 sys/statvfs.h** 153](#_Toc494808293)

[**14.4.87 sys/sysinfo.h** 153](#_Toc494808294)

[**14.4.88 sys/time.h** 154](#_Toc494808295)

[**14.4.89 sys/timeb.h** 154](#_Toc494808296)

[**14.4.90 sys/times.h** 154](#_Toc494808297)

[**14.4.91 sys/types.h** 155](#_Toc494808298)

[**14.4.92 sys/uio.h** 155](#_Toc494808299)

[**14.4.93 sys/un.h** 156](#_Toc494808300)

[**14.4.94 sys/utsname.h** 156](#_Toc494808301)

[**14.4.95 sys/wait.h** 156](#_Toc494808302)

[**14.4.96 sysexits.h** 156](#_Toc494808303)

[**14.4.97 syslog.h** 157](#_Toc494808304)

[**14.4.98 tar.h** 158](#_Toc494808305)

[**14.4.99 termios.h** 158](#_Toc494808306)

[**14.4.100 time.h** 160](#_Toc494808307)

[**14.4.101 ucontext.h** 162](#_Toc494808308)

[**14.4.102 ulimit.h** 162](#_Toc494808309)

[**14.4.103 unistd.h** 162](#_Toc494808310)

[**14.4.104 utime.h** 169](#_Toc494808311)

[**14.4.105 utmp.h** 170](#_Toc494808312)

[**14.4.106 utmpx.h** 170](#_Toc494808313)

[**14.4.107 wchar.h** 170](#_Toc494808314)

[**14.4.108 wctype.h** 173](#_Toc494808315)

[**14.4.109 wordexp.h** 174](#_Toc494808316)

[**14.5 Interface Definitions for libc** 175](#_Toc494808317)

[**14.6 Interfaces for libm** 359](#_Toc494808318)

[**14.6.1 Math** 359](#_Toc494808319)

[**14.7 Data Definitions for libm** 362](#_Toc494808320)

[**14.7.1 complex.h** 362](#_Toc494808321)

[**14.7.2 fenv.h** 363](#_Toc494808322)

[**14.7.3 math.h** 363](#_Toc494808323)

[**14.8 Interface Definitions for libm** 368](#_Toc494808324)

[**14.9 Interfaces for libpthread** 391](#_Toc494808325)

[**14.9.1 Realtime Threads** 391](#_Toc494808326)

[**14.9.2 Advanced Realtime Threads** 392](#_Toc494808327)

[**14.9.3 Posix Threads** 392](#_Toc494808328)

[**14.9.4 Thread aware versions of libc interfaces** 394](#_Toc494808329)

[**14.9.5 GNU Extensions for libpthread** 394](#_Toc494808330)

[**14.9.6 System Calls** 394](#_Toc494808331)

[**14.9.7 Standard I/O** 395](#_Toc494808332)

[**14.9.8 Signal Handling** 395](#_Toc494808333)

[**14.9.9 Standard Library** 395](#_Toc494808334)

[**14.9.10 Socket Interface** 395](#_Toc494808335)

[**14.9.11 Terminal Interface Functions** 396](#_Toc494808336)

[**14.10 Data Definitions for libpthread** 396](#_Toc494808337)

[**14.10.1 lsb/pthread.h** 396](#_Toc494808338)

[**14.10.2 pthread.h** 396](#_Toc494808339)

[**14.10.3 semaphore.h** 401](#_Toc494808340)

[**14.11 Interface Definitions for libpthread** 402](#_Toc494808341)

[**14.12 Interfaces for libgcc\_s** 406](#_Toc494808342)

[**14.12.1 Unwind Library** 406](#_Toc494808343)

[**14.13 Data Definitions for libgcc\_s** 406](#_Toc494808344)

[**14.13.1 unwind.h** 407](#_Toc494808345)

[**14.14 Interface Definitions for libgcc\_s** 408](#_Toc494808346)

[**14.15 Interfaces for libdl** 414](#_Toc494808347)

[**14.15.1 Dynamic Loader** 415](#_Toc494808348)

[**14.16 Data Definitions for libdl** 415](#_Toc494808349)

[**14.16.1 dlfcn.h** 415](#_Toc494808350)

[**14.17 Interface Definitions for libdl** 416](#_Toc494808351)

[**14.18 Interfaces for librt** 419](#_Toc494808352)

[**14.18.1 Shared Memory Objects** 419](#_Toc494808353)

[**14.18.2 Asynchronous I/O** 420](#_Toc494808354)

[**14.18.3 Clock** 420](#_Toc494808355)

[**14.18.4 Timers** 420](#_Toc494808356)

[**14.18.5 Message Queues** 420](#_Toc494808357)

[**14.19 Data Definitions for librt** 421](#_Toc494808358)

[**14.19.1 aio.h** 421](#_Toc494808359)

[**14.19.2 mqueue.h** 422](#_Toc494808360)

[**14.20 Interfaces for libcrypt** 423](#_Toc494808361)

[**14.20.1 Encryption** 423](#_Toc494808362)

[**14.21 Data Definitions for libcrypt** 423](#_Toc494808363)

[**14.21.1 crypt.h** 423](#_Toc494808364)

[**14.22 Interface Definitions for libcrypt** 424](#_Toc494808365)

[**14.23 Interfaces for libpam** 425](#_Toc494808366)

[**14.23.1 Pluggable Authentication API** 426](#_Toc494808367)

[**14.24 Data Definitions for libpam** 426](#_Toc494808368)

[**14.24.1 security/\_pam\_types.h** 427](#_Toc494808369)

[**14.24.2 security/pam\_appl.h** 428](#_Toc494808370)

[**14.24.3 security/pam\_modules.h** 428](#_Toc494808371)

[**14.25 Interface Definitions for libpam** 428](#_Toc494808372)

[**IV Utility Libraries** 441](#_Toc494808373)

[**15 Utility Libraries** 442](#_Toc494808374)

[**15.1 Introduction** 442](#_Toc494808375)

[**15.2 Interfaces for libz** 442](#_Toc494808376)

[**15.2.1 Compression Library** 442](#_Toc494808377)

[**15.3 Data Definitions for libz** 443](#_Toc494808378)

[**15.3.1 zconf.h** 443](#_Toc494808379)

[**15.3.2 zlib.h** 443](#_Toc494808380)

[**15.4 Interface Definitions for libz** 446](#_Toc494808381)

[**15.5 Interfaces for libncurses** 486](#_Toc494808382)

[**15.5.1 Curses** 487](#_Toc494808383)

[**15.6 Data Definitions for libncurses** 491](#_Toc494808384)

[**15.6.1 curses.h** 491](#_Toc494808385)

[**15.6.2 term.h** 499](#_Toc494808386)

[**15.7 Interface Definitions for libncurses** 500](#_Toc494808387)

[**15.8 Interfaces for libncursesw** 507](#_Toc494808388)

[**15.8.1 Curses Wide** 507](#_Toc494808389)

[**15.9 Data Definitions for libncursesw** 512](#_Toc494808390)

[**15.9.1 ncursesw/curses.h** 512](#_Toc494808391)

[**15.9.2 ncursesw/ncurses\_dll.h** 530](#_Toc494808392)

[**15.9.3 ncursesw/term.h** 530](#_Toc494808393)

[**15.9.4 ncursesw/unctrl.h** 539](#_Toc494808394)

[**15.10 Interface Definitions for libncursesw** 539](#_Toc494808395)

[**15.11 Interfaces for libutil** 539](#_Toc494808396)

[**15.11.1 Utility Functions** 539](#_Toc494808397)

[**15.12 Data Definitions for libutil** 539](#_Toc494808398)

[**15.12.1 pty.h** 540](#_Toc494808399)

[**15.13 Interface Definitions for libutil** 540](#_Toc494808400)

[**V C++ Libraries** 546](#_Toc494808401)

[**16 Libraries** 547](#_Toc494808402)

[**16.1 Interfaces for libstdcxx** 547](#_Toc494808403)

[**16.1.1 C++ Runtime Support** 547](#_Toc494808404)

[**16.1.2 C++ type descriptors for built-in types** 549](#_Toc494808405)

[**16.1.3 C++ \_Rb\_tree** 552](#_Toc494808406)

[**16.1.4 Class type\_info** 553](#_Toc494808407)

[**16.1.5 Class \_\_cxxabiv1::\_\_enum\_type\_info** 554](#_Toc494808408)

[**16.1.6 Class \_\_cxxabiv1::\_\_array\_type\_info** 555](#_Toc494808409)

[**16.1.7 Class \_\_cxxabiv1::\_\_class\_type\_info** 556](#_Toc494808410)

[**16.1.8 Class \_\_cxxabiv1::\_\_pbase\_type\_info** 557](#_Toc494808411)

[**16.1.9 Class \_\_cxxabiv1::\_\_pointer\_type\_info** 559](#_Toc494808412)

[**16.1.10 Class \_\_cxxabiv1::\_\_function\_type\_info** 560](#_Toc494808413)

[**16.1.11 Class \_\_cxxabiv1::\_\_si\_class\_type\_info** 561](#_Toc494808414)

[**16.1.12 Class \_\_cxxabiv1::\_\_vmi\_class\_type\_info** 563](#_Toc494808415)

[**16.1.13 Class \_\_cxxabiv1::\_\_fundamental\_type\_info** 564](#_Toc494808416)

[**16.1.14 Class \_\_cxxabiv1::\_\_pointer\_to\_member\_type\_info** 565](#_Toc494808417)

[**16.1.15 Class \_\_gnu\_cxx::stdio\_filebuf<char, char\_traits<char> >** 567](#_Toc494808418)

[**16.1.16 Class \_\_gnu\_cxx::stdio\_filebuf<wchar\_t, char\_traits<wchar\_t> >** 567](#_Toc494808419)

[**16.1.17 Class \_\_gnu\_cxx::\_\_pool\_alloc\_base** 567](#_Toc494808420)

[**16.1.18 Class \_\_gnu\_cxx::stdio\_sync\_filebuf<char, char\_traits<char> >** 568](#_Toc494808421)

[**16.1.19 Class \_\_gnu\_cxx::stdio\_sync\_filebuf<wchar\_t, char\_traits<wchar\_t> >** 569](#_Toc494808422)

[**16.1.20 Class exception** 571](#_Toc494808423)

[**16.1.21 Class bad\_typeid** 571](#_Toc494808424)

[**16.1.22 Class logic\_error** 572](#_Toc494808425)

[**16.1.23 Class range\_error** 573](#_Toc494808426)

[**16.1.24 Class domain\_error** 574](#_Toc494808427)

[**16.1.25 Class length\_error** 575](#_Toc494808428)

[**16.1.26 Class out\_of\_range** 576](#_Toc494808429)

[**16.1.27 Class bad\_exception** 577](#_Toc494808430)

[**16.1.28 Class runtime\_error** 578](#_Toc494808431)

[**16.1.29 Class overflow\_error** 579](#_Toc494808432)

[**16.1.30 Class underflow\_error** 580](#_Toc494808433)

[**16.1.31 Class invalid\_argument** 581](#_Toc494808434)

[**16.1.32 Class bad\_cast** 582](#_Toc494808435)

[**16.1.33 Class bad\_alloc** 582](#_Toc494808436)

[**16.1.34 struct \_\_numeric\_limits\_base** 583](#_Toc494808437)

[**16.1.35 struct numeric\_limits<long double>** 584](#_Toc494808438)

[**16.1.36 struct numeric\_limits<long long>** 585](#_Toc494808439)

[**16.1.37 struct numeric\_limits<unsigned long long>** 586](#_Toc494808440)

[**16.1.38 struct numeric\_limits<float>** 587](#_Toc494808441)

[**16.1.39 struct numeric\_limits<double>** 588](#_Toc494808442)

[**16.1.40 struct numeric\_limits<short>** 588](#_Toc494808443)

[**16.1.41 struct numeric\_limits<unsigned short>** 589](#_Toc494808444)

[**16.1.42 struct numeric\_limits<int>** 590](#_Toc494808445)

[**16.1.43 struct numeric\_limits<unsigned int>** 591](#_Toc494808446)

[**16.1.44 struct numeric\_limits<long>** 592](#_Toc494808447)

[**16.1.45 struct numeric\_limits<unsigned long>** 593](#_Toc494808448)

[**16.1.46 struct numeric\_limits<wchar\_t>** 593](#_Toc494808449)

[**16.1.47 struct numeric\_limits<unsigned char>** 594](#_Toc494808450)

[**16.1.48 struct numeric\_limits<signed char>** 595](#_Toc494808451)

[**16.1.49 struct numeric\_limits<char>** 596](#_Toc494808452)

[**16.1.50 struct numeric\_limits<bool>** 597](#_Toc494808453)

[**16.1.51 Class ctype\_base** 597](#_Toc494808454)

[**16.1.52 Class \_\_ctype\_abstract\_base<char>** 598](#_Toc494808455)

[**16.1.53 Class \_\_ctype\_abstract\_base<wchar\_t>** 599](#_Toc494808456)

[**16.1.54 Class ctype<char>** 600](#_Toc494808457)

[**16.1.55 Class ctype<wchar\_t>** 601](#_Toc494808458)

[**16.1.56 Class ctype\_byname<char>** 603](#_Toc494808459)

[**16.1.57 Class ctype\_byname<wchar\_t>** 605](#_Toc494808460)

[**16.1.58 Class basic\_string<char, char\_traits<char>, allocator<char> >** 606](#_Toc494808461)

[**16.1.59 Class basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >** 611](#_Toc494808462)

[**16.1.60 Class basic\_stringstream<char, char\_traits<char>, allocator<char> >** 617](#_Toc494808463)

[**16.1.61 Class basic\_stringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >** 618](#_Toc494808464)

[**16.1.62 Class basic\_istringstream<char, char\_traits<char>, allocator<char> >** 620](#_Toc494808465)

[**16.1.63 Class basic\_istringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >** 621](#_Toc494808466)

[**16.1.64 Class basic\_ostringstream<char, char\_traits<char>, allocator<char> >** 622](#_Toc494808467)

[**16.1.65 Class basic\_ostringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >** 624](#_Toc494808468)

[**16.1.66 Class basic\_stringbuf<char, char\_traits<char>, allocator<char> >** 625](#_Toc494808469)

[**16.1.67 Class basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >** 627](#_Toc494808470)

[**16.1.68 Class basic\_iostream<char, char\_traits<char> >** 630](#_Toc494808471)

[**16.1.69 Class basic\_iostream<wchar\_t, char\_traits<wchar\_t> >** 631](#_Toc494808472)

[**16.1.70 Class basic\_istream<char, char\_traits<char> >** 632](#_Toc494808473)

[**16.1.71 Class basic\_istream<wchar\_t, char\_traits<wchar\_t> >** 636](#_Toc494808474)

[**16.1.72 Class istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >** 639](#_Toc494808475)

[**16.1.73 Class istreambuf\_iterator<char, char\_traits<char> >** 639](#_Toc494808476)

[**16.1.74 Class basic\_ostream<char, char\_traits<char> >** 640](#_Toc494808477)

[**16.1.75 Class basic\_ostream<wchar\_t, char\_traits<wchar\_t> >** 643](#_Toc494808478)

[**16.1.76 Class basic\_fstream<char, char\_traits<char> >** 646](#_Toc494808479)

[**16.1.77 Class basic\_fstream<wchar\_t, char\_traits<wchar\_t> >** 648](#_Toc494808480)

[**16.1.78 Class basic\_ifstream<char, char\_traits<char> >** 649](#_Toc494808481)

[**16.1.79 Class basic\_ifstream<wchar\_t, char\_traits<wchar\_t> >** 650](#_Toc494808482)

[**16.1.80 Class basic\_ofstream<char, char\_traits<char> >** 652](#_Toc494808483)

[**16.1.81 Class basic\_ofstream<wchar\_t, char\_traits<wchar\_t> >** 653](#_Toc494808484)

[**16.1.82 Class basic\_streambuf<char, char\_traits<char> >** 654](#_Toc494808485)

[**16.1.83 Class basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >** 657](#_Toc494808486)

[**16.1.84 Class basic\_filebuf<char, char\_traits<char> >** 660](#_Toc494808487)

[**16.1.85 Class basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >** 663](#_Toc494808488)

[**16.1.86 Class ios\_base** 665](#_Toc494808489)

[**16.1.87 Class basic\_ios<char, char\_traits<char> >** 667](#_Toc494808490)

[**16.1.88 Class basic\_ios<wchar\_t, char\_traits<wchar\_t> >** 669](#_Toc494808491)

[**16.1.89 Class ios\_base::failure** 672](#_Toc494808492)

[**16.1.90 Class \_\_timepunct<char>** 673](#_Toc494808493)

[**16.1.91 Class \_\_timepunct<wchar\_t>** 674](#_Toc494808494)

[**16.1.92 Class messages\_base** 675](#_Toc494808495)

[**16.1.93 Class messages<char>** 676](#_Toc494808496)

[**16.1.94 Class messages<wchar\_t>** 677](#_Toc494808497)

[**16.1.95 Class messages\_byname<char>** 678](#_Toc494808498)

[**16.1.96 Class messages\_byname<wchar\_t>** 680](#_Toc494808499)

[**16.1.97 Class numpunct<char>** 681](#_Toc494808500)

[**16.1.98 Class numpunct<wchar\_t>** 682](#_Toc494808501)

[**16.1.99 Class numpunct\_byname<char>** 684](#_Toc494808502)

[**16.1.100 Class numpunct\_byname<wchar\_t>** 685](#_Toc494808503)

[**16.1.101 Class \_\_codecvt\_abstract\_base<char, char, \_\_mbstate\_t>** 686](#_Toc494808504)

[**16.1.102 Class \_\_codecvt\_abstract\_base<wchar\_t, char, \_\_mbstate\_t>** 687](#_Toc494808505)

[**16.1.103 Class codecvt\_base** 688](#_Toc494808506)

[**16.1.104 Class codecvt<char, char, \_\_mbstate\_t>** 689](#_Toc494808507)

[**16.1.105 Class codecvt<wchar\_t, char, \_\_mbstate\_t>** 690](#_Toc494808508)

[**16.1.106 Class codecvt\_byname<char, char, \_\_mbstate\_t>** 692](#_Toc494808509)

[**16.1.107 Class codecvt\_byname<wchar\_t, char, \_\_mbstate\_t>** 694](#_Toc494808510)

[**16.1.108 Class collate<char>** 695](#_Toc494808511)

[**16.1.109 Class collate<wchar\_t>** 696](#_Toc494808512)

[**16.1.110 Class collate\_byname<char>** 698](#_Toc494808513)

[**16.1.111 Class collate\_byname<wchar\_t>** 699](#_Toc494808514)

[**16.1.112 Class time\_base** 700](#_Toc494808515)

[**16.1.113 Class time\_get\_byname<char, istreambuf\_iterator<char, char\_traits<char> > >** 701](#_Toc494808516)

[**16.1.114 Class time\_get\_byname<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >** 703](#_Toc494808517)

[**16.1.115 Class time\_put\_byname<char, ostreambuf\_iterator<char, char\_traits<char> > >** 705](#_Toc494808518)

[**16.1.116 Class time\_put\_byname<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >** 706](#_Toc494808519)

[**16.1.117 Class time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >** 708](#_Toc494808520)

[**16.1.118 Class time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >** 711](#_Toc494808521)

[**16.1.119 Class time\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >** 714](#_Toc494808522)

[**16.1.120 Class time\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >** 715](#_Toc494808523)

[**16.1.121 Class moneypunct<char, false>** 717](#_Toc494808524)

[**16.1.122 Class moneypunct<char, true>** 719](#_Toc494808525)

[**16.1.123 Class moneypunct<wchar\_t, false>** 721](#_Toc494808526)

[**16.1.124 Class moneypunct<wchar\_t, true>** 723](#_Toc494808527)

[**16.1.125 Class moneypunct\_byname<char, false>** 725](#_Toc494808528)

[**16.1.126 Class moneypunct\_byname<char, true>** 726](#_Toc494808529)

[**16.1.127 Class moneypunct\_byname<wchar\_t, false>** 727](#_Toc494808530)

[**16.1.128 Class moneypunct\_byname<wchar\_t, true>** 729](#_Toc494808531)

[**16.1.129 Class money\_base** 730](#_Toc494808532)

[**16.1.130 Class money\_get<char, istreambuf\_iterator<char, char\_traits<char> > >** 731](#_Toc494808533)

[**16.1.131 Class money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >** 733](#_Toc494808534)

[**16.1.132 Class money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >** 736](#_Toc494808535)

[**16.1.133 Class money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >** 738](#_Toc494808536)

[**16.1.134 Class locale** 740](#_Toc494808537)

[**16.1.135 Class locale::facet** 741](#_Toc494808538)

[**16.1.136 facet functions** 742](#_Toc494808539)

[**16.1.137 Class \_\_num\_base** 745](#_Toc494808540)

[**16.1.138 Class num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >** 745](#_Toc494808541)

[**16.1.139 Class num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >** 751](#_Toc494808542)

[**16.1.140 Class num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >** 756](#_Toc494808543)

[**16.1.141 Class num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >** 760](#_Toc494808544)

[**16.1.142 Class \_\_basic\_file<char>** 764](#_Toc494808545)

[**16.1.143 Class \_List\_node\_base** 764](#_Toc494808546)

[**16.1.144 Class allocator<char>** 765](#_Toc494808547)

[**16.1.145 Class allocator<wchar\_t>** 765](#_Toc494808548)

[**16.1.146 Class \_\_gnu\_cxx::\_\_pool<true>** 765](#_Toc494808549)

[**16.1.147 Class \_\_gnu\_cxx::\_\_pool<false>** 766](#_Toc494808550)

[**16.1.148 Class \_\_gnu\_cxx::free\_list** 766](#_Toc494808551)

[**16.1.149 Class char\_traits<char>** 766](#_Toc494808552)

[**16.1.150 Class char\_traits<wchar\_t>** 766](#_Toc494808553)

[**16.2 Interface Definitions for libstdcxx** 767](#_Toc494808554)

[**VI Commands and Utilities** 768](#_Toc494808555)

[**17 Commands and Utilities** 769](#_Toc494808556)

[**17.1 Commands and Utilities** 769](#_Toc494808557)

[**17.2 Command Behavior** 770](#_Toc494808558)

[**VII Execution Environment** 829](#_Toc494808559)

[**18 File System Hierarchy** 830](#_Toc494808560)

[**18.1 /dev: Device Files** 830](#_Toc494808561)

[**18.2 /etc: Host-specific system configuration** 830](#_Toc494808562)

[**18.2.1 File Naming Conventions** 831](#_Toc494808563)

[**18.3 User Accounting Databases** 832](#_Toc494808564)

[**18.4 Path For System Administration Utilities** 832](#_Toc494808565)

[**19 Additional Recommendations** 833](#_Toc494808566)

[**19.1 Recommendations for applications on ownership and permissions** 833](#_Toc494808567)

[**19.1.1 Directory Write Permissions** 833](#_Toc494808568)

[**19.1.2 File Write Permissions** 833](#_Toc494808569)

[**19.1.3 File Read and execute Permissions** 833](#_Toc494808570)

[**19.1.4 SUID and SGID Permissions** 833](#_Toc494808571)

[**19.1.5 Privileged users** 833](#_Toc494808572)

[**19.1.6 Changing permissions** 833](#_Toc494808573)

[**19.1.7 Removable Media (Cdrom, Floppy, etc.)** 834](#_Toc494808574)

[**19.1.8 Installable applications** 834](#_Toc494808575)

[**20 Additional Behaviors** 835](#_Toc494808576)

[**20.1 Mandatory Optional Behaviors** 835](#_Toc494808577)

[**20.1.1 Special Requirements** 835](#_Toc494808578)

[**20.2 Optional Mandatory Behaviors** 836](#_Toc494808579)

[**20.3 Executable Scripts** 836](#_Toc494808580)

[**21 Localization** 838](#_Toc494808581)

[**21.1 Introduction** 838](#_Toc494808582)

[**21.2 Regular Expressions** 838](#_Toc494808583)

[**21.3 Pattern Matching Notation** 838](#_Toc494808584)

[**VIII System Initialization** 840](#_Toc494808585)

[**22 System Initialization** 841](#_Toc494808586)

[**22.1 Cron Jobs** 841](#_Toc494808587)

[**22.2 Init Script Actions** 842](#_Toc494808588)

[**22.3 Comment Conventions for Init Scripts** 843](#_Toc494808589)

[**22.4 Installation and Removal of Init Scripts** 845](#_Toc494808590)

[**22.5 Run Levels** 846](#_Toc494808591)

[**22.6 Facility Names** 846](#_Toc494808592)

[**22.7 Script Names** 847](#_Toc494808593)

[**22.8 Init Script Functions** 847](#_Toc494808594)

[**IX Users & Groups** 850](#_Toc494808595)

[**23 Users & Groups** 851](#_Toc494808596)

[**23.1 User and Group Database** 851](#_Toc494808597)

[**23.2 User & Group Names** 851](#_Toc494808598)

[**23.3 User ID Ranges** 852](#_Toc494808599)

[**23.4 Rationale** 852](#_Toc494808600)

[**X Network Security Services** 853](#_Toc494808601)

[**24 Libraries** 854](#_Toc494808602)

[**24.1 Interfaces for libnspr4** 854](#_Toc494808603)

[**24.1.1 Netscape Portable Runtime** 854](#_Toc494808604)

[**24.2 Data Definitions for libnspr4** 855](#_Toc494808605)

[**24.2.1 nspr4/nspr.h** 856](#_Toc494808606)

[**24.2.2 nspr4/plarena.h** 856](#_Toc494808607)

[**24.2.3 nspr4/plhash.h** 856](#_Toc494808608)

[**24.2.4 nspr4/pratom.h** 856](#_Toc494808609)

[**24.2.5 nspr4/prclist.h** 857](#_Toc494808610)

[**24.2.6 nspr4/prcvar.h** 857](#_Toc494808611)

[**24.2.7 nspr4/prerror.h** 857](#_Toc494808612)

[**24.2.8 nspr4/prinit.h** 857](#_Toc494808613)

[**24.2.9 nspr4/prinrval.h** 857](#_Toc494808614)

[**24.2.10 nspr4/prio.h** 858](#_Toc494808615)

[**24.2.11 nspr4/private/pprio.h** 863](#_Toc494808616)

[**24.2.12 nspr4/prlock.h** 863](#_Toc494808617)

[**24.2.13 nspr4/prmem.h** 863](#_Toc494808618)

[**24.2.14 nspr4/prmon.h** 863](#_Toc494808619)

[**24.2.15 nspr4/prnetdb.h** 863](#_Toc494808620)

[**24.2.16 nspr4/prthread.h** 864](#_Toc494808621)

[**24.2.17 nspr4/prtime.h** 865](#_Toc494808622)

[**24.2.18 nspr4/prtypes.h** 865](#_Toc494808623)

[**24.3 Interfaces for libnss3** 866](#_Toc494808624)

[**24.3.1 NSS Utility** 866](#_Toc494808625)

[**24.4 Data Definitions for libnss3** 866](#_Toc494808626)

[**24.4.1 nss3/blapit.h** 867](#_Toc494808627)

[**24.4.2 nss3/cert.h** 867](#_Toc494808628)

[**24.4.3 nss3/certt.h** 868](#_Toc494808629)

[**24.4.4 nss3/cmsreclist.h** 875](#_Toc494808630)

[**24.4.5 nss3/cryptoht.h** 875](#_Toc494808631)

[**24.4.6 nss3/hasht.h** 875](#_Toc494808632)

[**24.4.7 nss3/key.h** 876](#_Toc494808633)

[**24.4.8 nss3/keyhi.h** 876](#_Toc494808634)

[**24.4.9 nss3/keyt.h** 876](#_Toc494808635)

[**24.4.10 nss3/keythi.h** 876](#_Toc494808636)

[**24.4.11 nss3/nss.h** 877](#_Toc494808637)

[**24.4.12 nss3/nssb64.h** 878](#_Toc494808638)

[**24.4.13 nss3/nssb64t.h** 878](#_Toc494808639)

[**24.4.14 nss3/nssilckt.h** 878](#_Toc494808640)

[**24.4.15 nss3/nssrwlkt.h** 878](#_Toc494808641)

[**24.4.16 nss3/ocspt.h** 879](#_Toc494808642)

[**24.4.17 nss3/pk11pub.h** 879](#_Toc494808643)

[**24.4.18 nss3/pkcs11t.h** 879](#_Toc494808644)

[**24.4.19 nss3/pkcs7t.h** 880](#_Toc494808645)

[**24.4.20 nss3/secasn1t.h** 880](#_Toc494808646)

[**24.4.21 nss3/seccomon.h** 881](#_Toc494808647)

[**24.4.22 nss3/secdert.h** 881](#_Toc494808648)

[**24.4.23 nss3/secdigt.h** 881](#_Toc494808649)

[**24.4.24 nss3/secmodt.h** 882](#_Toc494808650)

[**24.4.25 nss3/secoidt.h** 884](#_Toc494808651)

[**24.4.26 nss3/secpkcs5.h** 889](#_Toc494808652)

[**24.4.27 nss3/secport.h** 889](#_Toc494808653)

[**24.5 Interfaces for libssl3** 889](#_Toc494808654)

[**24.5.1 NSS SSL** 890](#_Toc494808655)

[**24.6 Data Definitions for libssl3** 891](#_Toc494808656)

[**24.6.1 nss3/ecl-exp.h** 891](#_Toc494808657)

[**24.6.2 nss3/ssl.h** 892](#_Toc494808658)

[**24.6.3 nss3/sslerr.h** 894](#_Toc494808659)

[**24.6.4 nss3/sslproto.h** 896](#_Toc494808660)

[**24.6.5 nss3/sslt.h** 898](#_Toc494808661)

[**XI Package Format and Installation** 900](#_Toc494808662)

[**25 Software Installation** 901](#_Toc494808663)

[**25.1 Introduction** 901](#_Toc494808664)

[**25.2 Package File Format** 901](#_Toc494808665)

[**25.2.1 Lead Section** 901](#_Toc494808666)

[**25.2.2 Header Structure** 902](#_Toc494808667)

[**25.2.3 Signature Section** 905](#_Toc494808668)

[**25.2.4 Header Section** 906](#_Toc494808669)

[**25.2.5 Payload Section** 917](#_Toc494808670)

[**25.3 Package Script Restrictions** 919](#_Toc494808671)

[**25.4 Package Tools** 919](#_Toc494808672)

[**25.5 Package Naming Conventions** 920](#_Toc494808673)

[**25.6 Package Dependencies** 920](#_Toc494808674)

[**25.7 Package Architecture Considerations** 921](#_Toc494808675)

[**Annex A Alphabetical Listing of Interfaces by Library** 922](#_Toc494808676)

[**A.1 libc** 922](#_Toc494808677)

[**A.2 libcrypt** 935](#_Toc494808678)

[**A.3 libdl** 935](#_Toc494808679)

[**A.4 libgcc\_s** 935](#_Toc494808680)

[**A.5 libm** 935](#_Toc494808681)

[**A.6 libncurses** 938](#_Toc494808682)

[**A.7 libncursesw** 942](#_Toc494808683)

[**A.8 libpam** 947](#_Toc494808684)

[**A.9 libpthread** 948](#_Toc494808685)

[**A.10 librt** 950](#_Toc494808686)

[**A.11 libutil** 951](#_Toc494808687)

[**A.12 libz** 951](#_Toc494808688)

[**A.13 libnspr4** 952](#_Toc494808689)

[**A.14 libnss3** 953](#_Toc494808690)

[**A.15 libssl3** 954](#_Toc494808691)

[**Annex B GNU Free Documentation License (Informative)** 956](#_Toc494808692)

[**B.1 PREAMBLE** 956](#_Toc494808693)

[**B.2 APPLICABILITY AND DEFINITIONS** 956](#_Toc494808694)

[**B.3 VERBATIM COPYING** 957](#_Toc494808695)

[**B.4 COPYING IN QUANTITY** 957](#_Toc494808696)

[**B.5 MODIFICATIONS** 958](#_Toc494808697)

[**B.6 COMBINING DOCUMENTS** 959](#_Toc494808698)

[**B.7 COLLECTIONS OF DOCUMENTS** 959](#_Toc494808699)

[**B.8 AGGREGATION WITH INDEPENDENT WORKS** 959](#_Toc494808700)

[**B.9 TRANSLATION** 960](#_Toc494808701)

[**B.10 TERMINATION** 960](#_Toc494808702)

[**B.11 FUTURE REVISIONS OF THIS LICENSE** 960](#_Toc494808703)

[**B.12 How to use this License for your documents** 960](#_Toc494808704)

**No table of figures entries found.**

# **List of Figures**

[10-1 Version Definition Entries](#ID_VERDEFENTRIES)

[10-2 Version Definition Auxiliary Entries](#ID_VERDEFEXTS)

[10-3 Version Needed Entries](#ID_VERNEEDFIG)

[10-4 Version Needed Auxiliary Entries](#ID_VERNEEDEXTFIG)

[11-1 Dynamic Structure](#_1788)

[12-1 Category 1 Virtual Table](#ID_CXX_45_VIRTUALTABLE_45_CAT1)

[12-2 Category 2 Virtual Table](#ID_CXX_45_VIRTUALTABLE_45_CAT2)

[12-3 Run-Time Type Information Prefix](#ID_CXX_45_RTTI)

[12-4 Run-Time Type Information For Classes with no base class](#ID_CXX_45_RTTI_45_CLASS)

[12-5 Run-Time Type Information for Classes with a single base class](#ID_CXX_45_RTTI_45_SICLASS)

[12-6 Run-Time Type Information for classes with multiple inheritance](#ID_CXX_45_RTTI_45_VMICLASS)

[12-7 Run-Time Type Information for pointer types](#ID_CXX_45_RTTI_45_POINTERCLASS)

[12-8 Run-Time Type Information for pointer to member types](#ID_CXX_45_RTTI_45_PTOMEMBERCLASS)

# **Foreword**

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

The committee responsible for this document is Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages, their environments and system software interfaces*.

This document is a direct adoption of the Linux Standards Base (LSB) 5.0 Core Generic specification, issued by the Linux Foundation, which was the second part of ISO/IEC 23360-1. The previous release of this standard, ISO/IEC 23360-1 through -8:2005 was published under the ISO/IEC/JTC 1 Publicly Available Specification process. This document is published under the GNU Free Documentation License (See Annex B).

This is version 1.0 of the Linux Standard Base Core Generic Standard. This standard replaces the Runtime Languages portion of ISO/IEC 23360-1 Linux Standard Base, which is cancelled and replaced by ISO/IEC 23360-1-1 through -1-5. This document, ISO/IEC 23360-1-2 LSB Core Generic is augmented by processor-specific material for

* the Intel 32 bit architecture in ISO/IEC 23360-2-2
* the Intel 64 bit architecture in ISO/IEC 23360-3-2,
* the PowerPC 32 bit architecture in ISO/IEC 23360-4-2,
* the PowerPC 64 bit architecture in ISO/IEC 23360-5-2,
* the IBM S390 architecture in ISO/IEC 23360-6-2,
* the IBM S390X architecture in ISO/IEC 23360-7-2, and
* the AMD 64 bit architecture in ISO/IEC 23360-8-2.

# **Status of this Document**

A list of current released Linux Standard Base (LSB) specifications is available at http://refspecs.linuxbase.org (http://refspecs.linuxbase.org/).

If you wish to make comments regarding this document in a manner that is tracked by the LSB project, please submit them using the Linux Foundation public bug database at http://bugs.linuxbase.org. Please enter your feedback, carefully indicating the title of the section for which you are submitting feedback, and the volume and version of the specification where you found the problem, quoting the incorrect text if appropriate. If you are suggesting a new feature, please indicate what the problem you are trying to solve is. That is more important than the solution, in fact.

If you do not have or wish to create a bug database account then you can also e-mail feedback to <lsb-discuss@lists.linuxfoundation.org> (subscribe (http://lists.linuxfoundation.org/mailman/listinfo/lsb-discuss), archives (http://lists.linuxfoundation.org/pipermail/lsb-discuss/)), and arrangements will be made to transpose the comments to our public bug database.

# **Introduction**

The LSB defines a binary interface for application programs that are compiled and packaged for LSB-conforming implementations on many different hardware architectures. A binary specification must include information specific to the computer processor architecture for which it is intended. To avoid the complexity of conditional descriptions, the specification has instead been divided into generic parts which are augmented by one of several architecture-specific parts, depending on the target processor architecture; the generic part will indicate when reference must be made to the architecture part, and vice versa.

This document should be used in conjunction with the documents it references. This document enumerates the system components it includes, but descriptions of those components may be included entirely or partly in this document, partly in other documents, or entirely in other reference documents. For example, the section that describes system service routines includes a list of the system routines supported in this interface, formal declarations of the data structures they use that are visible to applications, and a pointer to the underlying referenced specification for information about the syntax and semantics of each call. Only those routines not described in standards referenced by this document, or extensions to those standards, are described in the detail. Information referenced in this way is as much a part of this document as is the information explicitly included here.

The specification carries a version number of either the form *x.y* or *x.y.z*. This version number carries the following meaning:

1. The first number (*x*) is the major version number. Versions sharing the same major version number shall be compatible in a backwards direction; that is, a newer version shall be compatible with an older version. Any deletion of a library results in a new major version number. Interfaces marked as deprecated may be removed from the specification at a major version change.

2. The second number (*y*) is the minor version number. Libraries and individual interfaces may be added, but not removed. Interfaces may be marked as deprecated at a minor version change. Other minor changes may be permitted at the discretion of the LSB workgroup.

3. The third number (*z*), if present, is the editorial level. Only editorial changes should be included in such versions.

Since this specification is a descriptive Application Binary Interface, and not a source level API specification, it is not possible to make a guarantee of 100% backward compatibility between major releases. However, it is the intent that those parts of the binary interface that are visible in the source level API will remain backward compatible from version to version, except where a feature marked as "Deprecated" in one release may be removed from a future release. Implementors are strongly encouraged to make use of symbol versioning to permit simultaneous support of applications conforming to different releases of this specification.

LSB is a trademark of the Linux Foundation. Developers of applications or implementations interested in using the trademark should see the Linux Foundation Certification Policy for details.

# **I Introductory Elements**

# **1 Scope**

## **1.1 General**

*Note: Very minor modifications are made to this subclause to reflect the numbering employed by ISO/IEC. The changes made are in italics.*

The Linux Standard Base (LSB) defines a system interface for compiled applications and a minimal environment for support of installation scripts. Its purpose is to enable a uniform industry standard environment for high-volume applications conforming to the LSB.

*The LSB specification set is divided into modules, each of which provides fundamental system interfaces, libraries, and runtime environment upon which all conforming applications and libraries using that module depend.*

*The Parts of the Linux Standard Base are:*

*• ISO/IEC 23360-1-1 Information technology—Operating Systems – Linux Standards Base common definitions,*

*• ISO/IEC 23360-1-2 Information technology—Operating Systems – Linux Standards Base core specification, generic part*

*• ISO/IEC 23360-1-3 Information technology—Operating Systems – Linux Standards Base desktop specification, generic part*

*• ISO/IEC 23360-1-4 Information technology—Operating Systems – Linux Standards Base languages specification, generic part*

*• ISO/IEC 23360-1-5 Information technology—Operating Systems – Linux Standards Base imaging specification, generic part*

*• ISO/IEC 23360-1-6 Information technology—Operating Systems – Linux Standards Base graphic specification, generic part*

These specifications are composed of two basic parts: a common part describing those parts of the interface that remain constant across all implementations of the LSB, and an architecture-specific part describing the parts of the interface that vary by processor architecture. Together, the common part and the relevant architecture-specific part for a single hardware architecture provide a complete interface specification for compiled application programs on systems that share a common hardware architecture.

*Processor-specific versions of ISO/IEC 23360-1-2 and 23360-1-3 are published as follows:*

• *ISO/IEC 23360-2-2 Information technology—Operating Systems – Linux Standards Base core specification for the X86*

*• ISO/IEC 23360-2-3 Information technology—Operating Systems – Linux Standards Base desktop specification for the X86*

*• ISO/IEC 23360-3-2 Information technology—Operating Systems – Linux Standards Base core specification for the X86*

*• ISO/IEC 23360-3-3 Information technology—Operating Systems – Linux Standards Base desktop specification for Itanium*

*• ISO/IEC 23360-4-2 Information technology—Operating Systems – Linux Standards Base core specification for PPC32*

*• ISO/IEC 23360-4-3 Information technology—Operating Systems – Linux Standards Base desktop specification for PPC32*

*• ISO/IEC 23360-5-2 Information technology—Operating Systems – Linux Standards Base core specification for PPC64*

*• ISO/IEC 23360-5-3 Information technology—Operating Systems – Linux Standards Base desktop specification for PPC64*

*• ISO/IEC 23360-6-2 Information technology—Operating Systems – Linux Standards Base core specification for S390*

*• ISO/IEC 23360-6-3 Information technology—Operating Systems – Linux Standards Base desktop specification for S390*

*• ISO/IEC 23360-7-2 Information technology—Operating Systems – Linux Standards Base core specification for S390X*

*• ISO/IEC 23360-7-3 Information technology—Operating Systems – Linux Standards Base desktop specification for S390X*

*• ISO/IEC 23360-8-2 Information technology—Operating Systems – Linux Standards Base core specification for AMD64*

*• ISO/IEC 23360-8-3 Information technology—Operating Systems – Linux Standards Base desktop specification for AMD64*

The LSB contains both a set of Application Program Interfaces (APIs) and Application Binary Interfaces (ABIs). APIs may appear in the source code of portable applications, while the compiled binary of that application may use the larger set of ABIs. A conforming implementation provides all of the ABIs listed here. The compilation system may replace (e.g. by macro definition) certain APIs with calls to one or more of the underlying binary interfaces, and may insert calls to binary interfaces as needed.

The LSB is primarily a binary interface definition. Not all of the source level APIs available to applications may be contained in this specification.

## **1.2 Module Specific Scope**

This is the common part of the Core module of the Linux Standard Base (LSB), LSB Core - Generic. This module provides the fundamental system interfaces, libraries, and runtime environment upon which all conforming applications and libraries depend.

LSB Core - Generic, the common part, should be used in conjunction with an architecture-specific part. Whenever a section of the common part is supplemented by architecture-specific information, the common part includes a reference to the architecture-specific part. Architecture-specific parts of the LSB Core Specification may also contain additional information that is not referenced in the common part.

Interfaces described in this part of the LSB Core Specification are mandatory except where explicitly listed otherwise. Interfaces described in the LSB Core module are supplemented by other LSB modules. All other modules depend on the presence of LSB Core.

# **2 References**

## **2.1 Normative References**

The following specifications are incorporated by reference into this specification. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced specification (including any amendments) applies.

**Note:** Where copies of a referenced specification are available on the World Wide Web, a Uniform Resource Locator (URL) is given, for informative purposes only. Such URL might at any given time resolve to a more recent copy of the specification, or be out of date (not resolve). Reference copies of specifications at the revision level indicated may be found at the Linux Foundation's Reference Specifications (http://refspecs.linuxbase.org) site.

**Table 2-1 Normative References**

| **Name** | **Title** | **URL** |
| --- | --- | --- |
| Filesystem Hierarchy Standard | Filesystem Hierarchy Standard (FHS) 3.0 | http://refspecs.linuxbase.org/fhs |
| ISO C (1999) | ISO/IEC 9899:1999 - Programming Languages -- C |  |
| ISO/IEC 14882: 2003 C++ Language | ISO/IEC 14882: 2003 Programming languages --C++ |  |
| Itanium™ C++ ABI | Itanium™ C++ ABI (Revision 1.86) | http://refspecs.linuxfoundation.org/cxxabi-1.86.html |
| Large File Support | Large File Support | http://www.UNIX-systems.org/version2/whatsnew/lfs20mar.html |
| Libncursesw API | Libncursesw API | http://invisible-island.net/ncurses/man/ncurses.3x.html |
| Libncursesw Placeholder | Libncursesw Specification Placeholder | http://refspecs.linux-foundation.org/libncursesw/libncurses.html |
| POSIX 1003.1-2001 (ISO/IEC 9945-2003) | ISO/IEC 9945-1:2003 Information technology -- Portable Operating System Interface (POSIX) -- Part 1: Base Definitions  ISO/IEC 9945-2:2003 Information technology -- Portable Operating System Interface (POSIX) -- Part 2: System Interfaces  ISO/IEC 9945-3:2003 Information technology -- Portable Operating System Interface (POSIX) -- Part 3: Shell and Utilities  ISO/IEC 9945-4:2003 Information technology -- Portable Operating System Interface (POSIX) -- Part 4: Rationale  Including Technical Cor. 1: 2004 | http://www.unix.org/version3/ |
| POSIX 1003.1-2008 (ISO/IEC 9945-2009) | Portable Operating System Interface (POSIX®) 2008 Edition / The Open Group Technical Standard Base Specifications, Issue 7 | http://www.unix.org/version4/ |
| SUSv2 | CAE Specification, January 1997, System Interfaces and Headers (XSH),Issue 5 (ISBN: 1-85912-181-0, C606) | http://www.opengroup.org/publications/catalog/un.htm |
| SVID Issue 3 | American Telephone and Telegraph Company, System V Interface Definition, Issue 3; Morristown, NJ, UNIX Press, 1989. (ISBN 0201566524) |  |
| SVID Issue 4 | System V Interface Definition, Fourth Edition | http://refspecs.linuxfoundation.org/svid4/ |
| System V ABI | System V Application Binary Interface, Edition 4.1 | http://www.sco.com/developers/devspecs/gabi41.pdf |
| System V ABI Update | System V Application Binary Interface - DRAFT - 17 December 2003 | http://www.sco.com/developers/gabi/2003-12-17/contents.html |
| X/Open Curses, Issue 7 | X/Open Curses, Issue 7 (ISBN: 1-931624-83-6, The Open Group, November 2009) | https://www2.opengroup.org/ogsys/catalog/C094 |

## **2.2 Informative References/Bibliography**

The documents listed below provide essential background information to implementors of this specification. These references are included for information only, and do not represent normative parts of this specification.

**Table 2-2 Other References**

| **Name** | **Title** | **URL** |
| --- | --- | --- |
| DWARF Debugging Information Format, Version 4 | DWARF Debugging Information Format, Version 4 (June 10, 2010) | http://www.dwarfstd.org/doc/DWARF4.pdf |
| IEC 60559/IEEE 754 Floating Point | IEC 60559:1989 Binary floating-point arithmetic for microprocessor systems | http://www.ieee.org/ |
| ISO/IEC TR14652 | ISO/IEC Technical Report 14652:2002 Specification method for cultural conventions |  |
| ITU-T V.42 | International Telecommunication Union Recommendation V.42 (2002): Error-correcting procedures for DCEs using asynchronous-to-synchronous conversionITUV | http://www.itu.int/rec/recommendation.asp?type=folders&lang=e&parent=T-REC-V.42 |
| Li18nux Globalization Specification | LI18NUX 2000 Globalization Specification, Version 1.0 with Amendment 4 | http://www.openi18n.org/docs/html/LI18NUX-2000-amd4.htm |
| Linux Allocated Device Registry | LINUX ALLOCATED DEVICES | http://www.lanana.org/docs/device-list/devices-2.6+.txt |
| Linux Assigned Names And Numbers Authority | Linux Assigned Names And Numbers Authority | http://www.lanana.org/ |
| Mozilla's NSS SSL Reference | Mozilla's NSS SSL Reference | http://www.mozilla.org/projects/security/pki/nss/ref/ssl/ |
| NSPR Reference | Mozilla's NSPR Reference | http://refspecs.linuxfoundation.org/NSPR\_API\_Reference/NSPR\_API.html |
| PAM | Open Software Foundation, Request For Comments: 86.0 , October 1995, V. Samar & R.Schemers (SunSoft) | http://www.opengroup.org/tech/rfc/mirror-rfc/rfc86.0.txt |
| RFC 1321: The MD5 Message-Digest Algorithm | IETF RFC 1321: The MD5 Message-Digest Algorithm | http://www.ietf.org/rfc/rfc1321.txt |
| RFC 1833: Binding Protocols for ONC RPC Version 2 | IETF RFC 1833: Binding Protocols for ONC RPC Version 2 | http://www.ietf.org/rfc/rfc1833.txt |
| RFC 1950: ZLIB Compressed Data Format Specication | IETF RFC 1950: ZLIB Compressed Data Format Specification | http://www.ietf.org/rfc/rfc1950.txt |
| RFC 1951: DEFLATE Compressed Data Format Specification | IETF RFC 1951: DEFLATE Compressed Data Format Specification version 1.3 | http://www.ietf.org/rfc/rfc1951.txt |
| RFC 1952: GZIP File Format Specification | IETF RFC 1952: GZIP file format specification version 4.3 | http://www.ietf.org/rfc/rfc1952.txt |
| RFC 2440: OpenPGP Message Format | IETF RFC 2440: OpenPGP Message Format | http://www.ietf.org/rfc/rfc2440.txt |
| RFC 2821:Simple Mail Transfer Protocol | IETF RFC 2821: Simple Mail Transfer Protocol | http://www.ietf.org/rfc/rfc2821.txt |
| RFC 2822:Internet Message Format | IETF RFC 2822: Internet Message Format | http://www.ietf.org/rfc/rfc2822.txt |
| RFC 5531/4506 RPC & XDR | IETF RFC 5531 & 4506 | http://www.ietf.org/ |
| RFC 791:Internet Protocol | IETF RFC 791: Internet Protocol Specification | http://www.ietf.org/rfc/rfc791.txt |
| RPM Package Format | RPM Package Format V3.0 | http://www.rpm.org/max-rpm/s1-rpm-file-format-rpm-file-format.html |
| zlib Manual | zlib 1.2 Manual | http://www.gzip.org/zlib/ |

# **3 Requirements**

## **3.1 Relevant Libraries**

The libraries listed in [Table 3-1](#ID_TBL_45_GLSB_45_STDLIB) shall be available on a Linux Standard Base system, with the specified runtime names. The libraries listed in [Table 3-2](#ID_TBL_45_LSBGENERIC_45_ARCHLIB) are architecture specific, but shall be available on all LSB conforming systems. This list may be supplemented or amended by the relevant architecture specific part of the LSB Core Specification.

**Table 3-1 Standard Library Names**

| **Library** | **Runtime Name** |
| --- | --- |
| libcrypt | libcrypt.so.1 |
| libdl | libdl.so.2 |
| libgcc\_s | libgcc\_s.so.1 |
| libncurses | libncurses.so.5 |
| libncursesw | libncursesw.so.5 |
| libnspr4 | libnspr4.so |
| libnss3 | libnss3.so |
| libpam | libpam.so.0 |
| libpthread | libpthread.so.0 |
| librt | librt.so.1 |
| libssl3 | libssl3.so |
| libstdcxx | libstdc++.so.6 |
| libutil | libutil.so.1 |
| libz | libz.so.1 |

**Table 3-2 Standard Library Names defined in the Architecture Specific Parts of the LSB Core Specification**

| **Library** | **Runtime Name** |
| --- | --- |
| libc | See architecture specific part. |
| libm | See architecture specific part. |
| proginterp | See architecture specific part. |

These libraries will be in an implementation-defined directory which the dynamic linker shall search by default.

## **3.2 LSB Implementation Conformance**

A conforming implementation is necessarily architecture specific, and must provide the interfaces specified by both the generic LSB Core specification (LSB Core - Generic) and the relevant architecture specific part of the LSB Core Specification.

**Rationale:** An implementation must provide *at least* the interfaces specified in these specifications. It may also provide additional interfaces.

A conforming implementation shall satisfy the following requirements:

• A processor architecture represents a family of related processors which may not have identical feature sets. The architecture specific parts of the LSB Core Specification that supplement this specification for a given target processor architecture describe a minimum acceptable processor. The implementation shall provide all features of this processor, whether in hardware or through emulation transparent to the application.

• The implementation shall be capable of executing compiled applications having the format and using the system interfaces described in this specification.

• The implementation shall provide libraries containing the interfaces specified by this specification, and shall provide a dynamic linking mechanism that allows these interfaces to be attached to applications at runtime. All the interfaces shall behave as specified in this specification.

• The map of virtual memory provided by the implementation shall conform to the requirements of this specification.

• The implementation's low-level behavior with respect to function call linkage, system traps, signals, and other such activities shall conform to the formats described in this specification.

• The implementation shall provide all of the mandatory interfaces in their entirety.

• The implementation may provide one or more of the optional interfaces. Each optional interface that is provided shall be provided in its entirety. The product documentation shall state which optional interfaces are provided.

• The implementation shall provide all files and utilities specified as part of this specification in the format defined here and in other documents normatively included by reference. All commands and utilities shall behave as required by this specification. The implementation shall also provide all mandatory components of an application's runtime environment that are included or referenced in this specification.

• The implementation, when provided with standard data formats and values at a named interface, shall provide the behavior defined for those values and data formats at that interface. However, a conforming implementation may consist of components which are separately packaged and/or sold. For example, a vendor of a conforming implementation might sell the hardware, operating system, and windowing system as separately packaged items.

• The implementation may provide additional interfaces with different names. It may also provide additional behavior corresponding to data values outside the standard ranges, for standard named interfaces.

## **3.3 LSB Application Conformance**

A conforming application containing object files is necessarily architecture specific, and must conform to both the generic LSB Core specification (LSB Core - Generic) and the relevant architecture specific part of the LSB Core Specification. A conforming application which contains no object files may be architecture neutral. Architecture neutral applications shall conform only to the requirements of the generic LSB Core specification (LSB Core - Generic).

A conforming application shall satisfy the following requirements:

• Executable files shall be either object files in the format defined in the Object Format section of this specification, or script files in a scripting language where the interpreter is required by this specification.

• Object files shall participate in dynamic linking as defined in the Program Loading and Linking section of this specification.

• Object files shall employ only the instructions, traps, and other low-level facilities defined as being for use by applications in the Low-Level System Information section of this specification

• If the application requires any optional interface defined in this specification in order to be installed or to execute successfully, the requirement for that optional interface shall be stated in the application's documentation.

• The application shall not use any interface or data format that is not required to be provided by a conforming implementation, unless such an interface or data format is supplied by another application through direct invocation of that application during execution. The other application must also be a conforming application, and the use of such interface or data format, as well as its source (in other words, the other conforming application), shall be identified in the documentation of the application.

• The application shall not use any values for a named interface that are reserved for vendor extensions.

A strictly conforming application shall not require or use any interface, facility, or implementation-defined extension not defined in this specification in order to be installed or to execute successfully.

# **4 Terms and Definitions**

For the purposes of this document, the terms given in *ISO/IEC Directives, Part 2, Annex H* and the following apply.

archLSB

  Some LSB specification documents have both a generic, architecture-neutral part and an architecture-specific part. The latter describes elements whose definitions may be unique to a particular processor architecture. The term archLSB may be used in the generic part to refer to the corresponding section of the architecture-specific part.

Binary Standard, ABI

  The total set of interfaces that are available to be used in the compiled binary code of a conforming application, including the run-time details such as calling conventions, binary format, C++ name mangling, etc.

Implementation-defined

  Describes a value or behavior that is not defined by this document but is selected by an implementor. The value or behavior may vary among implementations that conform to this document. An application should not rely on the existence of the value or behavior. An application that relies on such a value or behavior cannot be assured to be portable across conforming implementations. The implementor shall document such a value or behavior so that it can be used correctly by an application.

Shell Script

  A file that is read by an interpreter (e.g., awk). The first line of the shell script includes a reference to its interpreter binary.

Source Standard, API

  The total set of interfaces that are available to be used in the source code of a conforming application. Due to translations, the Binary Standard and the Source Standard may contain some different interfaces.

Undefined

  Describes the nature of a value or behavior not defined by this document which results from use of an invalid program construct or invalid data input. The value or behavior may vary among implementations that conform to this document. An application should not rely on the existence or validity of the value or behavior. An application that relies on any particular value or behavior cannot be assured to be portable across conforming implementations.

Unspecified

  Describes the nature of a value or behavior not specified by this document which results from use of a valid program construct or valid data input. The value or behavior may vary among implementations that conform to this document. An application should not rely on the existence or validity of the value or behavior. An application that relies on any particular value or behavior cannot be assured to be portable across conforming implementations.

In addition, for the portions of this specification which build on IEEE Std 1003.1-2001, the definitions given in *IEEE Std 1003.1-2001, Base Definitions, Chapter 3* apply.

# **5 Documentation Conventions**

Throughout this document, the following typographic conventions are used:

function()

  the name of a function

**command**

  the name of a command or utility

CONSTANT

  a constant value

*parameter*

  a parameter

variable

  a variable

Throughout this specification, several tables of interfaces are presented. Each entry in these tables has the following format:

name

  the name of the interface

(symver)

  An optional symbol version identifier, if required.

[*refno*]

  A reference number indexing the table of referenced specifications that follows this table.

For example,

|  |
| --- |
| forkpty(GLIBC\_2.0) [SUSv4] |

refers to the interface named forkpty() with symbol version GLIBC\_2.0 that is defined in the reference indicated by the tag SUSv4.

**Note:** For symbols with versions which differ between architectures, the symbol versions are defined in the architecture specific parts of of this module specification only. In the generic part, they will appear without symbol versions.

# **6 Relationship To ISO/IEC 9945 POSIX**

This specification includes many interfaces described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4). Unless otherwise specified, such interfaces should behave exactly as described in that specification. Any conflict between the requirements described here and the [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) standard is unintentional, except as explicitly noted otherwise.

**Note:** In addition to the differences noted in this specification, a report, *ISO/IEC TR 24715-Technical Report on the Conflicts Between the ISO/IEC 9945 (POSIX) Standard and the Linux Standard Base Specification (LSB)*, identifies the differences between edition 3.1 of this specification and *POSIX 1003.1-2001 (ISO/IEC 9945-2003)* (more precisely, POSIX 2001 plus the first two corrigenda, informally known as the 2004 edition). It is the long term plan of the Linux Foundation to converge the LSB Core specification with the ISO/IEC POSIX specification.

The LSB Specification Authority is responsible for deciding the meaning of conformance to normative referenced standards in the LSB context. Problem reports regarding underlying or referenced standards in any other context will be referred to the relevant maintenance body for that standard.

# **7 Relationship To Other Linux Foundation Specifications**

The LSB is the base for several other specification projects under the umbrella of the Linux Foundation (LF). This specification is the foundation, and other specifications build on the interfaces defined here. However, beyond those specifications listed as Normative References, this specification has no dependencies on other LF projects.

ISO/IEC 23360 corresponds to an earlier edition of this specification (version 3.1), published as an ISO/IEC standard in 2006 after submission by the Linux Foundation. The ISO edition is also the subject of the technical report ISO/IEC TR 24715 referenced in the previous chapter.

# **II Executable And Linking Format (ELF)**

# **8 Introduction**

Executable and Linking Format (ELF) defines the object format for compiled applications. This specification supplements the information found in [System V ABI Update](#ID_STD_46_ABIUPDATE) and is intended to document additions made since the publication of that document.

# **9 Low Level System Information**

## **9.1 Operating System Interface**

LSB-conforming applications shall assume that stack, heap and other allocated memory regions will be non-executable. The application must take steps to make them executable if needed.

## **9.2 Machine Interface**

### **9.2.1 Data Representation**

LSB-conforming applications shall use the data representation as defined in the Architecture specific ELF documents.

#### 9.2.1.1 Fundamental Types

In addition to the fundamental types specified in the relevant architecture specific part of the LSB Core Specification, a 1 byte data type is defined here.

**Table 9-1 Scalar Types**

| **Type** | **C** | **C++** | **sizeof** | **Alignment (bytes)** | **Architecture Representation** |
| --- | --- | --- | --- | --- | --- |
| Integral | \_Bool | bool | 1 | 1 | byte |
|  |  |
|  |  |  |  |  |

# **10 Object Format**

## **10.1 Object Files**

LSB-conforming implementations shall support the Executable and Linking Format (ELF) object file format as defined by the following documents:

• [System V ABI](#ID_STD_46_GABI41)

• [System V ABI Update](#ID_STD_46_ABIUPDATE)

• the relevant architecture specific ABI supplement.

• this specification

• the relevant architecture specific part of the LSB Core Specification

Conforming implementations may also support other unspecified object file formats.

## **10.2 Sections**

### **10.2.1 Introduction**

As described in [System V ABI](#ID_STD_46_GABI41), an ELF object file contains a number of *sections*.

### **10.2.2 Sections Types**

The section header table is an array of Elf32\_Shdr or Elf64\_Shdr structures as described in [System V ABI](#ID_STD_46_GABI41). The *sh\_type* member shall be either a value from [Table 10-1](#ID_TBL_46_ABIUPDATE), drawn from the System V ABI, or one of the additional values specified in [Table 10-2](#ID_TBL_46_LSB).

A section header's *sh\_type* member specifies the sections's semantics.

#### 10.2.2.1 ELF Section Types

The following section types are defined in the [System V ABI](#ID_STD_46_GABI41) and the [System V ABI Update](#ID_STD_46_ABIUPDATE).

**Table 10-1 ELF Section Types**

| **Name** | **Value** | **Description** |
| --- | --- | --- |
| SHT\_DYNAMIC | 0x6 | The section holds information for dynamic linking. Currently, an object file shall have only one dynamic section, but this restriction may be relaxed in the future. See `Dynamic Section' in Chapter 5 of System V ABI Update for details. |
| SHT\_DYNSYM | 0xb | This section holds a minimal set of symbols adequate for dynamic linking. See also SHT\_SYMTAB. Currently, an object file may have either a section of SHT\_SYMTAB type or a section of SHT\_DYNSYM type, but not both. This restriction may be relaxed in the future. |
| SHT\_FINI\_ARRAY | 0xf | This section contains an array of pointers to termination functions, as described in `Initialization and Termination Functions' in Chapter 5 of System V ABI Update. Each pointer in the array is taken as a parameterless procedure with a void return. |
| SHT\_HASH | 0x5 | The section holds a symbol hash table. Currently, an object file shall have only one hash table, but this restriction may be relaxed in the future. See `Hash Table' in Chapter 5 of System V ABI Update for details. |
| SHT\_INIT\_ARRAY | 0xe | This section contains an array of pointers to initialization functions, as described in `Initialization and Termination Functions' in Chapter 5 of System V ABI Update. Each pointer in the array is taken as a parameterless procedure with a void return. |
| SHT\_NOBITS | 0x8 | A section of this type occupies no space in the file but otherwise resembles SHT\_PROGBITS. Although this section contains no bytes, the sh\_offset member contains the conceptual file offset. |
| SHT\_NOTE | 0x7 | The section holds information that marks the file in some way. See `Note Section' in Chapter 5 of System V ABI Update for details. |
| SHT\_NULL | 0x0 | This value marks the section header as inactive; it does not have an associated section. Other members of the section header have undefined values. |
| SHT\_PREINIT\_ARRAY | 0x10 | This section contains an array of pointers to functions that are invoked before all other initialization functions, as described in `Initialization and Termination Functions' in Chapter 5 of System V ABI Update. Each pointer in the array is taken as a parameterless proceure with a void return. |
| SHT\_PROGBITS | 0x1 | The section holds information defined by the program, whose format and meaning are determined solely by the program. |
| SHT\_REL | 0x9 | The section holds relocation entries without explicit addends, such as type Elf32\_Rel for the 32-bit class of object files or type Elf64\_Rel for the 64-bit class of object files. An object file may have multiple relocation sections. See `Relocation' in Chapter 4 of System V ABI Update for details. |
| SHT\_RELA | 0x4 | The section holds relocation entries with explicit addends, such as type Elf32\_Rela for the 32-bit class of object files or type Elf64\_Rela for the 64-bit class of object files. An object file may have multiple relocation sections. See `Relocation' in Chapter 4 of System V ABI Update for details. |
| SHT\_STRTAB | 0x3 | The section holds a string table. An object file may have multiple string table sections. See `String Table' in Chapter 4 of System V ABI Update for details. |
| SHT\_SYMTAB | 0x2 | This section holds a symbol table. Currently, an object file may have either a section of SHT\_SYMTAB type or a section of SHT\_DYNSYM type, but not both. This restriction may be relaxed in the future. Typically, SHT\_SYMTAB provides symbols for link editing, though it may also be used for dynamic linking. As a complete symbol table, it may contain many symbols unnecessary for dynamic linking. |

#### 10.2.2.2 Additional Section Types

The following additional section types are defined here.

**Table 10-2 Additional Section Types**

| **Name** | **Value** | **Description** |
| --- | --- | --- |
| SHT\_GNU\_verdef | 0x6ffffffd | This section contains the symbol versions that are provided. |
| SHT\_GNU\_verneed | 0x6ffffffe | This section contains the symbol versions that are required. |
| SHT\_GNU\_versym | 0x6fffffff | This section contains the Symbol Version Table. |

## **10.3 Special Sections**

### **10.3.1 Special Sections**

Various sections hold program and control information. Sections in the lists below are used by the system and have the indicated types and attributes.

#### 10.3.1.1 ELF Special Sections

The following sections are defined in the [System V ABI](#ID_STD_46_GABI41) and the [System V ABI Update](#ID_STD_46_ABIUPDATE).

**Table 10-3 ELF Special Sections**

| **Name** | **Type** | **Attributes** |
| --- | --- | --- |
| .bss | SHT\_NOBITS | SHF\_ALLOC+SHF\_WRITE |
| .comment | SHT\_PROGBITS | SHF\_MERGE+SHF\_STRINGS |
| .data | SHT\_PROGBITS | SHF\_ALLOC+SHF\_WRITE |
| .data1 | SHT\_PROGBITS | SHF\_ALLOC+SHF\_WRITE |
| .debug | SHT\_PROGBITS | 0 |
| .dynamic | SHT\_DYNAMIC | SHF\_ALLOC+SHF\_WRITE |
| .dynstr | SHT\_STRTAB | SHF\_ALLOC |
| .dynsym | SHT\_DYNSYM | SHF\_ALLOC |
| .fini | SHT\_PROGBITS | SHF\_ALLOC+SHF\_EXECINSTR |
| .fini\_array | SHT\_FINI\_ARRAY | SHF\_ALLOC+SHF\_WRITE |
| .hash | SHT\_HASH | SHF\_ALLOC |
| .init | SHT\_PROGBITS | SHF\_ALLOC+SHF\_EXECINSTR |
| .init\_array | SHT\_INIT\_ARRAY | SHF\_ALLOC+SHF\_WRITE |
| .interp | SHT\_PROGBITS | SHF\_ALLOC |
| .line | SHT\_PROGBITS | 0 |
| .note | SHT\_NOTE | 0 |
| .preinit\_array | SHT\_PREINIT\_ARRAY | SHF\_ALLOC+SHF\_WRITE |
| .rodata | SHT\_PROGBITS | SHF\_ALLOC+SHF\_MERGE+SHF\_STRINGS |
| .rodata1 | SHT\_PROGBITS | SHF\_ALLOC+SHF\_MERGE+SHF\_STRINGS |
| .shstrtab | SHT\_STRTAB | 0 |
| .strtab | SHT\_STRTAB | SHF\_ALLOC |
| .symtab | SHT\_SYMTAB | SHF\_ALLOC |
| .tbss | SHT\_NOBITS | SHF\_ALLOC+SHF\_WRITE+SHF\_TLS |
| .tdata | SHT\_PROGBITS | SHF\_ALLOC+SHF\_WRITE+SHF\_TLS |
| .text | SHT\_PROGBITS | SHF\_ALLOC+SHF\_EXECINSTR |

.bss

  This section holds data that contributes to the program's memory image. The program may treat this data as uninitialized. However, the system shall initialize this data with zeroes when the program begins to run. The section occupies no file space, as indicated by the section type, SHT\_NOBITS.

.comment

  This section holds version control information.

.data

  This section holds initialized data that contribute to the program's memory image.

.data1

  This section holds initialized data that contribute to the program's memory image.

.debug

  This section holds information for symbolic debugging. The contents are unspecified. All section names with the prefix .debug hold information for symbolic debugging. The contents of these sections are unspecified.

.dynamic

  This section holds dynamic linking information. The section's attributes will include the SHF\_ALLOC bit. Whether the SHF\_WRITE bit is set is processor specific. See Chapter 5 of System V ABI Update for more information.

.dynstr

  This section holds strings needed for dynamic linking, most commonly the strings that represent the names associated with symbol table entries. See Chapter 5 of System V ABI Update for more information.

.dynsym

  This section holds the dynamic linking symbol table, as described in `Symbol Table' of System V ABI Update.

.fini

  This section holds executable instructions that contribute to the process termination code. That is, when a program exits normally, the system arranges to execute the code in this section.

.fini\_array

  This section holds an array of function pointers that contributes to a single termination array for the executable or shared object containing the section.

.hash

  This section holds a symbol hash table. See `Hash Table' in Chapter 5 of System V ABI Update for more information.

.init

  This section holds executable instructions that contribute to the process initialization code. When a program starts to run, the system arranges to execute the code in this section before calling the main program entry point (called main for C programs).

.init\_array

  This section holds an array of function pointers that contributes to a single initialization array for the executable or shared object containing the section.

.interp

  This section holds the path name of a program interpreter. If the file has a loadable segment that includes relocation, the sections' attributes will include the SHF\_ALLOC bit; otherwise, that bit will be off. See Chapter 5 of System V ABI Update for more information.

.line

  This section holds line number information for symbolic debugging, which describes the correspondence between the source program and the machine code. The contents are unspecified.

.note

  This section holds information in the format that `Note Section' in Chapter 5 of System V ABI Update describes.

.preinit\_array

  This section holds an array of function pointers that contributes to a single pre-initialization array for the executable or shared object containing the section.

.rodata

  This section holds read-only data that typically contribute to a non-writable segment in the process image. See `Program Header' in Chapter 5 of System V ABI Update for more information.

.rodata1

  This section holds read-only data that typically contribute to a non-writable segment in the process image. See `Program Header' in Chapter 5 of System V ABI Update for more information.

.shstrtab

  This section holds section names.

.strtab

  This section holds strings, most commonly the strings that represent the names associated with symbol table entries. If the file has a loadable segment that includes the symbol string table, the section's attributes will include the SHF\_ALLOC bit; otherwise, that bit will be off.

.symtab

  This section holds a symbol table, as `Symbol Table' in Chapter 4 of System V ABI Update describes. If the file has a loadable segment that includes the symbol table, the section's attributes will include the SHF\_ALLOC bit; otherwise, that bit will be off.

.tbss

  This section holds uninitialized thread-local data that contribute to the program's memory image. By definition, the system initializes the data with zeros when the data is instantiated for each new execution flow. The section occupies no file space, as indicated by the section type, SHT\_NOBITS. Implementations need not support thread-local storage.

.tdata

  This section holds initialized thread-local data that contributes to the program's memory image. A copy of its contents is instantiated by the system for each new execution flow. Implementations need not support thread-local storage.

.text

  This section holds the `text', or executable instructions, of a program.

#### 10.3.1.2 Additional Special Sections

Object files in an LSB conforming application may also contain one or more of the additional special sections described below.

**Table 10-4 Additional Special Sections**

| **Name** | **Type** | **Attributes** |
| --- | --- | --- |
| .ctors | SHT\_PROGBITS | SHF\_ALLOC+SHF\_WRITE |
| .data.rel.ro | SHT\_PROGBITS | SHF\_ALLOC+SHF\_WRITE |
| .dtors | SHT\_PROGBITS | SHF\_ALLOC+SHF\_WRITE |
| .eh\_frame | SHT\_PROGBITS | SHF\_ALLOC |
| .eh\_frame\_hdr | SHT\_PROGBITS | SHF\_ALLOC |
| .gcc\_except\_table | SHT\_PROGBITS | SHF\_ALLOC |
| .gnu.version | SHT\_GNU\_versym | SHF\_ALLOC |
| .gnu.version\_d | SHT\_GNU\_verdef | SHF\_ALLOC |
| .gnu.version\_r | SHT\_GNU\_verneed | SHF\_ALLOC |
| .got.plt | SHT\_PROGBITS | SHF\_ALLOC+SHF\_WRITE |
| .jcr | SHT\_PROGBITS | SHF\_ALLOC+SHF\_WRITE |
| .note.ABI-tag | SHT\_NOTE | SHF\_ALLOC |
| .stab | SHT\_PROGBITS | 0 |
| .stabstr | SHT\_STRTAB | 0 |

.ctors

  This section contains a list of global constructor function pointers.

.data.rel.ro

  This section holds initialized data that contribute to the program's memory image. This section may be made read-only after relocations have been applied.

.dtors

  This section contains a list of global destructor function pointers.

.eh\_frame

  This section contains information necessary for frame unwinding during exception handling. See [Section 10.6.1](#ID_EHFRAME).

.eh\_frame\_hdr

  This section contains a pointer to the .eh\_frame section which is accessible to the runtime support code of a C++ application. This section may also contain a binary search table which may be used by the runtime support code to more efficiently access records in the .eh\_frame section. See [Section 10.6.2](#ID_EHFRAMEHDR).

.gcc\_except\_table

  This section holds Language Specific Data.

.gnu.version

  This section contains the Symbol Version Table. See [Section 10.7.2](#ID_SYMVERTBL).

.gnu.version\_d

  This section contains the Version Definitions. See [Section 10.7.3](#ID_SYMVERDEFS).

.gnu.version\_r

  This section contains the Version Requirements. See [Section 10.7.4](#ID_SYMVERRQMTS).

.got.plt

  This section holds the read-only portion of the GLobal Offset Table. This section may be made read-only after relocations have been applied.

.jcr

  This section contains information necessary for registering compiled Java classes. The contents are compiler-specific and used by compiler initialization functions.

.note.ABI-tag

  Specify ABI details. See [Section 10.8](#ID_NOTEABITAG).

.stab

  This section contains debugging information. The contents are not specified as part of the LSB.

.stabstr

  This section contains strings associated with the debugging infomation contained in the .stab section.

## **10.4 Symbol Mapping**

### **10.4.1 Introduction**

Symbols in a source program are translated by the compilation system into symbols that exist in the object file.

#### 10.4.1.1 C Language

External C symbols shall be unchanged in an object file's symbol table.

## **10.5 DWARF Extensions**

**10.5.1 General**

The LSB does not specify debugging information, however, some additional sections contain information which is encoded using the the encoding as specified by [DWARF Debugging Information Format, Version 4](#ID_STD_46_DWARF4) with extensions defined here.

### **10.5.2 DWARF Exception Header Encoding**

The DWARF Exception Header Encoding is used to describe the type of data used in the .eh\_frame and .eh\_frame\_hdr section. The upper 4 bits indicate how the value is to be applied. The lower 4 bits indicate the format of the data.

**Table 10-5 DWARF Exception Header value format**

| **Name** | **Value** | **Meaning** |
| --- | --- | --- |
| DW\_EH\_PE\_absptr | 0x00 | The Value is a literal pointer whose size is determined by the architecture. |
| DW\_EH\_PE\_uleb128 | 0x01 | Unsigned value is encoded using the Little Endian Base 128 (LEB128) as defined by [DWARF Debugging Information Format, Version 4](#ID_STD_46_DWARF4). |
| DW\_EH\_PE\_udata2 | 0x02 | A 2 bytes unsigned value. |
| DW\_EH\_PE\_udata4 | 0x03 | A 4 bytes unsigned value. |
| DW\_EH\_PE\_udata8 | 0x04 | An 8 bytes unsigned value. |
| DW\_EH\_PE\_sleb128 | 0x09 | Signed value is encoded using the Little Endian Base 128 (LEB128) as defined by [DWARF Debugging Information Format, Version 4](#ID_STD_46_DWARF4). |
| DW\_EH\_PE\_sdata2 | 0x0A | A 2 bytes signed value. |
| DW\_EH\_PE\_sdata4 | 0x0B | A 4 bytes signed value. |
| DW\_EH\_PE\_sdata8 | 0x0C | An 8 bytes signed value. |

**Table 10-6 DWARF Exception Header application**

| **Name** | **Value** | **Meaning** |
| --- | --- | --- |
| DW\_EH\_PE\_pcrel | 0x10 | Value is relative to the current program counter. |
| DW\_EH\_PE\_textrel | 0x20 | Value is relative to the beginning of the .text section. |
| DW\_EH\_PE\_datarel | 0x30 | Value is relative to the beginning of the .got or .eh\_frame\_hdr section. |
| DW\_EH\_PE\_funcrel | 0x40 | Value is relative to the beginning of the function. |
| DW\_EH\_PE\_aligned | 0x50 | Value is aligned to an address unit sized boundary. |

One special encoding, 0xff (DW\_EH\_PE\_omit), shall be used to indicate that no value ispresent.

### **10.5.3 DWARF CFI Extensions**

In addition to the Call Frame Instructions defined in section 6.4.2 of [DWARF Debugging Information Format, Version 4](#ID_STD_46_DWARF4), the following additional Call Frame Instructions may also be used.

**Table 10-7 Additional DWARF Call Frame Instructions**

| **Name** | **Value** | **Meaning** |
| --- | --- | --- |
| DW\_CFA\_GNU\_args\_size | 0x2e | The DW\_CFA\_GNU\_args\_size instruction takes an unsigned LEB128 operand representing an argument size. This instruction specifies the total of the size of the arguments which have been pushed onto the stack. |
| DW\_CFA\_GNU\_negative\_offset\_extended | 0x2f | The DW\_CFA\_def\_cfa\_sf instruction takes two operands: an unsigned LEB128 value representing a register number and an unsigned LEB128 which represents the magnitude of the offset. This instruction is identical to DW\_CFA\_offset\_extended\_sf except that the operand is subtracted to produce the offset. This instructions is obsoleted by DW\_CFA\_offset\_extended\_sf. |

## **10.6 Exception Frames**

**10.6.1 General**

When using languages that support exceptions, such as C++, additional information must be provided to the runtime environment that describes the call frames that must be unwound during the processing of an exception. This information is contained in the special sections .eh\_frame and .eh\_framehdr.

**Note:** The format of the .eh\_frame section is similar in format and purpose to the .debug\_frame section which is specified in [DWARF Debugging Information Format, Version 4](#ID_STD_46_DWARF4). Readers are advised that there are some subtle difference, and care should be taken when comparing the two sections.

### **10.6.2 The .eh\_frame section**

The .eh\_frame section shall contain 1 or more Call Frame Information (CFI) records. The number of records present shall be determined by size of the section as contained in the section header. Each CFI record contains a Common Information Entry (CIE) record followed by 1 or more Frame Description Entry (FDE) records. Both CIEs and FDEs shall be aligned to an addressing unit sized boundary.

**Table 10-8 Call Frame Information Format**

|  |
| --- |
| Common Information Entry Record |
| Frame Description Entry Record(s) |

#### 10.6.2.1 The Common Information Entry Format

**Table 10-9 Common Information Entry Format**

|  |  |
| --- | --- |
| Length | Required |
| Extended Length | Optional |
| CIE ID | Required |
| Version | Required |
| Augmentation String | Required |
| Code Alignment Factor | Required |
| Data Alignment Factor | Required |
| Return Address Register | Required |
| Augmentation Data Length | Optional |
| Augmentation Data | Optional |
| Initial Instructions | Required |
| Padding |  |

*Length*

  A 4 byte unsigned value indicating the length in bytes of the CIE structure, not including the *Length* field itself. If *Length* contains the value 0xffffffff, then the length is contained in the *Extended Length* field. If *Length* contains the value 0, then this CIE shall be considered a terminator and processing shall end.

*Extended Length*

  A 8 byte unsigned value indicating the length in bytes of the CIE structure, not including the *Length* and *Extended Length* fields themselves. This field is not present unless the *Length* field contains the value 0xffffffff.

*CIE ID*

  A 4 byte unsigned value that is used to distinguish CIE records from FDE records. This value shall always be 0, which indicates this record is a CIE.

*Version*

  A 1 byte value that identifies the version number of the frame information structure. This value shall be 1.

*Augmentation String*

  This value is a NUL terminated string that identifies the augmentation to the CIE or to the FDEs associated with this CIE. A zero length string indicates that no augmentation data is present. The augmentation string is case sensitive and shall be interpreted as described below.

*Code Alignment Factor*

  An unsigned LEB128 encoded value that is factored out of all advance location instructions that are associated with this CIE or its FDEs. This value shall be multiplied by the delta argument of an adavance location instruction to obtain the new location value.

*Data Alignment Factor*

  A signed LEB128 encoded value that is factored out of all offset instructions that are associated with this CIE or its FDEs. This value shall be multiplied by the register offset argument of an offset instruction to obtain the new offset value.

*Augmentation Length*

  An unsigned LEB128 encoded value indicating the length in bytes of the Augmentation Data. This field is only present if the Augmentation String contains the character 'z'.

*Augmentation Data*

  A block of data whose contents are defined by the contents of the Augmentation String as described below. This field is only present if the Augmentation String contains the character 'z'. The size of this data is given by the Augentation Length.

*Initial Instructions*

  Initial set of Call Frame Instructions. The number of instructions is determined by the remaining space in the CIE record.

*Padding*

  Extra bytes to align the CIE structure to an addressing unit size boundary.

##### *10.6.2.1.1 Augmentation String Format*

The Agumentation String indicates the presence of some optional fields, and how those fields should be intepreted. This string is case sensitive. Each character in the augmentation string in the CIE can be interpreted as below:

'z'

  A 'z' may be present as the first character of the string. If present, the Augmentation Data field shall be present. The contents of the Augmentation Data shall be intepreted according to other characters in the Augmentation String.

'L'

  A 'L' may be present at any position after the first character of the string. This character may only be present if 'z' is the first character of the string. If present, it indicates the presence of one argument in the Augmentation Data of the CIE, and a corresponding argument in the Augmentation Data of the FDE. The argument in the Augmentation Data of the CIE is 1-byte and represents the pointer encoding used for the argument in the Augmentation Data of the FDE, which is the address of a language-specific data area (LSDA). The size of the LSDA pointer is specified by the pointer encoding used.

'P'

  A 'P' may be present at any position after the first character of the string. This character may only be present if 'z' is the first character of the string. If present, it indicates the presence of two arguments in the Augmentation Data of the CIE. The first argument is 1-byte and represents the pointer encoding used for the second argument, which is the address of a *personality routine* handler. The personality routine is used to handle language and vendor-specific tasks. The system unwind library interface accesses the language-specific exception handling semantics via the pointer to the personality routine. The personality routine does not have an ABI-specific name. The size of the personality routine pointer is specified by the pointer encoding used.

'R'

  A 'R' may be present at any position after the first character of the string. This character may only be present if 'z' is the first character of the string. If present, The Augmentation Data shall include a 1 byte argument that represents the pointer encoding for the address pointers used in the FDE.

#### 10.6.2.2 The Frame Description Entry Format

**Table 10-10 Frame Description Entry Format**

|  |  |
| --- | --- |
| Length | Required |
| Extended Length | Optional |
| FDE Pointer | Required |
| PC Begin | Required |
| PC Range | Required |
| Augmentation Data Length | Optional |
| Augmentation Data | Optional |
| Call Frame Instructions | Required |
| Padding |  |

*Length*

  A 4 byte unsigned value indicating the length in bytes of the FDE structure, not including the *Length* field itself. If *Length* contains the value 0xffffffff, then the length is contained the *Extended Length* field. If *Length* contains the value 0, then this FDE shall be considered a terminator and processing shall end.

*Extended Length*

  A 8 byte unsigned value indicating the length in bytes of the FDE structure, not including the *Length* or *Extended Length* field themselves. This field is not present unless the *Length* field contains the value 0xffffffff.

*CIE Pointer*

  A 4 byte unsigned value that when subtracted from the offset of the CIE Pointer in the current FDE yields the offset of the start of the associated CIE. This value shall never be 0.

*PC Begin*

  An encoded value that indicates the address of the initial location associated with this FDE. The encoding format is specified in the Augmentation Data.

*PC Range*

  An absolute value that indicates the number of bytes of instructions associated with this FDE.

*Augmentation Length*

  An unsigned LEB128 encoded value indicating the length in bytes of the Augmentation Data. This field is only present if the Augmentation String in the associated CIE contains the character 'z'.

*Augmentation Data*

  A block of data whose contents are defined by the contents of the Augmentation String in the associated CIE as described above. This field is only present if the Augmentation String in the associated CIE contains the character 'z'. The size of this data is given by the Augentation Length.

*Call Frame Instructions*

  A set of Call Frame Instructions.

*Padding*

  Extra bytes to align the FDE structure to an addressing unit size boundary.

### **10.6.2 The .eh\_frame\_hdr section**

The .eh\_frame\_hdr section contains additional information about the .eh\_frame section. A pointer to the start of the .eh\_frame data, and optionally, a binary search table of pointers to the .eh\_frame records are found in this section.

Data in this section is encoded according to [Section 10.5.1](#ID_DWARFEHENCODING).

**Table 10-11 .eh\_frame\_hdr Section Format**

| **Encoding** | **Field** |
| --- | --- |
| unsigned byte | version |
| unsigned byte | eh\_frame\_ptr\_enc |
| unsigned byte | fde\_count\_enc |
| unsigned byte | table\_enc |
| encoded | eh\_frame\_ptr |
| encoded | fde\_count |
|  | binary search table |

version

  Version of the .eh\_frame\_hdr format. This value shall be 1.

eh\_frame\_ptr\_enc

  The encoding format of the eh\_frame\_ptr field.

fde\_count\_enc

  The encoding format of the fde\_count field. A value of DW\_EH\_PE\_omit indicates the binary search table is not present.

table\_enc

  The encoding format of the entries in the binary search table. A value of DW\_EH\_PE\_omit indicates the binary search table is not present.

eh\_frame\_ptr

  The encoded value of the pointer to the start of the .eh\_frame section.

fde\_count

  The encoded value of the count of entries in the binary search table.

binary search table

  A binary search table containing fde\_count entries. Each entry of the table consist of two encoded values, the initial location, and the address. The entries are sorted in an increasing order by the initial location value.

## **10.7 Symbol Versioning**

### **10.7.1 Introduction**

This chapter describes the Symbol Versioning mechanism. All ELF objects may provide or depend on versioned symbols. Symbol Versioning is implemented by 3 section types: SHT\_GNU\_versym, SHT\_GNU\_verdef, and SHT\_GNU\_verneed.

The prefix Elfxx in the following descriptions and code fragments stands for either "Elf32" or "Elf64", depending on the architecture.

Versions are described by strings. The structures that are used for symbol versions also contain a member that holds the ELF hashing values of the strings. This allows for more efficient processing.

### **10.7.2 Symbol Version Table**

The special section .gnu.version which has a section type of SHT\_GNU\_versym shall contain the Symbol Version Table. This section shall have the same number of entries as the Dynamic Symbol Table in the .dynsym section.

The .gnu.version section shall contain an array of elements of type Elfxx\_Half. Each entry specifies the version defined for or required by the corresponding symbol in the Dynamic Symbol Table.

The values in the Symbol Version Table are specific to the object in which they are located. These values are identifiers that are provided by the the *vna\_other* member of the Elfxx\_Vernaux structure or the *vd\_ndx* member of the Elfxx\_Verdef structure.

The values 0 and 1 are reserved.

0

  The symbol is local, not available outside the object.

1

  The symbol is defined in this object and is globally available.

All other values are used to identify version strings located in one of the other Symbol Version sections. The value itself is not the version associated with the symbol. The string identified by the value defines the version of the symbol.

### **10.7.3 Version Definitions**

The special section .gnu.version\_d which has a section type of SHT\_GNU\_verdef shall contain symbol version definitions. The number of entries in this section shall be contained in the DT\_VERDEFNUM entry of the Dynamic Section .dynamic. The sh\_link member of the section header (see figure 4-8 in the [System V ABI](#ID_STD_46_GABI41)) shall point to the section that contains the strings referenced by this section.

The section shall contain an array of Elfxx\_Verdef structures, as described in [Figure 10-1](#ID_VERDEFENTRIES), optionally followed by an array of Elfxx\_Verdaux structures, as defined in [Figure 10-2](#ID_VERDEFEXTS).

|  |
| --- |
| typedef struct {  Elfxx\_Half vd\_version;  Elfxx\_Half vd\_flags;  Elfxx\_Half vd\_ndx;  Elfxx\_Half vd\_cnt;  Elfxx\_Word vd\_hash;  Elfxx\_Word vd\_aux;  Elfxx\_Word vd\_next;  } Elfxx\_Verdef; |

Figure **10‑1 Version Definition Entries**

*vd\_version*

  Version revision. This field shall be set to 1.

*vd\_flags*

  Version information flag bitmask.

*vd\_ndx*

  Version index numeric value referencing the SHT\_GNU\_versym section.

*vd\_cnt*

  Number of associated verdaux array entries.

*vd\_hash*

  Version name hash value (ELF hash function).

*vd\_aux*

  Offset in bytes to a corresponding entry in an array of Elfxx\_Verdaux structures as defined in [Figure 10-2](#ID_VERDEFEXTS)

*vd\_next*

  Offset to the next verdef entry, in bytes.

|  |
| --- |
| typedef struct {  Elfxx\_Word vda\_name;  Elfxx\_Word vda\_next;  } Elfxx\_Verdaux; |

Figure 10-2 Version Definition Auxiliary

*vda\_name*

  Offset to the version or dependency name string in the section header, in bytes.

*vda\_next*

  Offset to the next verdaux entry, in bytes.

### **10.7.4 Version Requirements**

The special section .gnu.version\_r which has a section type of SHT\_GNU\_verneed shall contain required symbol version definitions. The number of entries in this section shall be contained in the DT\_VERNEEDNUM entry of the Dynamic Section .dynamic. The *sh\_link* member of the section header (see figure 4-8 in [System V ABI](#ID_STD_46_GABI41)) shall point to the section that contains the strings referenced by this section.

The section shall contain an array of Elfxx\_Verneed structures, as described in [Figure 10-3](#ID_VERNEEDFIG), optionally followed by an array of Elfxx\_Vernaux structures, as defined in [Figure 10-4](#ID_VERNEEDEXTFIG).

|  |
| --- |
| typedef struct {  Elfxx\_Half vn\_version;  Elfxx\_Half vn\_cnt;  Elfxx\_Word vn\_file;  Elfxx\_Word vn\_aux;  Elfxx\_Word vn\_next;  } Elfxx\_Verneed; |

Figure 10-3 Version Needed Entries

*vn\_version*

  Version of structure. This value is currently set to 1, and will be reset if the versioning implementation is incompatibly altered.

*vn\_cnt*

  Number of associated verneed array entries.

*vn\_file*

  Offset to the file name string in the section header, in bytes.

*vn\_aux*

  Offset to a corresponding entry in the vernaux array, in bytes.

*vn\_next*

  Offset to the next verneed entry, in bytes.

|  |
| --- |
| typedef struct {  Elfxx\_Word vna\_hash;  Elfxx\_Half vna\_flags;  Elfxx\_Half vna\_other;  Elfxx\_Word vna\_name;  Elfxx\_Word vna\_next;  } Elfxx\_Vernaux; |

Figure 10-4 Version Needed Auxiliary Ent

*vna\_hash*

  Dependency name hash value (ELF hash function).

*vna\_flags*

  Dependency information flag bitmask.

*vna\_other*

  Object file version identifier used in the .gnu.version symbol version array. Bit number 15 controls whether or not the object is hidden; if this bit is set, the object cannot be used and the static linker will ignore the symbol's presence in the object.

*vna\_name*

  Offset to the dependency name string in the section header, in bytes.

*vna\_next*

  Offset to the next vernaux entry, in bytes.

### **10.7.5 Startup Sequence**

When loading a sharable object the system shall analyze version definition data from the loaded object to assure that it meets the version requirements of the calling object. This step is referred to as definition testing. The dynamic loader shall retrieve the entries in the caller's Elfxx\_Verneed array and attempt to find matching definition information in the loaded Elfxx\_Verdef table.

Each object and dependency shall be tested in turn. If a symbol definition is missing and the vna\_flags bit for VER\_FLG\_WEAK is not set, the loader shall return an error and exit. If the vna\_flags bit for VER\_FLG\_WEAK is set in the Elfxx\_Vernaux entry, and the loader shall issue a warning and continue operation.

When the versions referenced by undefined symbols in the loaded object are found, version availability is certified. The test completes without error and the object shall be made available.

### **10.7.6 Symbol Resolution**

When symbol versioning is used in an object, relocations extend definition testing beyond the simple match of symbol name strings: the version of the reference shall also equal the name of the definition.

The same index that is used in the symbol table can be referenced in the SHT\_GNU\_versym section, and the value of this index is then used to acquire name data. The corresponding requirement string is retrieved from the Elfxx\_Verneed array, and likewise, the corresponding definition string from the Elfxx\_Verdef table.

If the high order bit (bit number 15) of the version symbolis set, the object cannot be used and the static linker shall ignore the symbol's presence in the object.

When an object with a reference and an object with the definition are being linked, the following rules shall govern the result:

• The object with the reference and the object with the definitions both use versioning. All described matching is processed in this case. A fatal error shall be triggered when no matching definition can be found in the object whose name is the one referenced by the *vn\_name* element in the Elfxx\_Verneed entry.

• The object with the reference does not use versioning, while the object with the definitions does. In this instance, only the definitions with index numbers 1 and 2 will be used in the reference match, the same identified by the static linker as the base definition. In cases where the static linker was not used, such as in calls to dlopen(), a version that does not have the base definition index shall be acceptable if it is the only version for which the symbol is defined.

• The object with the reference uses versioning, but the object with the definitions specifies none. A matching symbol shall be accepted in this case. A fatal error shall be triggered if a corruption in the required symbols list obscures an outdated object file and causes a match on the object filename in the Elfxx\_Verneed entry.

• Neither the object with the reference nor the object with the definitions use versioning. The behavior in this instance shall default to pre-existing symbol rules.

## **10.8 ABI note tag**

Every executable shall contain a section named .note.ABI-tag of type SHT\_NOTE. This section is structured as a note section as documented in the ELF spec. The section shall contain at least the following entry. The name field (namesz/name) contains the string "GNU". The type field shall be 1. The descsz field shall be at least 16, and the first 16 bytes of the desc field shall be as follows.

The first 32-bit word of the desc field shall be 0 (this signifies a Linux executable). The second, third, and fourth 32-bit words of the desc field contain the earliest compatible kernel version. For example, if the 3 words are 2, 2, and 5, this signifies a 2.2.5 kernel.

# **11 Dynamic Linking**

## **11.1 Program Loading and Dynamic Linking**

LSB-conforming implementations shall support the object file information and system actions that create running programs as specified in the [System V ABI](#ID_STD_46_GABI41) and [System V ABI Update](#ID_STD_46_ABIUPDATE) and as further required by this specification and the relevant architecture specific part of the LSB Core Specification.

Any shared object that is loaded shall contain sufficient DT\_NEEDED records to satisfy the symbols on the shared library.

## **11.2 Program Header**

In addition to the Segment Types defined in the [System V ABI](#ID_STD_46_GABI41) and [System V ABI Update](#ID_STD_46_ABIUPDATE) the following Segment Types shall also be supported.

**Table 11-1 Linux Segment Types**

| **Name** | **Value** |
| --- | --- |
| PT\_GNU\_EH\_FRAME | 0x6474e550 |
| PT\_GNU\_STACK | 0x6474e551 |
| PT\_GNU\_RELRO | 0x6474e552 |

PT\_GNU\_EH\_FRAME

  The array element specifies the location and size of the exception handling information as defined by the .eh\_frame\_hdr section.

PT\_GNU\_STACK

  The *p\_flags* member specifies the permissions on the segment containing the stack and is used to indicate wether the stack should be executable. The absense of this header indicates that the stack will be executable.

PT\_GNU\_RELRO

  the array element specifies the location and size of a segment which may be made read-only after relocations have been processed.

## **11.3 Dynamic Entries**

### **11.3.1 Introduction**

As described in [System V ABI](#ID_STD_46_GABI41), if an object file participates in dynamic linking, its program header table shall have an element of type PT\_DYNAMIC. This `segment' contains the .dynamic section. A special symbol, \_DYNAMIC, labels the section, which contains an array of the following structures.

|  |
| --- |
| typedef struct {  Elf32\_Sword d\_tag;  union {  Elf32\_Word d\_val;  Elf32\_Addr d\_ptr;  } d\_un;  } Elf32\_Dyn;  extern Elf32\_Dyn \_DYNAMIC[];  typedef struct {  Elf64\_Sxword d\_tag;  union {  Elf64\_Xword d\_val;  Elf64\_Addr d\_ptr;  } d\_un;  } Elf64\_Dyn;  extern Elf64\_Dyn \_DYNAMIC[]; |

Figure 11-1 Dynamic Structure

For each object with this type, *d\_tag* controls the interpretation of *d\_un*.

### **11.3.2 Dynamic Entries**

#### 11.3.2.1 ELF Dynamic Entries

The following dynamic entries are defined in the [System V ABI](#ID_STD_46_GABI41) and [System V ABI Update](#ID_STD_46_ABIUPDATE).

DT\_BIND\_NOW

  Process relocations of object

DT\_DEBUG

  For debugging; unspecified

DT\_FINI

  Address of termination function

DT\_FINI\_ARRAY

  The address of an array of pointers to termination functions.

DT\_FINI\_ARRAYSZ

  Size in bytes of DT\_FINI\_ARRAY

DT\_FLAGS

  Flag values specific to the object being loaded

DT\_HASH

  Address of symbol hash table

DT\_HIPROC

  End of processor-specific

DT\_INIT

  Address of init function

DT\_INIT\_ARRAY

  The address of an array of pointers to initialization functions.

DT\_INIT\_ARRAYSZ

  Size in bytes of DT\_INIT\_ARRAY

DT\_JMPREL

  Address of PLT relocs

DT\_LOPROC

  Start of processor-specific

DT\_NEEDED

  Name of needed library

DT\_NULL

  Marks end of dynamic section

DT\_PLTREL

  Type of reloc in PLT

DT\_PLTRELSZ

  Size in bytes of PLT relocs

DT\_PREINIT\_ARRAY

  Array with addresses of preinit functions

DT\_PREINIT\_ARRAYSZ

  Size in bytes of DT\_PREINIT\_ARRAY

DT\_REL

  Address of Rel relocs

DT\_RELA

  Address of Rela relocs

DT\_RELAENT

  Size of one Rela reloc

DT\_RELASZ

  Total size of Rela relocs

DT\_RELENT

  Size of one Rel reloc

DT\_RELSZ

  Total size of Rel relocs

DT\_RPATH

  Library search path

DT\_RUNPATH

  null-terminated library search path string

DT\_SONAME

  Name of shared object

DT\_STRSZ

  Size of string table

DT\_STRTAB

  Address of string table

DT\_SYMBOLIC

  Start symbol search here

DT\_SYMENT

  Size of one symbol table entry

DT\_SYMTAB

  Address of symbol table

DT\_TEXTREL

  Reloc might modify .text

#### 11.3.2.2 Additional Dynamic Entries

An LSB conforming object may also use the following additional Dynamic Entry types.

DT\_ADDRRNGHI

  Values from DT\_ADDRRNGLO through DT\_ADDRRNGHI are reserved for definition by an architecture specific part.

DT\_ADDRRNGLO

  Values from DT\_ADDRRNGLO through DT\_ADDRRNGHI are reserved for definition by an architecture specific part.

DT\_AUXILIARY

  Shared object to load before self

DT\_FILTER

  Shared object to get values from

DT\_HIOS

  Values from DT\_LOOS through DT\_HIOS are reserved for definition by specific operating systems.

DT\_LOOS

  Values from DT\_LOOS through DT\_HIOS are reserved for definition by specific operating systems.

DT\_NUM

  Number of dynamic entry tags defined (excepting reserved ranges).

DT\_POSFLAG\_1

  Flags for DT\_\* entries, effecting the following DT\_\* entry

DT\_RELCOUNT

  All Elf32\_Rel R\_\*\_RELATIVE relocations have been placed into a single block and this entry specifies the number of entries in that block. This permits ld.so.1 to streamline the processing of RELATIVE relocations.

DT\_SYMINENT

  Entry size of syminfo

DT\_SYMINFO

  Address of the Syminfo table.

DT\_SYMINSZ

  Size of syminfo table (in bytes)

DT\_VALRNGHI

  Entries which fall between DT\_VALRNGHI & DT\_VALRNGLO use the Dyn.d\_un.d\_val field of the Elf\*\_Dyn structure.

DT\_VALRNGLO

  Entries which fall between DT\_VALRNGHI & DT\_VALRNGLO use the Dyn.d\_un.d\_val field of the Elf\*\_Dyn structure.

DT\_VERDEF

  Address of version definition table

DT\_VERDEFNUM

  Number of version definitions

DT\_VERNEED

  Address of table with needed versions

DT\_VERNEEDNUM

  Number of needed versions

DT\_VERSYM

  Address of the table provided by the .gnu.version section.

# **12 C++ Class Representations**

## **12.1 C++ Data Representation**

Support for the C++ language shall be as specified in [Itanium™ C++ ABI](#ID_STD_46_CXXABI).

**Note:** This document, although containing a few architecture specific matters, is written as a generic specification, to be usable by C++ implementations on a variety of architectures.

This section provides additional information to supplement [Itanium™ C++ ABI](#ID_STD_46_CXXABI). Many of the definitions in that document are made in terms of C++. This section provides addition explanations using C terms to avoid self-referential problems.

### **12.1.1 Class Representation**

An object file generated by the compilation process for a C++ program shall contain several closely related internal objects, or Class Components, to represent each C++ Class. Such objects are not a visible part of the source code. [Table 12-1](#ID_TBL_45_CXX_45_CLASSREP) describes these Class Components at a high level.

**Table 12-1 Class Components**

| **Object** | **Contains** |
| --- | --- |
| Class Data | All non-static Class members |
| Virtual Table | Information needed to dispatch virtual functions, access virtual base class subobjects and to access the RTTI information |
| RTTI | Run-Time Type Information used by the typeid and dynamic\_cast operators, and exception handlers |
| Typeinfo Name | String representation of Class name |
| Construction Virtual Table | Information needed during construction and destruction of Classes with non-trivial inheritance relationships. |
| VTT | A table of virtual table pointers which holds the addresses of construction and non-construction virtual tables. |

#### 12.1.1.1 Virtual Table

Virtual tables are specified in Section 2.5.3 of [Itanium™ C++ ABI](#ID_STD_46_CXXABI).

Of the various categories of virtual table described in that specification, Category 1 (Leaf) is further described in [Figure 12-1](#ID_CXX_45_VIRTUALTABLE_45_CAT1) and Category 2 (Non-virtual bases only) is further described in [Figure 12-2](#ID_CXX_45_VIRTUALTABLE_45_CAT2). LSB conforming systems shall support these categories.

|  |
| --- |
| struct {  ptrdiff\_t baseobject;  const char \*typeinfo;  fptr virtfuncs[0];  }; |

Figure 12-1 Category 1 Virtual Table

|  |
| --- |
| struct {  unsigned long vcalloffset;  ptrdiff\_t baseobject;  const char \*typeinfo;  fptr virtfuncs[0];  }; |

Figure 12-2 Category 2 Virtual Table

This specification describes requirements for virtual tables of C++ classes using tables of the following form:

**Table 12-2 Primary vtable for K (example)**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for K |
| vfunc[0]: | K::~K() |
| vfunc[1]: | K::~K() |
| vfunc[2]: | K::m1(int\*) |
| vfunc[3]: | X::m2() |
| vfunc[4]: | \_\_cxa\_pure\_virtual() |
| vfunc[5]: | NULL or X::m4(int) |

Each row starting from 'vfunc[i]:' refers to a vtable entry 'vfunc[i]' of a class K, which is an entry for a virtual function A::m, where A is a base class of the class K as described in the [Itanium™ C++ ABI](#ID_STD_46_CXXABI). This specification requires implementations to interpret the vtable entry information in the following way:

1. A conforming implementation shall contain a vtable of the class K in the specified shared library;

2. The corresponding entry of this vtable 'vfunc[i]' shall be an entry for the virtual function A::m;

3. If the second column of the row contains \_\_cxa\_pure\_virtual() the corresponding vtable entry of a LSB-conforming implementation shall contain \_\_cxa\_pure\_virtual() or 'Y::m', where Y is the class K, the class A or a base class of the class K derived from the class A.

**Note:** In this case virtual function A::m in class K is considered to be specified as pure virtual by this specification.

4. If the second column of the row contains 'X::m' the corresponding vtable entry of a LSB-conforming implementation shall contain 'Y::m', where Y is the class K, the class X or a base class of the class K derived from the class X.

5. If the second column of the row contains 'NULL or X::m' the corresponding vtable entry of a LSB-conforming implementation shall contain NULL or 'Y::m', where Y is the class K, the class X or a base class of the class K derived from the class X.

**Note:** In this case virtual function A::m in class K is considered to be specified as inline by this specification.

An application may use any non-pure virtual function specified in this specification, and can expect the specified behavior irrespective of which particular method implements this functionality. An application may not use inline virtual functions at the binary level since its vtable entry may be NULL.

#### 12.1.1.2 Run-Time Type Information

Each type used in a C++ program has a data structure associated with it that provide information about the type which is used at runtime. This Run Time Type Information (RTTI) is defined in section 2.9.5 in [Itanium™ C++ ABI](#ID_STD_46_CXXABI). Additional details about the layout of this data is provided here.

|  |
| --- |
| struct {  void \*basevtable;  char \*name;  }; |

Figure 12-3 Run-Time Type Information Prefix

|  |
| --- |
| struct {  void \*basevtable;  char \*name;  void \*basetypeinfo[0];  }; |

Figure 12-4 Run-Time Type Information For classes with no base class

|  |
| --- |
| struct {  void \*basevtable;  char \*name;  void \*basetype;  void \*basetypeinfo[0];  }; |

Figure 12-5 Run-Time Type Information - classes with one base class

|  |
| --- |
| struct base\_type\_info {  char \*base\_type;  unsigned long offset\_flags;  };  struct {  void \*basevtable;  char \*name;  unsigned int flags;  unsigned int base\_count;  struct base\_type\_info base\_info[0];  }; |

Figure 12-6 Run-Time Type Information – classes with multiple inheritance

|  |
| --- |
| struct {  void \*basevtable;  char \*name;  unsigned int flags;  void \*pointee;  void \*basetypeinfo[0];  }; |

Figure 12-7 Run-Time Type Information for pointer types

|  |
| --- |
| struct {  void \*basevtable;  char \*name;  unsigned int flags;  void \*pointee;  void \*context;  void \*basetypeinfo[0];  }; |

Figure 12-8 Run-Time Type Information for pointer to member types

# **13 Symbol Mapping**

This chapter defines how names are mapped from the source symbol to the object symbol.

## **13.1 Symbol Mapping**

Symbols in a source program are translated by the compilation system into symbols that exist in the object file. The rules for this translation are defined here.

### **13.1.1 C++ Language**

External symbol names in a C++ object file shall be encoded according to the "name mangling" rules described in the [Itanium™ C++ ABI](#ID_STD_46_CXXABI).

# **III Base Libraries**

# **14 Base Libraries**

## **14.1 Introduction**

An LSB-conforming implementation shall support the following base libraries which provide interfaces for accessing the operating system, processor and other hardware in the system.

• libc

• libm

• libgcc\_s

• libdl

• librt

• libcrypt

• libpam

There are three main parts to the definition of each of these libraries.

The "Interfaces" section defines the required library name and version, and the required public symbols (interfaces and global data), as well as symbol versions, if any.

The "Interface Definitions" section provides complete or partial definitions of certain interfaces where either this specification is the source specification, or where there are variations from the source specification. If an interface definition requires one or more header files, one of those headers shall include the function prototype for the interface.

For source definitions of interfaces which include a reference to a header file, the contents of such header files form a part of the specification. The "Data Definitions" section provides the binary-level details for the header files from the source specifications, such as values for macros and enumerated types, as well as structure layouts, sizes and padding, etc. These data definitions, although presented in the form of header files for convenience, should not be taken a representing complete header files, as they are a supplement to the source specifications. Application developers should follow the guidelines of the source specifications when determining which header files need to be included to completely resolve all references.

**Note:** While the Data Definitions supplement the source specifications, this specification itself does not require conforming implementations to supply any header files.

## **14.2 Program Interpreter**

The Program Interpreter is specified in the appropriate architecture specific part of the LSB Core Specification.

## **14.3 Interfaces for libc**

[Table 14-1](#ID_LIB_45_LIBC_45_DEF) defines the library name and shared object name for the libc library

**Table 14-1 libc Definition**

|  |  |
| --- | --- |
| Library: | libc |
| SONAME: | See architecture specific part. |

The behavior of the interfaces in this library is specified by the following specifications:

|  |
| --- |
| [LFS] [Large File Support](#ID_STD_46_LFS) |
| [LSB] [This Specification](#ID_STD_46_LSB) |
| [RPC + XDR] [RFC 5531/4506 RPC & XDR](#ID_STD_46_RPC_46_XDR) |
| [SUSv2] [SUSv2](#ID_STD_46_SUSV2) |
| [SUSv3] [POSIX 1003.1-2001 (ISO/IEC 9945-2003)](#ID_STD_46_SUSV3) |
| [SUSv4] [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) |
| [SVID.4] [SVID Issue 4](#ID_STD_46_SVID_46_4) |

### **14.3.1 RPC**

#### 14.3.1.1 Interfaces for RPC

An LSB conforming implementation shall provide the generic functions for RPC specified in [Table 14-2](#ID_TBL_45_LIBC_45_RPC_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-2 libc - RPC Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| authnone\_create [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | callrpc [[RPC + XDR]](#ID_REFSTD_46_LIBC_46_3) | clnt\_create [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | clnt\_pcreateerror [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) |
| clnt\_perrno [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | clnt\_perror [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | clnt\_spcreateerror [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | clnt\_sperrno [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) |
| clnt\_sperror [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | clntraw\_create [[RPC + XDR]](#ID_REFSTD_46_LIBC_46_3) | clnttcp\_create [[RPC + XDR]](#ID_REFSTD_46_LIBC_46_3) | clntudp\_bufcreate [[RPC + XDR]](#ID_REFSTD_46_LIBC_46_3) |
| clntudp\_create [[RPC + XDR]](#ID_REFSTD_46_LIBC_46_3) | key\_decryptsession [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | pmap\_getport [[LSB]](#ID_REFSTD_46_LIBC_46_2) | pmap\_set [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| pmap\_unset [[LSB]](#ID_REFSTD_46_LIBC_46_2) | svc\_getreqset [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | svc\_register [[LSB]](#ID_REFSTD_46_LIBC_46_2) | svc\_run [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| svc\_sendreply [[LSB]](#ID_REFSTD_46_LIBC_46_2) | svcerr\_auth [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | svcerr\_decode [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | svcerr\_noproc [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) |
| svcerr\_noprog [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | svcerr\_progvers [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | svcerr\_systemerr [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | svcerr\_weakauth [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) |
| svcfd\_create [[RPC + XDR]](#ID_REFSTD_46_LIBC_46_3) | svcraw\_create [[RPC + XDR]](#ID_REFSTD_46_LIBC_46_3) | svctcp\_create [[LSB]](#ID_REFSTD_46_LIBC_46_2) | svcudp\_create [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| xdr\_accepted\_reply [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | xdr\_array [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | xdr\_bool [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | xdr\_bytes [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) |
| xdr\_callhdr [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | xdr\_callmsg [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | xdr\_char [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | xdr\_double [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) |
| xdr\_enum [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | xdr\_float [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | xdr\_free [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | xdr\_int [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) |
| xdr\_long [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | xdr\_opaque [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | xdr\_opaque\_auth [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | xdr\_pointer [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) |
| xdr\_reference [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | xdr\_rejected\_reply [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | xdr\_replymsg [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | xdr\_short [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) |
| xdr\_string [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | xdr\_u\_char [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | xdr\_u\_int [[LSB]](#ID_REFSTD_46_LIBC_46_2) | xdr\_u\_long [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) |
| xdr\_u\_short [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | xdr\_union [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | xdr\_vector [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | xdr\_void [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) |
| xdr\_wrapstring [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | xdrmem\_create [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | xdrrec\_create [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | xdrrec\_endofrecord [[RPC + XDR]](#ID_REFSTD_46_LIBC_46_3) |
| xdrrec\_eof [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) | xdrrec\_skiprecord [[RPC + XDR]](#ID_REFSTD_46_LIBC_46_3) | xdrstdio\_create [[LSB]](#ID_REFSTD_46_LIBC_46_2) |  |

An LSB conforming implementation shall provide the generic deprecated functions for RPC specified in [Table 14-3](#ID_TBL_45_LIBC_45_RPC_45_DEPINTS), with the full mandatory functionality as described in the referenced underlying specification.

**Note:** These interfaces are deprecated, and applications should avoid using them. These interfaces may be withdrawn in future releases of this specification.

**Table 14-3 libc - RPC Deprecated Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| key\_decryptsession [[SVID.4]](#ID_REFSTD_46_LIBC_46_7) |  |  |  |

### **14.3.2 Epoll**

#### 14.3.2.1 Interfaces for Epoll

An LSB conforming implementation shall provide the generic functions for Epoll specified in [Table 14-4](#ID_TBL_45_LIBC_45_EPOLL_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-4 libc - Epoll Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| epoll\_create(GLIBC\_2.3.2) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | epoll\_ctl(GLIBC\_2.3.2) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | epoll\_wait(GLIBC\_2.3.2) [[LSB]](#ID_REFSTD_46_LIBC_46_2) |  |

### **14.3.3 System Calls**

#### 14.3.3.1 Interfaces for System Calls

An LSB conforming implementation shall provide the generic functions for System Calls specified in [Table 14-5](#ID_TBL_45_LIBC_45_SYSTE_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-5 libc - System Calls Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| \_\_chk\_fail(GLIBC\_2.3.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_fxstat [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_fxstatat(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_getgroups\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| \_\_getpgid [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_lxstat [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_read\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_readlink\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| \_\_stack\_chk\_fail(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_xmknod [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_xmknodat(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_xstat [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| access [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | acct [[LSB]](#ID_REFSTD_46_LIBC_46_2) | alarm [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | backtrace [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| backtrace\_symbols [[LSB]](#ID_REFSTD_46_LIBC_46_2) | backtrace\_symbols\_fd [[LSB]](#ID_REFSTD_46_LIBC_46_2) | brk [[SUSv2]](#ID_REFSTD_46_LIBC_46_4) | chdir [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| chmod [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | chown [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | chroot [[SUSv2]](#ID_REFSTD_46_LIBC_46_4) | clock [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| close [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | closedir [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | creat [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | dup [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| dup2 [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | execl [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | execle [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | execlp [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| execv [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | execve [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | execvp [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | exit [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| faccessat(GLIBC\_2.4) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | fchdir [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | fchmod [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | fchmodat(GLIBC\_2.4) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| fchown [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | fchownat(GLIBC\_2.4) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | fcntl [[LSB]](#ID_REFSTD_46_LIBC_46_2) | fdatasync [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| fdopendir(GLIBC\_2.4) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | fexecve [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | flock [[LSB]](#ID_REFSTD_46_LIBC_46_2) | fork [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| fstatfs [[LSB]](#ID_REFSTD_46_LIBC_46_2) | fstatvfs [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | fsync [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | ftime [[SUSv3]](#ID_REFSTD_46_LIBC_46_5) |
| ftruncate [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | futimens(GLIBC\_2.6) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | futimes(GLIBC\_2.3) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | getcontext [[SUSv3]](#ID_REFSTD_46_LIBC_46_5) |
| getdtablesize [[LSB]](#ID_REFSTD_46_LIBC_46_2) | getegid [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | geteuid [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getgid [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| getgroups [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getitimer [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getloadavg [[LSB]](#ID_REFSTD_46_LIBC_46_2) | getpagesize [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| getpgid [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getpgrp [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getpid [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getppid [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| getpriority [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getrlimit [[LSB]](#ID_REFSTD_46_LIBC_46_2) | getrusage [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getsid [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| getuid [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getwd [[SUSv3]](#ID_REFSTD_46_LIBC_46_5) | initgroups [[LSB]](#ID_REFSTD_46_LIBC_46_2) | ioctl [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| kill [[LSB]](#ID_REFSTD_46_LIBC_46_2) | killpg [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | lchown [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | link [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| linkat(GLIBC\_2.4) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | lockf [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | lseek [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | lutimes(GLIBC\_2.3) [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| mkdir [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | mkdirat(GLIBC\_2.4) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | mkfifo [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | mkfifoat(GLIBC\_2.4) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| mlock [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | mlockall [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | mmap [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | mprotect [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| mremap [[LSB]](#ID_REFSTD_46_LIBC_46_2) | msync [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | munlock [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | munlockall [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| munmap [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | nanosleep [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | nice [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | open [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| openat(GLIBC\_2.4) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | opendir [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | pathconf [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | pause [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| pipe [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | poll [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | pread [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | pselect [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| ptrace [[LSB]](#ID_REFSTD_46_LIBC_46_2) | pwrite [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | read [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | readdir [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| readdir\_r [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | readlink [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | readlinkat(GLIBC\_2.4) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | readv [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| rename [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | renameat(GLIBC\_2.4) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | rmdir [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | sbrk [[SUSv2]](#ID_REFSTD_46_LIBC_46_4) |
| sched\_get\_priority\_max [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | sched\_get\_priority\_min [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | sched\_getaffinity(GLIBC\_2.3.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | sched\_getparam [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| sched\_getscheduler [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | sched\_rr\_get\_interval [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | sched\_setaffinity(GLIBC\_2.3.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | sched\_setparam [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| sched\_setscheduler [[LSB]](#ID_REFSTD_46_LIBC_46_2) | sched\_yield [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | select [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | setcontext [[SUSv3]](#ID_REFSTD_46_LIBC_46_5) |
| setegid [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | seteuid [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | setgid [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | setitimer [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| setpgid [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | setpgrp [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | setpriority [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | setregid [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| setreuid [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | setrlimit [[LSB]](#ID_REFSTD_46_LIBC_46_2) | setrlimit64 [[LFS]](#ID_REFSTD_46_LIBC_46_1) | setsid [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| setuid [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | sleep [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | statfs [[LSB]](#ID_REFSTD_46_LIBC_46_2) | statvfs [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| stime [[LSB]](#ID_REFSTD_46_LIBC_46_2) | symlink [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | symlinkat(GLIBC\_2.4) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | sync [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| sysconf [[LSB]](#ID_REFSTD_46_LIBC_46_2) | sysinfo [[LSB]](#ID_REFSTD_46_LIBC_46_2) | time [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | times [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| truncate [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | ulimit [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | umask [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | uname [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| unlink [[LSB]](#ID_REFSTD_46_LIBC_46_2) | unlinkat(GLIBC\_2.4) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | utime [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | utimensat(GLIBC\_2.6) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| utimes [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | vfork [[SUSv3]](#ID_REFSTD_46_LIBC_46_5) | wait [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wait4 [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| waitid [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | waitpid [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | write [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | writev [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |

An LSB conforming implementation shall provide the generic deprecated functions for System Calls specified in [Table 14-6](#ID_TBL_45_LIBC_45_SYSTE_45_DEPINTS), with the full mandatory functionality as described in the referenced underlying specification.

**Note:** These interfaces are deprecated, and applications should avoid using them. These interfaces may be withdrawn in future releases of this specification.

**Table 14-6 libc - System Calls Deprecated Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| fstatfs [[LSB]](#ID_REFSTD_46_LIBC_46_2) | getdtablesize [[LSB]](#ID_REFSTD_46_LIBC_46_2) | getpagesize [[LSB]](#ID_REFSTD_46_LIBC_46_2) | getwd [[SUSv3]](#ID_REFSTD_46_LIBC_46_5) |
| statfs [[LSB]](#ID_REFSTD_46_LIBC_46_2) |  |  |  |

### **14.3.4 Standard I/O**

#### 14.3.4.1 Interfaces for Standard I/O

An LSB conforming implementation shall provide the generic functions for Standard I/O specified in [Table 14-7](#ID_TBL_45_LIBC_45_STAND_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-7 libc - Standard I/O Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| \_IO\_feof [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_IO\_getc [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_IO\_putc [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_IO\_puts [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| \_\_fgets\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_fgets\_unlocked\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_fgetws\_unlocked\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_fprintf\_chk [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| \_\_printf\_chk [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_snprintf\_chk [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_sprintf\_chk [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_vfprintf\_chk [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| \_\_vprintf\_chk [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_vsnprintf\_chk [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_vsprintf\_chk [[LSB]](#ID_REFSTD_46_LIBC_46_2) | asprintf [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| clearerr [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | clearerr\_unlocked [[LSB]](#ID_REFSTD_46_LIBC_46_2) | ctermid [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | dprintf [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| fclose [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | fdopen [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | feof [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | feof\_unlocked [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| ferror [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | ferror\_unlocked [[LSB]](#ID_REFSTD_46_LIBC_46_2) | fflush [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | fflush\_unlocked [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| fgetc [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | fgetc\_unlocked [[LSB]](#ID_REFSTD_46_LIBC_46_2) | fgetpos [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | fgets [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| fgets\_unlocked [[LSB]](#ID_REFSTD_46_LIBC_46_2) | fgetwc\_unlocked [[LSB]](#ID_REFSTD_46_LIBC_46_2) | fgetws\_unlocked [[LSB]](#ID_REFSTD_46_LIBC_46_2) | fileno [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| fileno\_unlocked [[LSB]](#ID_REFSTD_46_LIBC_46_2) | flockfile [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | fopen [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | fprintf [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| fputc [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | fputc\_unlocked [[LSB]](#ID_REFSTD_46_LIBC_46_2) | fputs [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | fputs\_unlocked [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| fputwc\_unlocked [[LSB]](#ID_REFSTD_46_LIBC_46_2) | fputws\_unlocked [[LSB]](#ID_REFSTD_46_LIBC_46_2) | fread [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | fread\_unlocked [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| freopen [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | fscanf [[LSB]](#ID_REFSTD_46_LIBC_46_2) | fseek [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | fseeko [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| fsetpos [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | ftell [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | ftello [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | fwrite [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| fwrite\_unlocked [[LSB]](#ID_REFSTD_46_LIBC_46_2) | getc [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getc\_unlocked [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getchar [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| getchar\_unlocked [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getdelim [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getline [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getw [[SUSv2]](#ID_REFSTD_46_LIBC_46_4) |
| getwc\_unlocked [[LSB]](#ID_REFSTD_46_LIBC_46_2) | getwchar\_unlocked [[LSB]](#ID_REFSTD_46_LIBC_46_2) | pclose [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | popen [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| printf [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | putc [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | putc\_unlocked [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | putchar [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| putchar\_unlocked [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | puts [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | putw [[SUSv2]](#ID_REFSTD_46_LIBC_46_4) | putwc\_unlocked [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| putwchar\_unlocked [[LSB]](#ID_REFSTD_46_LIBC_46_2) | remove [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | rewind [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | rewinddir [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| scanf [[LSB]](#ID_REFSTD_46_LIBC_46_2) | seekdir [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | setbuf [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | setbuffer [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| setvbuf [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | snprintf [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | sprintf [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | sscanf [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| telldir [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | tempnam [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | ungetc [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | vasprintf [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| vdprintf [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | vfprintf [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | vprintf [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | vsnprintf [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| vsprintf [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |  |  |  |

An LSB conforming implementation shall provide the generic deprecated functions for Standard I/O specified in [Table 14-8](#ID_TBL_45_LIBC_45_STAND_45_DEPINTS), with the full mandatory functionality as described in the referenced underlying specification.

**Note:** These interfaces are deprecated, and applications should avoid using them. These interfaces may be withdrawn in future releases of this specification.

**Table 14-8 libc - Standard I/O Deprecated Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| tempnam [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |  |  |  |

An LSB conforming implementation shall provide the generic data interfaces for Standard I/O specified in [Table 14-9](#ID_TBL_45_LIBC_45_STAND_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-9 libc - Standard I/O Data Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| stderr [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | stdin [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | stdout [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |  |

### **14.3.5 Signal Handling**

#### 14.3.5.1 Interfaces for Signal Handling

An LSB conforming implementation shall provide the generic functions for Signal Handling specified in [Table 14-10](#ID_TBL_45_LIBC_45_SIGNA_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-10 libc - Signal Handling Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| \_\_libc\_current\_sigrtmax [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_libc\_current\_sigrtmin [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_sigsetjmp [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_sysv\_signal [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| \_\_xpg\_sigpause [[LSB]](#ID_REFSTD_46_LIBC_46_2) | bsd\_signal [[SUSv3]](#ID_REFSTD_46_LIBC_46_5) | psiginfo(GLIBC\_2.10) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | psignal [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| raise [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | sigaction [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | sigaddset [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | sigaltstack [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| sigandset [[LSB]](#ID_REFSTD_46_LIBC_46_2) | sigdelset [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | sigemptyset [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | sigfillset [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| sighold [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | sigignore [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | siginterrupt [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | sigisemptyset [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| sigismember [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | siglongjmp [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | signal [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | sigorset [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| sigpause [[LSB]](#ID_REFSTD_46_LIBC_46_2) | sigpending [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | sigprocmask [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | sigqueue [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| sigrelse [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | sigreturn [[LSB]](#ID_REFSTD_46_LIBC_46_2) | sigset [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | sigsuspend [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| sigtimedwait [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | sigwait [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | sigwaitinfo [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |  |

An LSB conforming implementation shall provide the generic deprecated functions for Signal Handling specified in [Table 14-11](#ID_TBL_45_LIBC_45_SIGNA_45_DEPINTS), with the full mandatory functionality as described in the referenced underlying specification.

**Note:** These interfaces are deprecated, and applications should avoid using them. These interfaces may be withdrawn in future releases of this specification.

**Table 14-11 libc - Signal Handling Deprecated Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| sigpause [[LSB]](#ID_REFSTD_46_LIBC_46_2) |  |  |  |

An LSB conforming implementation shall provide the generic data interfaces for Signal Handling specified in [Table 14-12](#ID_TBL_45_LIBC_45_SIGNA_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-12 libc - Signal Handling Data Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| \_sys\_siglist [[LSB]](#ID_REFSTD_46_LIBC_46_2) |  |  |  |

### **14.3.6 Localization Functions**

#### 14.3.6.1 Interfaces for Localization Functions

An LSB conforming implementation shall provide the generic functions for Localization Functions specified in [Table 14-13](#ID_TBL_45_LIBC_45_LOCAL_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-13 libc - Localization Functions Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| bind\_textdomain\_codeset [[LSB]](#ID_REFSTD_46_LIBC_46_2) | bindtextdomain [[LSB]](#ID_REFSTD_46_LIBC_46_2) | catclose [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | catgets [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| catopen [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | dcgettext [[LSB]](#ID_REFSTD_46_LIBC_46_2) | dcngettext [[LSB]](#ID_REFSTD_46_LIBC_46_2) | dgettext [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| dngettext [[LSB]](#ID_REFSTD_46_LIBC_46_2) | duplocale(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | freelocale(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | gettext [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| iconv [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | iconv\_close [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | iconv\_open [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | localeconv [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| newlocale(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | ngettext [[LSB]](#ID_REFSTD_46_LIBC_46_2) | nl\_langinfo [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | setlocale [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| textdomain [[LSB]](#ID_REFSTD_46_LIBC_46_2) | uselocale(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |  |  |

An LSB conforming implementation shall provide the generic data interfaces for Localization Functions specified in [Table 14-14](#ID_TBL_45_LIBC_45_LOCAL_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-14 libc - Localization Functions Data Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| \_nl\_msg\_cat\_cntr [[LSB]](#ID_REFSTD_46_LIBC_46_2) |  |  |  |

### **14.3.7 Posix Spawn Option**

#### 14.3.7.1 Interfaces for Posix Spawn Option

An LSB conforming implementation shall provide the generic functions for Posix Spawn Option specified in [Table 14-15](#ID_TBL_45_LIBC_45_POSIX_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-15 libc - Posix Spawn Option Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| posix\_spawn [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | posix\_spawn\_file\_actions\_addclose [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | posix\_spawn\_file\_actions\_adddup2 [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | posix\_spawn\_file\_actions\_addopen [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| posix\_spawn\_file\_actions\_destroy [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | posix\_spawn\_file\_actions\_init [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | posix\_spawnattr\_destroy [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | posix\_spawnattr\_getflags [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| posix\_spawnattr\_getpgroup [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | posix\_spawnattr\_getschedparam [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | posix\_spawnattr\_getschedpolicy [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | posix\_spawnattr\_getsigdefault [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| posix\_spawnattr\_getsigmask [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | posix\_spawnattr\_init [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | posix\_spawnattr\_setflags [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | posix\_spawnattr\_setpgroup [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| posix\_spawnattr\_setschedparam [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | posix\_spawnattr\_setschedpolicy [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | posix\_spawnattr\_setsigdefault [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | posix\_spawnattr\_setsigmask [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| posix\_spawnp [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |  |  |  |

### **14.3.8 Posix Advisory Option**

#### 14.3.8.1 Interfaces for Posix Advisory Option

An LSB conforming implementation shall provide the generic functions for Posix Advisory Option specified in [Table 14-16](#ID_TBL_45_LIBC_45_POSIY_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-16 libc - Posix Advisory Option Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| posix\_fadvise [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | posix\_fallocate [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | posix\_madvise [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | posix\_memalign [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |

### **14.3.9 Socket Interface**

#### 14.3.9.1 Interfaces for Socket Interface

An LSB conforming implementation shall provide the generic functions for Socket Interface specified in [Table 14-17](#ID_TBL_45_LIBC_45_SOCKE_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-17 libc - Socket Interface Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| \_\_gethostname\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_h\_errno\_location [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_recv\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_recvfrom\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| accept [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | bind [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | bindresvport [[LSB]](#ID_REFSTD_46_LIBC_46_2) | connect [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| freeifaddrs(GLIBC\_2.3) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | gethostid [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | gethostname [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getifaddrs(GLIBC\_2.3) [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| getpeername [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getsockname [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getsockopt [[LSB]](#ID_REFSTD_46_LIBC_46_2) | if\_freenameindex [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| if\_indextoname [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | if\_nameindex [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | if\_nametoindex [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | listen [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| recv [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | recvfrom [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | recvmsg [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | send [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| sendmsg [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | sendto [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | setsockopt [[LSB]](#ID_REFSTD_46_LIBC_46_2) | shutdown [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| sockatmark [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | socket [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | socketpair [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |  |

An LSB conforming implementation shall provide the generic data interfaces for Socket Interface specified in [Table 14-18](#ID_TBL_45_LIBC_45_SOCKE_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-18 libc - Socket Interface Data Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| in6addr\_any [[SUSv3]](#ID_REFSTD_46_LIBC_46_5) | in6addr\_loopback [[SUSv3]](#ID_REFSTD_46_LIBC_46_5) |  |  |

### **14.3.10 Wide Characters**

#### 14.3.10.1 Interfaces for Wide Characters

An LSB conforming implementation shall provide the generic functions for Wide Characters specified in [Table 14-19](#ID_TBL_45_LIBC_45_WIDE_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-19 libc - Wide Characters Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| \_\_fgetws\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_fwprintf\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_mbsnrtowcs\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_mbsrtowcs\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| \_\_mbstowcs\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_swprintf\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_vfwprintf\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_vswprintf\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| \_\_vwprintf\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_wcpcpy\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_wcpncpy\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_wcrtomb\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| \_\_wcscat\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_wcscpy\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_wcsncat\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_wcsncpy\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| \_\_wcsnrtombs\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_wcsrtombs\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_wcstod\_internal [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_wcstof\_internal [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| \_\_wcstol\_internal [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_wcstold\_internal [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_wcstombs\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_wcstoul\_internal [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| \_\_wctomb\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_wmemcpy\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_wmemmove\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_wmempcpy\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| \_\_wmemset\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_wprintf\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | btowc [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | fgetwc [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| fgetws [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | fputwc [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | fputws [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | fwide [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| fwprintf [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | fwscanf [[LSB]](#ID_REFSTD_46_LIBC_46_2) | getwc [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getwchar [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| iswalnum\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | iswalpha\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | iswblank\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | iswcntrl\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| iswctype\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | iswdigit\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | iswgraph\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | iswlower\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| iswprint\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | iswpunct\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | iswspace\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | iswupper\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| iswxdigit\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | mblen [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | mbrlen [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | mbrtowc [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| mbsinit [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | mbsnrtowcs [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | mbsrtowcs [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | mbstowcs [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| mbtowc [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | putwc [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | putwchar [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | swprintf [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| swscanf [[LSB]](#ID_REFSTD_46_LIBC_46_2) | towctrans [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | towctrans\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | towlower [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| towlower\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | towupper [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | towupper\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | ungetwc [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| vfwprintf [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | vfwscanf [[LSB]](#ID_REFSTD_46_LIBC_46_2) | vswprintf [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | vswscanf [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| vwprintf [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | vwscanf [[LSB]](#ID_REFSTD_46_LIBC_46_2) | wcpcpy [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wcpncpy [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| wcrtomb [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wcscasecmp [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wcscasecmp\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wcscat [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| wcschr [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wcscmp [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wcscoll [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wcscoll\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| wcscpy [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wcscspn [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wcsdup [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wcsftime [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| wcslen [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wcsncasecmp [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wcsncasecmp\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wcsncat [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| wcsncmp [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wcsncpy [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wcsnlen [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wcsnrtombs [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| wcspbrk [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wcsrchr [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wcsrtombs [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wcsspn [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| wcsstr [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wcstod [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wcstof [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wcstoimax [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| wcstok [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wcstol [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wcstold [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wcstoll [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| wcstombs [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wcstoq [[LSB]](#ID_REFSTD_46_LIBC_46_2) | wcstoul [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wcstoull [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| wcstoumax [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wcstouq [[LSB]](#ID_REFSTD_46_LIBC_46_2) | wcswcs [[SUSv3]](#ID_REFSTD_46_LIBC_46_5) | wcswidth [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| wcsxfrm [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wcsxfrm\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wctob [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wctomb [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| wctrans [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wctrans\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wctype [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wctype\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| wcwidth [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wmemchr [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wmemcmp [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wmemcpy [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| wmemmove [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wmemset [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wprintf [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wscanf [[LSB]](#ID_REFSTD_46_LIBC_46_2) |

### **14.3.11 String Functions**

#### 14.3.11.1 Interfaces for String Functions

An LSB conforming implementation shall provide the generic functions for String Functions specified in [Table 14-20](#ID_TBL_45_LIBC_45_STRIN_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-20 libc - String Functions Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| \_\_memcpy\_chk(GLIBC\_2.3.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_memmove\_chk(GLIBC\_2.3.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_mempcpy [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_mempcpy\_chk(GLIBC\_2.3.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| \_\_memset\_chk(GLIBC\_2.3.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_rawmemchr [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_stpcpy [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_stpcpy\_chk(GLIBC\_2.3.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| \_\_stpncpy\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_strcat\_chk(GLIBC\_2.3.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_strcpy\_chk(GLIBC\_2.3.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_strdup [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| \_\_strncat\_chk(GLIBC\_2.3.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_strncpy\_chk(GLIBC\_2.3.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_strtod\_internal [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_strtof\_internal [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| \_\_strtok\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_strtol\_internal [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_strtold\_internal [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_strtoll\_internal [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| \_\_strtoul\_internal [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_strtoull\_internal [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_xpg\_strerror\_r(GLIBC\_2.3.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | bcmp [[SUSv3]](#ID_REFSTD_46_LIBC_46_5) |
| bcopy [[SUSv3]](#ID_REFSTD_46_LIBC_46_5) | bzero [[SUSv3]](#ID_REFSTD_46_LIBC_46_5) | ffs [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | index [[SUSv3]](#ID_REFSTD_46_LIBC_46_5) |
| memccpy [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | memchr [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | memcmp [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | memcpy [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| memmove [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | memrchr [[LSB]](#ID_REFSTD_46_LIBC_46_2) | memset [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | rindex [[SUSv3]](#ID_REFSTD_46_LIBC_46_5) |
| stpcpy [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | stpncpy [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strcasecmp [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strcasecmp\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| strcasestr [[LSB]](#ID_REFSTD_46_LIBC_46_2) | strcat [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strchr [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strcmp [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| strcoll [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strcoll\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strcpy [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strcspn [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| strdup [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strerror [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strerror\_l(GLIBC\_2.6) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strerror\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| strfmon [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strfmon\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strftime [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strftime\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| strlen [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strncasecmp [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strncasecmp\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strncat [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| strncmp [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strncpy [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strndup [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strnlen [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| strpbrk [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strptime [[LSB]](#ID_REFSTD_46_LIBC_46_2) | strrchr [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strsep [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| strsignal [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strspn [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strstr [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strtof [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| strtoimax [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strtok [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strtok\_r [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strtold [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| strtoll [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strtoq [[LSB]](#ID_REFSTD_46_LIBC_46_2) | strtoull [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strtoumax [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| strtouq [[LSB]](#ID_REFSTD_46_LIBC_46_2) | strxfrm [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strxfrm\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | swab [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |

An LSB conforming implementation shall provide the generic deprecated functions for String Functions specified in [Table 14-21](#ID_TBL_45_LIBC_45_STRIN_45_DEPINTS), with the full mandatory functionality as described in the referenced underlying specification.

**Note:** These interfaces are deprecated, and applications should avoid using them. These interfaces may be withdrawn in future releases of this specification.

**Table 14-21 libc - String Functions Deprecated Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| strerror\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) |  |  |  |

### **14.3.12 IPC Functions**

#### 14.3.12.1 Interfaces for IPC Functions

An LSB conforming implementation shall provide the generic functions for IPC Functions specified in [Table 14-22](#ID_TBL_45_LIBC_45_IPC_45_F_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-22 libc - IPC Functions Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| ftok [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | msgctl [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | msgget [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | msgrcv [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| msgsnd [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | semctl [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | semget [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | semop [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| shmat [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | shmctl [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | shmdt [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | shmget [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |

### **14.3.13 Regular Expressions**

#### 14.3.13.1 Interfaces for Regular Expressions

An LSB conforming implementation shall provide the generic functions for Regular Expressions specified in [Table 14-23](#ID_TBL_45_LIBC_45_REGUL_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-23 libc - Regular Expressions Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| regcomp [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | regerror [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | regexec [[LSB]](#ID_REFSTD_46_LIBC_46_2) | regfree [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |

### **14.3.14 Character Type Functions**

#### 14.3.14.1 Interfaces for Character Type Functions

An LSB conforming implementation shall provide the generic functions for Character Type Functions specified in [Table 14-24](#ID_TBL_45_LIBC_45_CHARA_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-24 libc - Character Type Functions Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| \_\_ctype\_b\_loc(GLIBC\_2.3) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_ctype\_get\_mb\_cur\_max [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_ctype\_tolower\_loc(GLIBC\_2.3) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_ctype\_toupper\_loc(GLIBC\_2.3) [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| \_tolower [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | \_toupper [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | isalnum [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | isalnum\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| isalpha [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | isalpha\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | isascii [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | isblank\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| iscntrl [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | iscntrl\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | isdigit [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | isdigit\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| isgraph [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | isgraph\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | islower [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | islower\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| isprint [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | isprint\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | ispunct [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | ispunct\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| isspace [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | isspace\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | isupper [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | isupper\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| iswalnum [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | iswalpha [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | iswblank [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | iswcntrl [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| iswctype [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | iswdigit [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | iswgraph [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | iswlower [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| iswprint [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | iswpunct [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | iswspace [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | iswupper [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| iswxdigit [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | isxdigit [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | isxdigit\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | toascii [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| tolower [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | tolower\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | toupper [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | toupper\_l(GLIBC\_2.3) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |

### **14.3.15 Time Manipulation**

#### 14.3.15.1 Interfaces for Time Manipulation

An LSB conforming implementation shall provide the generic functions for Time Manipulation specified in [Table 14-25](#ID_TBL_45_LIBC_45_TIME_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-25 libc - Time Manipulation Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| adjtime [[LSB]](#ID_REFSTD_46_LIBC_46_2) | asctime [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | asctime\_r [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | ctime [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| ctime\_r [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | difftime [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | gmtime [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | gmtime\_r [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| localtime [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | localtime\_r [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | mktime [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | tzset [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| ualarm [[SUSv3]](#ID_REFSTD_46_LIBC_46_5) |  |  |  |

An LSB conforming implementation shall provide the generic data interfaces for Time Manipulation specified in [Table 14-26](#ID_TBL_45_LIBC_45_TIME_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-26 libc - Time Manipulation Data Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| \_\_daylight [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_timezone [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_tzname [[LSB]](#ID_REFSTD_46_LIBC_46_2) | daylight [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| timezone [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | tzname [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |  |  |

### **14.3.16 Terminal Interface Functions**

#### 14.3.16.1 Interfaces for Terminal Interface Functions

An LSB conforming implementation shall provide the generic functions for Terminal Interface Functions specified in [Table 14-27](#ID_TBL_45_LIBC_45_TERMI_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-27 libc - Terminal Interface Functions Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| cfgetispeed [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | cfgetospeed [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | cfmakeraw [[LSB]](#ID_REFSTD_46_LIBC_46_2) | cfsetispeed [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| cfsetospeed [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | cfsetspeed [[LSB]](#ID_REFSTD_46_LIBC_46_2) | tcdrain [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | tcflow [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| tcflush [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | tcgetattr [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | tcgetpgrp [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | tcgetsid [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| tcsendbreak [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | tcsetattr [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | tcsetpgrp [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |  |

### **14.3.17 System Database Interface**

#### 14.3.17.1 Interfaces for System Database Interface

An LSB conforming implementation shall provide the generic functions for System Database Interface specified in [Table 14-28](#ID_TBL_45_LIBC_45_SYSTF_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-28 libc - System Database Interface Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| endgrent [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | endprotoent [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | endpwent [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | endservent [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| endutent [[LSB]](#ID_REFSTD_46_LIBC_46_2) | endutxent [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getgrent [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getgrent\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| getgrgid [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getgrgid\_r [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getgrnam [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getgrnam\_r [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| getgrouplist [[LSB]](#ID_REFSTD_46_LIBC_46_2) | gethostbyaddr [[SUSv3]](#ID_REFSTD_46_LIBC_46_5) | gethostbyaddr\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) | gethostbyname [[SUSv3]](#ID_REFSTD_46_LIBC_46_5) |
| gethostbyname2 [[LSB]](#ID_REFSTD_46_LIBC_46_2) | gethostbyname2\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) | gethostbyname\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) | getprotobyname [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| getprotobyname\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) | getprotobynumber [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getprotobynumber\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) | getprotoent [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| getprotoent\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) | getpwent [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getpwent\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) | getpwnam [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| getpwnam\_r [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getpwuid [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getpwuid\_r [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getservbyname [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| getservbyname\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) | getservbyport [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getservbyport\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) | getservent [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| getservent\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) | getutent [[LSB]](#ID_REFSTD_46_LIBC_46_2) | getutent\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) | getutxent [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| getutxid [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getutxline [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | pututxline [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | setgrent [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| setgroups [[LSB]](#ID_REFSTD_46_LIBC_46_2) | setprotoent [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | setpwent [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | setservent [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| setutent [[LSB]](#ID_REFSTD_46_LIBC_46_2) | setutxent [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | utmpname [[LSB]](#ID_REFSTD_46_LIBC_46_2) |  |

An LSB conforming implementation shall provide the generic deprecated functions for System Database Interface specified in [Table 14-29](#ID_TBL_45_LIBC_45_SYSTF_45_DEPINTS), with the full mandatory functionality as described in the referenced underlying specification.

**Note:** These interfaces are deprecated, and applications should avoid using them. These interfaces may be withdrawn in future releases of this specification.

**Table 14-29 libc - System Database Interface Deprecated Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| gethostbyaddr [[SUSv3]](#ID_REFSTD_46_LIBC_46_5) | gethostbyaddr\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) | gethostbyname [[SUSv3]](#ID_REFSTD_46_LIBC_46_5) | gethostbyname2 [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| gethostbyname2\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) | gethostbyname\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) |  |  |

### **14.3.18 Language Support**

#### 14.3.18.1 Interfaces for Language Support

An LSB conforming implementation shall provide the generic functions for Language Support specified in [Table 14-30](#ID_TBL_45_LIBC_45_LANGU_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-30 libc - Language Support Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| \_\_libc\_start\_main [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_register\_atfork(GLIBC\_2.3.2) [[LSB]](#ID_REFSTD_46_LIBC_46_2) |  |  |

### **14.3.19 Large File Support**

#### 14.3.19.1 Interfaces for Large File Support

An LSB conforming implementation shall provide the generic functions for Large File Support specified in [Table 14-31](#ID_TBL_45_LIBC_45_LARGE_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-31 libc - Large File Support Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| \_\_fxstat64 [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_fxstatat64(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_lxstat64 [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_xstat64 [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| creat64 [[LFS]](#ID_REFSTD_46_LIBC_46_1) | fgetpos64 [[LFS]](#ID_REFSTD_46_LIBC_46_1) | fopen64 [[LFS]](#ID_REFSTD_46_LIBC_46_1) | freopen64 [[LFS]](#ID_REFSTD_46_LIBC_46_1) |
| fseeko64 [[LFS]](#ID_REFSTD_46_LIBC_46_1) | fsetpos64 [[LFS]](#ID_REFSTD_46_LIBC_46_1) | fstatfs64 [[LSB]](#ID_REFSTD_46_LIBC_46_2) | fstatvfs64 [[LFS]](#ID_REFSTD_46_LIBC_46_1) |
| ftello64 [[LFS]](#ID_REFSTD_46_LIBC_46_1) | ftruncate64 [[LFS]](#ID_REFSTD_46_LIBC_46_1) | ftw64 [[LFS]](#ID_REFSTD_46_LIBC_46_1) | getrlimit64 [[LFS]](#ID_REFSTD_46_LIBC_46_1) |
| lockf64 [[LFS]](#ID_REFSTD_46_LIBC_46_1) | lseek64 [[LFS]](#ID_REFSTD_46_LIBC_46_1) | mkstemp64 [[LSB]](#ID_REFSTD_46_LIBC_46_2) | mmap64 [[LFS]](#ID_REFSTD_46_LIBC_46_1) |
| nftw64 [[LFS]](#ID_REFSTD_46_LIBC_46_1) | open64 [[LFS]](#ID_REFSTD_46_LIBC_46_1) | openat64(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | posix\_fadvise64 [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| posix\_fallocate64 [[LSB]](#ID_REFSTD_46_LIBC_46_2) | pread64 [[LSB]](#ID_REFSTD_46_LIBC_46_2) | pwrite64 [[LSB]](#ID_REFSTD_46_LIBC_46_2) | readdir64 [[LFS]](#ID_REFSTD_46_LIBC_46_1) |
| readdir64\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) | statfs64 [[LSB]](#ID_REFSTD_46_LIBC_46_2) | statvfs64 [[LFS]](#ID_REFSTD_46_LIBC_46_1) | tmpfile64 [[LFS]](#ID_REFSTD_46_LIBC_46_1) |
| truncate64 [[LFS]](#ID_REFSTD_46_LIBC_46_1) |  |  |  |

An LSB conforming implementation shall provide the generic deprecated functions for Large File Support specified in [Table 14-32](#ID_TBL_45_LIBC_45_LARGE_45_DEPINTS), with the full mandatory functionality as described in the referenced underlying specification.

**Note:** These interfaces are deprecated, and applications should avoid using them. These interfaces may be withdrawn in future releases of this specification.

**Table 14-32 libc - Large File Support Deprecated Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| fstatfs64 [[LSB]](#ID_REFSTD_46_LIBC_46_2) | statfs64 [[LSB]](#ID_REFSTD_46_LIBC_46_2) |  |  |

### **14.3.20 Inotify**

#### 14.3.20.1 Interfaces for Inotify

An LSB conforming implementation shall provide the generic functions for Inotify specified in [Table 14-33](#ID_TBL_45_LIBC_45_INOTI_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-33 libc - Inotify Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| inotify\_add\_watch(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | inotify\_init(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | inotify\_rm\_watch(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) |  |

### **14.3.21 Standard Library**

#### 14.3.21.1 Interfaces for Standard Library

An LSB conforming implementation shall provide the generic functions for Standard Library specified in [Table 14-34](#ID_TBL_45_LIBC_45_STANE_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-34 libc - Standard Library Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| \_Exit [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | \_\_assert\_fail [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_confstr\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_cxa\_atexit [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| \_\_cxa\_finalize [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_errno\_location [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_fpending [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_getcwd\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| \_\_getlogin\_r\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_getpagesize [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_isinf [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_isinff [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| \_\_isinfl [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_isnan [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_isnanf [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_isnanl [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| \_\_pread64\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_pread\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_realpath\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_sysconf [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| \_\_syslog\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_ttyname\_r\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_vsyslog\_chk(GLIBC\_2.4) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_\_xpg\_basename [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| \_exit [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | \_longjmp [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | \_setjmp [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | a64l [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| abort [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | abs [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | alphasort [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | alphasort64 [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| argz\_add [[LSB]](#ID_REFSTD_46_LIBC_46_2) | argz\_add\_sep [[LSB]](#ID_REFSTD_46_LIBC_46_2) | argz\_append [[LSB]](#ID_REFSTD_46_LIBC_46_2) | argz\_count [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| argz\_create [[LSB]](#ID_REFSTD_46_LIBC_46_2) | argz\_create\_sep [[LSB]](#ID_REFSTD_46_LIBC_46_2) | argz\_delete [[LSB]](#ID_REFSTD_46_LIBC_46_2) | argz\_extract [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| argz\_insert [[LSB]](#ID_REFSTD_46_LIBC_46_2) | argz\_next [[LSB]](#ID_REFSTD_46_LIBC_46_2) | argz\_replace [[LSB]](#ID_REFSTD_46_LIBC_46_2) | argz\_stringify [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| atof [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | atoi [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | atol [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | atoll [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| basename [[LSB]](#ID_REFSTD_46_LIBC_46_2) | bsearch [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | calloc [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | closelog [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| confstr [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | cuserid [[SUSv2]](#ID_REFSTD_46_LIBC_46_4) | daemon [[LSB]](#ID_REFSTD_46_LIBC_46_2) | dirfd [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| dirname [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | div [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | dl\_iterate\_phdr [[LSB]](#ID_REFSTD_46_LIBC_46_2) | drand48 [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| drand48\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) | ecvt [[SUSv3]](#ID_REFSTD_46_LIBC_46_5) | envz\_add [[LSB]](#ID_REFSTD_46_LIBC_46_2) | envz\_entry [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| envz\_get [[LSB]](#ID_REFSTD_46_LIBC_46_2) | envz\_merge [[LSB]](#ID_REFSTD_46_LIBC_46_2) | envz\_remove [[LSB]](#ID_REFSTD_46_LIBC_46_2) | envz\_strip [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| erand48 [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | erand48\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) | err [[LSB]](#ID_REFSTD_46_LIBC_46_2) | error [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| errx [[LSB]](#ID_REFSTD_46_LIBC_46_2) | fcvt [[SUSv3]](#ID_REFSTD_46_LIBC_46_5) | fmemopen [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | fmtmsg [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| fnmatch [[LSB]](#ID_REFSTD_46_LIBC_46_2) | fpathconf [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | free [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | freeaddrinfo [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| ftrylockfile [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | ftw [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | funlockfile [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | gai\_strerror [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| gcvt [[SUSv3]](#ID_REFSTD_46_LIBC_46_5) | getaddrinfo [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getcwd [[LSB]](#ID_REFSTD_46_LIBC_46_2) | getdate [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| getdomainname [[LSB]](#ID_REFSTD_46_LIBC_46_2) | getenv [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getlogin [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getlogin\_r [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| getnameinfo [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | getopt [[LSB]](#ID_REFSTD_46_LIBC_46_2) | getopt\_long [[LSB]](#ID_REFSTD_46_LIBC_46_2) | getopt\_long\_only [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| getsubopt [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | gettimeofday [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | glob [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | glob64 [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| globfree [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | globfree64 [[LSB]](#ID_REFSTD_46_LIBC_46_2) | grantpt [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | hcreate [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| hcreate\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) | hdestroy [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | hdestroy\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) | hsearch [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| hsearch\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) | htonl [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | htons [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | imaxabs [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| imaxdiv [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | inet\_addr [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | inet\_aton [[LSB]](#ID_REFSTD_46_LIBC_46_2) | inet\_ntoa [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| inet\_ntop [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | inet\_pton [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | initstate [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | initstate\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| insque [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | isatty [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | isblank [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | jrand48 [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| jrand48\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) | l64a [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | labs [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | lcong48 [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| lcong48\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) | ldiv [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | lfind [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | llabs [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| lldiv [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | longjmp [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | lrand48 [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | lrand48\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| lsearch [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | makecontext [[SUSv3]](#ID_REFSTD_46_LIBC_46_5) | malloc [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | memmem [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| mkdtemp [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | mkstemp [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | mktemp [[SUSv3]](#ID_REFSTD_46_LIBC_46_5) | mrand48 [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| mrand48\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) | nftw [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | nrand48 [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | nrand48\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| ntohl [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | ntohs [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | open\_memstream [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | open\_wmemstream(GLIBC\_2.4) [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| openlog [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | perror [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | posix\_openpt [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | ptsname [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| putenv [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | qsort [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | rand [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | rand\_r [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| random [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | random\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) | realloc [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | realpath [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| remque [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | scandir [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | scandir64 [[LSB]](#ID_REFSTD_46_LIBC_46_2) | seed48 [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| seed48\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) | sendfile [[LSB]](#ID_REFSTD_46_LIBC_46_2) | sendfile64(GLIBC\_2.3) [[LSB]](#ID_REFSTD_46_LIBC_46_2) | setenv [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| sethostname [[LSB]](#ID_REFSTD_46_LIBC_46_2) | setlogmask [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | setstate [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | setstate\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| srand [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | srand48 [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | srand48\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) | srandom [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| srandom\_r [[LSB]](#ID_REFSTD_46_LIBC_46_2) | strtod [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strtol [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | strtoul [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| swapcontext [[SUSv3]](#ID_REFSTD_46_LIBC_46_5) | syslog [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | system [[LSB]](#ID_REFSTD_46_LIBC_46_2) | tdelete [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| tfind [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | tmpfile [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | tmpnam [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | tsearch [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| ttyname [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | ttyname\_r [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | twalk [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | unlockpt [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| unsetenv [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | usleep [[SUSv3]](#ID_REFSTD_46_LIBC_46_5) | verrx [[LSB]](#ID_REFSTD_46_LIBC_46_2) | vfscanf [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| vscanf [[LSB]](#ID_REFSTD_46_LIBC_46_2) | vsscanf [[LSB]](#ID_REFSTD_46_LIBC_46_2) | vsyslog [[LSB]](#ID_REFSTD_46_LIBC_46_2) | warn [[LSB]](#ID_REFSTD_46_LIBC_46_2) |
| warnx [[LSB]](#ID_REFSTD_46_LIBC_46_2) | wordexp [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | wordfree [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |  |

An LSB conforming implementation shall provide the generic deprecated functions for Standard Library specified in [Table 14-35](#ID_TBL_45_LIBC_45_STANE_45_DEPINTS), with the full mandatory functionality as described in the referenced underlying specification.

**Note:** These interfaces are deprecated, and applications should avoid using them. These interfaces may be withdrawn in future releases of this specification.

**Table 14-35 libc - Standard Library Deprecated Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| basename [[LSB]](#ID_REFSTD_46_LIBC_46_2) | getdomainname [[LSB]](#ID_REFSTD_46_LIBC_46_2) | inet\_aton [[LSB]](#ID_REFSTD_46_LIBC_46_2) | tmpnam [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |

An LSB conforming implementation shall provide the generic data interfaces for Standard Library specified in [Table 14-36](#ID_TBL_45_LIBC_45_STANE_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-36 libc - Standard Library Data Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| \_\_environ [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_environ [[LSB]](#ID_REFSTD_46_LIBC_46_2) | \_sys\_errlist [[LSB]](#ID_REFSTD_46_LIBC_46_2) | environ [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| getdate\_err [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | optarg [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | opterr [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) | optind [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |
| optopt [[SUSv4]](#ID_REFSTD_46_LIBC_46_6) |  |  |  |

### **14.3.22 GNU Extensions for libc**

#### 14.3.22.1 Interfaces for GNU Extensions for libc

An LSB conforming implementation shall provide the generic functions for GNU Extensions for libc specified in [Table 14-37](#ID_TBL_45_LIBC_45_GNU_45_E_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-37 libc - GNU Extensions for libc Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| gnu\_get\_libc\_release [[LSB]](#ID_REFSTD_46_LIBC_46_2) | gnu\_get\_libc\_version [[LSB]](#ID_REFSTD_46_LIBC_46_2) |  |  |

## **14.4 Data Definitions for libc**

This section defines global identifiers and their values that are associated with interfaces contained in libc. These definitions are organized into groups that correspond to system headers. This convention is used as a convenience for the reader, and does not imply the existence of these headers, or their content. Where an interface is defined as requiring a particular system header file all of the data definitions for that system header file presented here shall be in effect.

This section gives data definitions to promote binary application portability, not to repeat source interface definitions available elsewhere. System providers and application developers should use this ABI to supplement - not to replace - source interface definition specifications.

This specification uses the [ISO C (1999)](#ID_STD_46_ISOC99) C Language as the reference programming language, and data definitions are specified in ISO C format. The C language is used here as a convenient notation. Using a C language description of these data objects does not preclude their use by other programming languages.

### **14.4.1 argz.h**

typedef int error\_t;

extern error\_t argz\_add(char \*\*argz, size\_t \* argz\_len, const char \*str);

extern error\_t argz\_add\_sep(char \*\*argz, size\_t \* argz\_len,

const char \*str, int sep);

extern error\_t argz\_append(char \*\*argz, size\_t \* argz\_len, const char \*buf,

size\_t buf\_len);

extern size\_t argz\_count(const char \*argz, size\_t \* argz\_len);

extern error\_t argz\_create(char \*const argv[], char \*\*argz,

size\_t \* argz\_len);

extern error\_t argz\_create\_sep(const char \*str, int sep, char \*\*argz,

size\_t \* argz\_len);

extern void argz\_delete(char \*\*argz, size\_t \* argz\_len, char \*entry);

extern void argz\_extract(const char \*argz, size\_t argz\_len, char \*\*argv);

extern error\_t argz\_insert(char \*\*argz\_insert, size\_t \* argz\_len,

char \*before, const char \*entry);

extern char argz\_next(const char \*argz, size\_t argz\_len,

const char \*entry);

extern error\_t argz\_replace(char \*\*argz, size\_t \* argz\_len,

const char \*str, const char \*with,

unsigned int \*replace\_count);

extern void argz\_stringify(char \*argz, size\_t argz\_len, int sep);

### **14.4.2 arpa/inet.h**

extern uint32\_t htonl(uint32\_t);

extern uint16\_t htons(uint16\_t);

extern in\_addr\_t inet\_addr(const char \*\_\_cp);

extern int inet\_aton(const char \*\_\_cp, struct in\_addr \*\_\_inp);

extern char \*inet\_ntoa(struct in\_addr \_\_in);

extern const char \*inet\_ntop(int \_\_af, const void \*\_\_cp, char \*\_\_buf,

socklen\_t \_\_len);

extern int inet\_pton(int \_\_af, const char \*\_\_cp, void \*\_\_buf);

extern uint32\_t ntohl(uint32\_t);

extern uint16\_t ntohs(uint16\_t);

### **14.4.3 assert.h**

#ifdef NDEBUG

#define assert(expr) ((void)0)

#else

#define assert(expr) ((void) ((expr) ? 0 : (\_\_assert\_fail (#expr, \_\_FILE\_\_, \_\_LINE\_\_, \_\_PRETTY\_FUNCTION\_\_), 0)))

#endif

extern void \_\_assert\_fail(const char \*\_\_assertion, const char \*\_\_file,

unsigned int \_\_line, const char \*\_\_function);

### **14.4.4 cpio.h**

#define C\_IXOTH 000001

#define C\_IWOTH 000002

#define C\_IROTH 000004

#define C\_IXGRP 000010

#define C\_IWGRP 000020

#define C\_IRGRP 000040

#define C\_IXUSR 000100

#define C\_IWUSR 000200

#define C\_IRUSR 000400

#define C\_ISVTX 001000

#define C\_ISGID 002000

#define C\_ISUID 004000

#define C\_ISFIFO 010000

#define C\_ISREG 0100000

#define C\_ISCTG 0110000

#define C\_ISLNK 0120000

#define C\_ISSOCK 0140000

#define C\_ISCHR 020000

#define C\_ISDIR 040000

#define C\_ISBLK 060000

#define MAGIC "070707"

### **14.4.5 ctype.h**

extern const unsigned short \*\*\_\_ctype\_b\_loc(void);

extern const int32\_t \*\*\_\_ctype\_tolower\_loc(void);

extern const int32\_t \*\*\_\_ctype\_toupper\_loc(void);

extern int \_tolower(int);

extern int \_toupper(int);

extern int isalnum(int);

extern int isalnum\_l(int c, locale\_t locale);

extern int isalpha(int);

extern int isalpha\_l(int c, locale\_t locale);

extern int isascii(int \_\_c);

extern int isblank(int);

extern int isblank\_l(int c, locale\_t locale);

extern int iscntrl(int);

extern int iscntrl\_l(int c, locale\_t locale);

extern int isdigit(int);

extern int isdigit\_l(int c, locale\_t locale);

extern int isgraph(int);

extern int isgraph\_l(int c, locale\_t locale);

extern int islower(int);

extern int islower\_l(int c, locale\_t locale);

extern int isprint(int);

extern int isprint\_l(int c, locale\_t locale);

extern int ispunct(int);

extern int ispunct\_l(int c, locale\_t locale);

extern int isspace(int);

extern int isspace\_l(int c, locale\_t locale);

extern int isupper(int);

extern int isupper\_l(int c, locale\_t locale);

extern int isxdigit(int);

extern int isxdigit\_l(int c, locale\_t locale);

extern int toascii(int \_\_c);

extern int tolower(int \_\_c);

extern int tolower\_l(int c, locale\_t locale);

extern int toupper(int \_\_c);

extern int toupper\_l(int c, locale\_t locale);

### **14.4.6 dirent.h**

#define MAXNAMLEN NAME\_MAX

typedef struct \_\_dirstream DIR;

struct dirent {

long int d\_ino;

off\_t d\_off;

unsigned short d\_reclen;

unsigned char d\_type;

char d\_name[256];

};

struct dirent64 {

uint64\_t d\_ino;

int64\_t d\_off;

unsigned short d\_reclen;

unsigned char d\_type;

char d\_name[256];

};

extern int alphasort(const struct dirent \*\*\_\_e1,

const struct dirent \*\*\_\_e2);

extern int alphasort64(const struct dirent64 \*\*\_\_e1,

const struct dirent64 \*\*\_\_e2);

extern int closedir(DIR \* \_\_dirp);

extern int dirfd(DIR \* \_\_dirp);

extern DIR \*fdopendir(int \_\_fd);

extern DIR \*opendir(const char \*\_\_name);

extern struct dirent \*readdir(DIR \* \_\_dirp);

extern struct dirent64 \*readdir64(DIR \* \_\_dirp);

extern int readdir64\_r(DIR \* \_\_dirp, struct dirent64 \*\_\_entry,

struct dirent64 \*\*\_\_result);

extern int readdir\_r(DIR \* \_\_dirp, struct dirent \*\_\_entry,

struct dirent \*\*\_\_result);

extern void rewinddir(DIR \* \_\_dirp);

extern int scandir(const char \*\_\_dir, struct dirent \*\*\*\_\_namelist,

int (\*\_\_selector) (const struct dirent \*),

int (\*\_\_cmp) (const struct dirent \* \*,

const struct dirent \* \*));

extern int scandir64(const char \*\_\_dir, struct dirent64 \*\*\*\_\_namelist,

int (\*\_\_selector) (const struct dirent64 \*),

int (\*\_\_cmp) (const struct dirent64 \* \*,

const struct dirent64 \* \*));

extern void seekdir(DIR \* \_\_dirp, long int \_\_pos);

extern long int telldir(DIR \* \_\_dirp);

### **14.4.7 elf.h**

#define ELFMAG1 'E'

#define ELFMAG3 'F'

#define ELFMAG2 'L'

#define ELF64\_R\_INFO(sym, type) ((((Elf64\_Xword) (sym)) << 32) + (type))

#define ELF32\_ST\_INFO(bind,type) (((bind) << 4) + ((type) & 0xf))

#define ELF32\_R\_INFO(sym, type) (((sym) << 8) + ((type) & 0xff))

#define ELF32\_M\_INFO(sym, size) (((sym) << 8) + (unsigned char) (size))

#define ELF32\_ST\_BIND(val) (((unsigned char) (val)) >> 4)

#define ELF64\_R\_TYPE(i) ((i) & 0xffffffff)

#define ELF64\_R\_SYM(i) ((i) >> 32)

#define ELF32\_M\_SYM(info) ((info) >> 8)

#define ELF32\_ST\_VISIBILITY(o) ((o) & 0x03)

#define ELF32\_M\_SIZE(info) ((unsigned char) (info))

#define ELF32\_ST\_TYPE(val) ((val) & 0xf)

#define ELF32\_R\_TYPE(val) ((val) & 0xff)

#define ELF32\_R\_SYM(val) ((val) >> 8)

#define PF\_X (1 << 0)

#define SHF\_WRITE (1 << 0)

#define PF\_W (1 << 1)

#define SHF\_ALLOC (1 << 1)

#define SHF\_TLS (1 << 10)

#define PF\_R (1 << 2)

#define SHF\_EXECINSTR (1 << 2)

#define SHF\_MERGE (1 << 4)

#define SHF\_STRINGS (1 << 5)

#define SHF\_INFO\_LINK (1 << 6)

#define SHF\_LINK\_ORDER (1 << 7)

#define SHF\_OS\_NONCONFORMING (1 << 8)

#define SHF\_GROUP (1 << 9)

#define EI\_NIDENT (16)

#define DT\_ADDRTAGIDX(tag) (DT\_ADDRRNGHI - (tag))

#define DT\_IA\_64\_PLT\_RESERVE (DT\_LOPROC + 0)

#define DT\_PPC64\_GLINK (DT\_LOPROC + 0)

#define DT\_PPC\_GOT (DT\_LOPROC + 0)

#define DT\_PPC64\_OPD (DT\_LOPROC + 1)

#define DT\_PPC64\_OPDSZ (DT\_LOPROC + 2)

#define DT\_VALTAGIDX(tag) (DT\_VALRNGHI - (tag))

#define DT\_VERSIONTAGIDX(tag) (DT\_VERNEEDNUM - (tag))

#define PT\_IA\_64\_ARCHEXT (PT\_LOPROC + 0)

#define PT\_IA\_64\_UNWIND (PT\_LOPROC + 1)

#define SHT\_IA\_64\_EXT (SHT\_LOPROC + 0)

#define SHT\_IA\_64\_UNWIND (SHT\_LOPROC + 1)

#define DT\_NULL 0

#define EI\_MAG0 0

#define ELFCLASSNONE 0

#define ELFDATANONE 0

#define ELFOSABI\_NONE 0

#define ELFOSABI\_SYSV 0

#define ELF\_NOTE\_OS\_LINUX 0

#define EM\_NONE 0

#define ET\_NONE 0

#define EV\_NONE 0

#define PT\_NULL 0

#define R\_386\_NONE 0

#define R\_390\_NONE 0

#define R\_PPC\_NONE 0

#define R\_X86\_64\_NONE 0

#define SHN\_UNDEF 0

#define SHT\_NULL 0

#define STB\_LOCAL 0

#define STN\_UNDEF 0

#define STT\_NOTYPE 0

#define STV\_DEFAULT 0

#define SYMINFO\_NONE 0

#define R\_IA64\_NONE 0x00

#define DF\_1\_NOW 0x00000001

#define DF\_ORIGIN 0x00000001 /\* Object may use DF\_ORIGIN \*/

#define DF\_P1\_LAZYLOAD 0x00000001

#define DTF\_1\_PARINIT 0x00000001

#define EF\_S390\_HIGH\_GPRS 0x00000001

#define DF\_1\_GLOBAL 0x00000002

#define DF\_P1\_GROUPPERM 0x00000002

#define DF\_SYMBOLIC 0x00000002 /\* Symbol resolutions start with this object \*/

#define DTF\_1\_CONFEXP 0x00000002

#define DF\_1\_GROUP 0x00000004

#define DF\_TEXTREL 0x00000004 /\* Object contains text relocations \*/

#define DF\_1\_NODELETE 0x00000008

#define DF\_BIND\_NOW 0x00000008 /\* No lazy binding for this object \*/

#define EF\_IA\_64\_MASKOS 0x0000000f

#define DF\_1\_LOADFLTR 0x00000010

#define DF\_STATIC\_TLS 0x00000010 /\* Module uses the static TLS model \*/

#define EF\_IA\_64\_ABI64 0x00000010

#define DF\_1\_INITFIRST 0x00000020

#define DF\_1\_NOOPEN 0x00000040

#define DF\_1\_ORIGIN 0x00000080

#define DF\_1\_DIRECT 0x00000100

#define DF\_1\_TRANS 0x00000200

#define DF\_1\_INTERPOSE 0x00000400

#define DF\_1\_NODEFLIB 0x00000800

#define DF\_1\_NODUMP 0x00001000

#define DF\_1\_CONFALT 0x00002000

#define DF\_1\_ENDFILTEE 0x00004000

#define DF\_1\_DISPRELDNE 0x00008000

#define SYMINFO\_FLG\_DIRECT 0x0001

#define DF\_1\_DISPRELPND 0x00010000

#define SYMINFO\_FLG\_PASSTHRU 0x0002

#define SYMINFO\_FLG\_COPY 0x0004

#define SYMINFO\_FLG\_LAZYLOAD 0x0008

#define EF\_CPU32 0x00810000

#define PF\_MASKOS 0x0ff00000

#define SHF\_MASKOS 0x0ff00000

#define GRP\_COMDAT 0x1

#define SHF\_IA\_64\_SHORT 0x10000000

#define SHF\_IA\_64\_NORECOV 0x20000000

#define R\_IA64\_IMM14 0x21

#define R\_IA64\_IMM22 0x22

#define R\_IA64\_IMM64 0x23

#define R\_IA64\_DIR32MSB 0x24

#define R\_IA64\_DIR32LSB 0x25

#define R\_IA64\_DIR64MSB 0x26

#define R\_IA64\_DIR64LSB 0x27

#define R\_IA64\_GPREL22 0x2a

#define R\_IA64\_GPREL64I 0x2b

#define R\_IA64\_GPREL32MSB 0x2c

#define R\_IA64\_GPREL32LSB 0x2d

#define R\_IA64\_GPREL64MSB 0x2e

#define R\_IA64\_GPREL64LSB 0x2f

#define R\_IA64\_LTOFF22 0x32

#define R\_IA64\_LTOFF64I 0x33

#define R\_IA64\_PLTOFF22 0x3a

#define R\_IA64\_PLTOFF64I 0x3b

#define R\_IA64\_PLTOFF64MSB 0x3e

#define R\_IA64\_PLTOFF64LSB 0x3f

#define R\_IA64\_FPTR64I 0x43

#define R\_IA64\_FPTR32MSB 0x44

#define R\_IA64\_FPTR32LSB 0x45

#define R\_IA64\_FPTR64MSB 0x46

#define R\_IA64\_FPTR64LSB 0x47

#define R\_IA64\_PCREL60B 0x48

#define R\_IA64\_PCREL21B 0x49

#define R\_IA64\_PCREL21M 0x4a

#define R\_IA64\_PCREL21F 0x4b

#define R\_IA64\_PCREL32MSB 0x4c

#define R\_IA64\_PCREL32LSB 0x4d

#define R\_IA64\_PCREL64MSB 0x4e

#define R\_IA64\_PCREL64LSB 0x4f

#define R\_IA64\_LTOFF\_FPTR22 0x52

#define R\_IA64\_LTOFF\_FPTR64I 0x53

#define R\_IA64\_LTOFF\_FPTR32MSB 0x54

#define R\_IA64\_LTOFF\_FPTR32LSB 0x55

#define R\_IA64\_LTOFF\_FPTR64MSB 0x56

#define R\_IA64\_LTOFF\_FPTR64LSB 0x57

#define R\_IA64\_SEGREL32MSB 0x5c

#define R\_IA64\_SEGREL32LSB 0x5d

#define R\_IA64\_SEGREL64MSB 0x5e

#define R\_IA64\_SEGREL64LSB 0x5f

#define PT\_LOOS 0x60000000

#define SHT\_LOOS 0x60000000

#define DT\_LOOS 0x6000000d

#define R\_IA64\_SECREL32MSB 0x64

#define PT\_GNU\_EH\_FRAME 0x6474e550

#define PT\_GNU\_STACK 0x6474e551

#define PT\_GNU\_RELRO 0x6474e552

#define R\_IA64\_SECREL32LSB 0x65

#define R\_IA64\_SECREL64MSB 0x66

#define R\_IA64\_SECREL64LSB 0x67

#define R\_IA64\_REL32MSB 0x6c

#define R\_IA64\_REL32LSB 0x6d

#define R\_IA64\_REL64MSB 0x6e

#define R\_IA64\_REL64LSB 0x6f

#define DT\_HIOS 0x6ffff000

#define DT\_VALRNGLO 0x6ffffd00

#define DT\_GNU\_PRELINKED 0x6ffffdf5

#define DT\_GNU\_CONFLICTSZ 0x6ffffdf6

#define DT\_GNU\_LIBLISTSZ 0x6ffffdf7

#define DT\_CHECKSUM 0x6ffffdf8

#define DT\_PLTPADSZ 0x6ffffdf9

#define DT\_MOVEENT 0x6ffffdfa

#define DT\_MOVESZ 0x6ffffdfb

#define DT\_FEATURE\_1 0x6ffffdfc

#define DT\_POSFLAG\_1 0x6ffffdfd

#define DT\_SYMINSZ 0x6ffffdfe

#define DT\_SYMINENT 0x6ffffdff

#define DT\_VALRNGHI 0x6ffffdff

#define DT\_ADDRRNGLO 0x6ffffe00

#define DT\_GNU\_HASH 0x6ffffef5

#define DT\_TLSDESC\_PLT 0x6ffffef6

#define DT\_TLSDESC\_GOT 0x6ffffef7

#define DT\_GNU\_CONFLICT 0x6ffffef8

#define DT\_GNU\_LIBLIST 0x6ffffef9

#define DT\_CONFIG 0x6ffffefa

#define DT\_DEPAUDIT 0x6ffffefb

#define DT\_AUDIT 0x6ffffefc

#define DT\_PLTPAD 0x6ffffefd

#define DT\_MOVETAB 0x6ffffefe

#define DT\_ADDRRNGHI 0x6ffffeff

#define DT\_SYMINFO 0x6ffffeff

#define DT\_VERSYM 0x6ffffff0

#define SHT\_GNU\_ATTRIBUTES 0x6ffffff5

#define SHT\_GNU\_HASH 0x6ffffff6

#define SHT\_GNU\_LIBLIST 0x6ffffff7

#define SHT\_CHECKSUM 0x6ffffff8

#define DT\_RELACOUNT 0x6ffffff9

#define DT\_RELCOUNT 0x6ffffffa

#define DT\_FLAGS\_1 0x6ffffffb

#define DT\_VERDEF 0x6ffffffc

#define DT\_VERDEFNUM 0x6ffffffd

#define SHT\_GNU\_verdef 0x6ffffffd

#define DT\_VERNEED 0x6ffffffe

#define SHT\_GNU\_verneed 0x6ffffffe

#define DT\_VERNEEDNUM 0x6fffffff

#define PT\_HIOS 0x6fffffff

#define SHT\_GNU\_versym 0x6fffffff

#define SHT\_HIOS 0x6fffffff

#define DT\_LOPROC 0x70000000

#define PT\_LOPROC 0x70000000

#define SHT\_LOPROC 0x70000000

#define R\_IA64\_LTV32MSB 0x74

#define R\_IA64\_LTV32LSB 0x75

#define R\_IA64\_LTV64MSB 0x76

#define R\_IA64\_LTV64LSB 0x77

#define R\_IA64\_PCREL21BI 0x79

#define R\_IA64\_PCREL22 0x7a

#define R\_IA64\_PCREL64I 0x7b

#define ELFMAG0 0x7f

#define DT\_AUXILIARY 0x7ffffffd

#define DT\_FILTER 0x7fffffff

#define DT\_HIPROC 0x7fffffff

#define PT\_HIPROC 0x7fffffff

#define SHT\_HIPROC 0x7fffffff

#define R\_IA64\_IPLTMSB 0x80

#define PF\_IA\_64\_NORECOV 0x80000000

#define SHT\_LOUSER 0x80000000

#define R\_IA64\_IPLTLSB 0x81

#define R\_IA64\_COPY 0x84

#define R\_IA64\_SUB 0x85

#define R\_IA64\_LTOFF22X 0x86

#define R\_IA64\_LDXMOV 0x87

#define SHT\_HIUSER 0x8fffffff

#define R\_IA64\_TPREL14 0x91

#define R\_IA64\_TPREL22 0x92

#define R\_IA64\_TPREL64I 0x93

#define R\_IA64\_TPREL64MSB 0x96

#define R\_IA64\_TPREL64LSB 0x97

#define R\_IA64\_LTOFF\_TPREL22 0x9a

#define R\_IA64\_DTPMOD64MSB 0xa6

#define R\_IA64\_DTPMOD64LSB 0xa7

#define R\_IA64\_LTOFF\_DTPMOD22 0xaa

#define R\_IA64\_DTPREL14 0xb1

#define R\_IA64\_DTPREL22 0xb2

#define R\_IA64\_DTPREL64I 0xb3

#define R\_IA64\_DTPREL32MSB 0xb4

#define R\_IA64\_DTPREL32LSB 0xb5

#define R\_IA64\_DTPREL64MSB 0xb6

#define R\_IA64\_DTPREL64LSB 0xb7

#define R\_IA64\_LTOFF\_DTPREL22 0xba

#define PF\_MASKPROC 0xf0000000

#define SHF\_MASKPROC 0xf0000000

#define ET\_LOOS 0xfe00

#define ET\_HIOS 0xfeff

#define ET\_LOPROC 0xff00

#define SHN\_LOPROC 0xff00

#define SHN\_LORESERVE 0xff00

#define SYMINFO\_BT\_LOWRESERVE 0xff00

#define EF\_IA\_64\_ARCH 0xff000000

#define SHN\_HIPROC 0xff1f

#define SHN\_LOOS 0xff20

#define SHN\_HIOS 0xff3f

#define SHN\_ABS 0xfff1

#define SHN\_COMMON 0xfff2

#define SYMINFO\_BT\_PARENT 0xfffe

#define ET\_HIPROC 0xffff

#define PN\_XNUM 0xffff

#define SHN\_HIRESERVE 0xffff

#define SHN\_XINDEX 0xffff

#define SYMINFO\_BT\_SELF 0xffff

#define DT\_IA\_64\_NUM 1

#define DT\_NEEDED 1

#define DT\_PPC\_NUM 1

#define EI\_MAG1 1

#define ELFCLASS32 1

#define ELFDATA2LSB 1

#define ELF\_NOTE\_OS\_GNU 1

#define ET\_REL 1

#define EV\_CURRENT 1

#define NT\_GNU\_ABI\_TAG 1

#define PT\_LOAD 1

#define R\_386\_32 1

#define R\_390\_8 1

#define R\_PPC\_ADDR32 1

#define R\_X86\_64\_64 1

#define SHT\_PROGBITS 1

#define STB\_GLOBAL 1

#define STT\_OBJECT 1

#define STV\_INTERNAL 1

#define SYMINFO\_CURRENT 1

#define DT\_STRSZ 10

#define R\_386\_GOTPC 10

#define R\_390\_GLOB\_DAT 10

#define R\_PPC\_REL24 10

#define R\_X86\_64\_32 10

#define SHT\_SHLIB 10

#define STB\_LOOS 10

#define STT\_GNU\_IFUNC 10

#define STT\_LOOS 10

#define R\_PPC64\_TPREL16\_HIGHESTA 100

#define R\_PPC64\_DTPREL16\_DS 101

#define R\_PPC64\_DTPREL16\_LO\_DS 102

#define R\_PPC64\_DTPREL16\_HIGHER 103

#define R\_PPC64\_DTPREL16\_HIGHERA 104

#define R\_PPC64\_DTPREL16\_HIGHEST 105

#define R\_PPC64\_DTPREL16\_HIGHESTA 106

#define DT\_ADDRNUM 11

#define DT\_SYMENT 11

#define R\_386\_32PLT 11

#define R\_390\_JMP\_SLOT 11

#define R\_PPC\_REL14 11

#define R\_X86\_64\_32S 11

#define SHT\_DYNSYM 11

#define DT\_INIT 12

#define DT\_VALNUM 12

#define R\_390\_RELATIVE 12

#define R\_PPC\_REL14\_BRTAKEN 12

#define R\_X86\_64\_16 12

#define STB\_HIOS 12

#define STT\_HIOS 12

#define DT\_FINI 13

#define R\_390\_GOTOFF32 13

#define R\_PPC\_REL14\_BRNTAKEN 13

#define R\_X86\_64\_PC16 13

#define STB\_LOPROC 13

#define STT\_LOPROC 13

#define DT\_SONAME 14

#define R\_386\_TLS\_TPOFF 14

#define R\_390\_GOTPC 14

#define R\_PPC\_GOT16 14

#define R\_X86\_64\_8 14

#define SHT\_INIT\_ARRAY 14

#define DT\_RPATH 15

#define R\_386\_TLS\_IE 15

#define R\_390\_GOT16 15

#define R\_PPC\_GOT16\_LO 15

#define R\_X86\_64\_PC8 15

#define SHT\_FINI\_ARRAY 15

#define STB\_HIPROC 15

#define STT\_HIPROC 15

#define DT\_SYMBOLIC 16

#define DT\_VERSIONTAGNUM 16

#define R\_386\_TLS\_GOTIE 16

#define R\_390\_PC16 16

#define R\_PPC\_GOT16\_HI 16

#define R\_X86\_64\_DTPMOD64 16

#define SHT\_PREINIT\_ARRAY 16

#define DT\_REL 17

#define R\_386\_TLS\_LE 17

#define R\_390\_PC16DBL 17

#define R\_PPC\_GOT16\_HA 17

#define R\_X86\_64\_DTPOFF64 17

#define SHT\_GROUP 17

#define DT\_RELSZ 18

#define R\_386\_TLS\_GD 18

#define R\_390\_PLT16DBL 18

#define R\_PPC\_PLTREL24 18

#define R\_X86\_64\_TPOFF64 18

#define SHT\_SYMTAB\_SHNDX 18

#define DT\_RELENT 19

#define R\_386\_TLS\_LDM 19

#define R\_390\_PC32DBL 19

#define R\_PPC\_COPY 19

#define R\_X86\_64\_TLSGD 19

#define DT\_PLTRELSZ 2

#define EI\_MAG2 2

#define ELFCLASS64 2

#define ELFDATA2MSB 2

#define ELF\_NOTE\_OS\_SOLARIS2 2

#define ET\_EXEC 2

#define EV\_NUM 2

#define PT\_DYNAMIC 2

#define R\_386\_PC32 2

#define R\_390\_12 2

#define R\_PPC\_ADDR24 2

#define R\_X86\_64\_PC32 2

#define SHT\_SYMTAB 2

#define STB\_WEAK 2

#define STT\_FUNC 2

#define STV\_HIDDEN 2

#define SYMINFO\_NUM 2

#define DT\_PLTREL 20

#define EM\_PPC 20

#define R\_386\_16 20

#define R\_390\_PLT32DBL 20

#define R\_PPC\_GLOB\_DAT 20

#define R\_X86\_64\_TLSLD 20

#define DT\_DEBUG 21

#define EM\_PPC64 21

#define R\_386\_PC16 21

#define R\_390\_GOTPCDBL 21

#define R\_PPC\_JMP\_SLOT 21

#define R\_X86\_64\_DTPOFF32 21

#define DT\_TEXTREL 22

#define EM\_S390 22

#define R\_386\_8 22

#define R\_390\_64 22

#define R\_PPC\_RELATIVE 22

#define R\_X86\_64\_GOTTPOFF 22

#define DT\_JMPREL 23

#define R\_386\_PC8 23

#define R\_390\_PC64 23

#define R\_PPC\_LOCAL24PC 23

#define R\_X86\_64\_TPOFF32 23

#define DT\_BIND\_NOW 24

#define R\_386\_TLS\_GD\_32 24

#define R\_390\_GOT64 24

#define R\_PPC\_UADDR32 24

#define R\_X86\_64\_PC64 24

#define R\_PPC64\_JMP\_IREL 247

#define R\_PPC64\_IRELATIVE 248

#define R\_PPC\_IRELATIVE 248

#define R\_PPC64\_REL16 249

#define R\_PPC\_REL16 249

#define DT\_INIT\_ARRAY 25

#define R\_386\_TLS\_GD\_PUSH 25

#define R\_390\_PLT64 25

#define R\_PPC\_UADDR16 25

#define R\_X86\_64\_GOTOFF64 25

#define R\_PPC64\_REL16\_LO 250

#define R\_PPC\_REL16\_LO 250

#define R\_PPC64\_REL16\_HI 251

#define R\_PPC\_REL16\_HI 251

#define R\_PPC64\_REL16\_HA 252

#define R\_PPC\_REL16\_HA 252

#define R\_PPC\_TOC16 255

#define DT\_FINI\_ARRAY 26

#define R\_386\_TLS\_GD\_CALL 26

#define R\_390\_GOTENT 26

#define R\_PPC\_REL32 26

#define R\_X86\_64\_GOTPC32 26

#define DT\_INIT\_ARRAYSZ 27

#define R\_386\_TLS\_GD\_POP 27

#define R\_390\_GOTOFF16 27

#define R\_PPC\_PLT32 27

#define R\_X86\_64\_GOT64 27

#define DT\_FINI\_ARRAYSZ 28

#define R\_386\_TLS\_LDM\_32 28

#define R\_390\_GOTOFF64 28

#define R\_PPC\_PLTREL32 28

#define R\_X86\_64\_GOTPCREL64 28

#define DT\_RUNPATH 29

#define R\_386\_TLS\_LDM\_PUSH 29

#define R\_390\_GOTPLT12 29

#define R\_PPC\_PLT16\_LO 29

#define R\_X86\_64\_GOTPC64 29

#define DT\_EXTRANUM 3

#define DT\_PLTGOT 3

#define DT\_PPC64\_NUM 3

#define EI\_MAG3 3

#define ELFCLASSNUM 3

#define ELFDATANUM 3

#define ELFOSABI\_LINUX 3

#define ELF\_NOTE\_OS\_FREEBSD 3

#define EM\_386 3

#define ET\_DYN 3

#define PT\_INTERP 3

#define R\_386\_GOT32 3

#define R\_390\_16 3

#define R\_PPC\_ADDR16 3

#define R\_X86\_64\_GOT32 3

#define SHT\_STRTAB 3

#define STB\_NUM 3

#define STT\_SECTION 3

#define STV\_PROTECTED 3

#define DT\_FLAGS 30

#define R\_386\_TLS\_LDM\_CALL 30

#define R\_390\_GOTPLT16 30

#define R\_PPC\_PLT16\_HI 30

#define R\_X86\_64\_GOTPLT64 30

#define R\_386\_TLS\_LDM\_POP 31

#define R\_390\_GOTPLT32 31

#define R\_PPC\_PLT16\_HA 31

#define R\_X86\_64\_PLTOFF64 31

#define DT\_ENCODING 32

#define DT\_PREINIT\_ARRAY 32

#define R\_386\_TLS\_LDO\_32 32

#define R\_390\_GOTPLT64 32

#define R\_PPC\_SDAREL16 32

#define R\_X86\_64\_SIZE32 32

#define DT\_PREINIT\_ARRAYSZ 33

#define R\_386\_TLS\_IE\_32 33

#define R\_390\_GOTPLTENT 33

#define R\_PPC\_SECTOFF 33

#define R\_X86\_64\_SIZE64 33

#define DT\_NUM 34

#define R\_386\_TLS\_LE\_32 34

#define R\_390\_PLTOFF16 34

#define R\_PPC\_SECTOFF\_LO 34

#define R\_X86\_64\_GOTPC32\_TLSDESC 34

#define R\_386\_TLS\_DTPMOD32 35

#define R\_390\_PLTOFF32 35

#define R\_PPC\_SECTOFF\_HI 35

#define R\_X86\_64\_TLSDESC\_CALL 35

#define R\_386\_TLS\_DTPOFF32 36

#define R\_390\_PLTOFF64 36

#define R\_PPC\_SECTOFF\_HA 36

#define R\_X86\_64\_TLSDESC 36

#define R\_386\_TLS\_TPOFF32 37

#define R\_390\_TLS\_LOAD 37

#define R\_PPC64\_ADDR30 37

#define R\_X86\_64\_IRELATIVE 37

#define R\_390\_TLS\_GDCALL 38

#define R\_PPC64\_ADDR64 38

#define R\_X86\_64\_NUM 38

#define R\_386\_TLS\_GOTDESC 39

#define R\_390\_TLS\_LDCALL 39

#define R\_PPC64\_ADDR16\_HIGHER 39

#define DT\_HASH 4

#define EI\_CLASS 4

#define ET\_CORE 4

#define PT\_NOTE 4

#define R\_386\_PLT32 4

#define R\_390\_32 4

#define R\_PPC\_ADDR16\_LO 4

#define R\_X86\_64\_PLT32 4

#define SELFMAG 4

#define SHT\_RELA 4

#define STT\_FILE 4

#define EM\_ARM 40

#define R\_386\_TLS\_DESC\_CALL 40

#define R\_390\_TLS\_GD32 40

#define R\_PPC64\_ADDR16\_HIGHERA 40

#define R\_386\_TLS\_DESC 41

#define R\_390\_TLS\_GD64 41

#define R\_PPC64\_ADDR16\_HIGHEST 41

#define R\_386\_IRELATIVE 42

#define R\_390\_TLS\_GOTIE12 42

#define R\_PPC64\_ADDR16\_HIGHESTA 42

#define R\_386\_NUM 43

#define R\_390\_TLS\_GOTIE32 43

#define R\_PPC64\_UADDR64 43

#define R\_390\_TLS\_GOTIE64 44

#define R\_PPC64\_REL64 44

#define R\_390\_TLS\_LDM32 45

#define R\_PPC64\_PLT64 45

#define R\_390\_TLS\_LDM64 46

#define R\_PPC64\_PLTREL64 46

#define R\_390\_TLS\_IE32 47

#define R\_PPC64\_TOC16 47

#define R\_390\_TLS\_IE64 48

#define R\_PPC64\_TOC16\_LO 48

#define R\_390\_TLS\_IEENT 49

#define R\_PPC64\_TOC16\_HI 49

#define DT\_STRTAB 5

#define EI\_DATA 5

#define ET\_NUM 5

#define PT\_SHLIB 5

#define R\_386\_COPY 5

#define R\_390\_PC32 5

#define R\_PPC\_ADDR16\_HI 5

#define R\_X86\_64\_COPY 5

#define SHT\_HASH 5

#define STT\_COMMON 5

#define EM\_IA\_64 50

#define R\_390\_TLS\_LE32 50

#define R\_PPC64\_TOC16\_HA 50

#define R\_390\_TLS\_LE64 51

#define R\_PPC64\_TOC 51

#define R\_390\_TLS\_LDO32 52

#define R\_PPC64\_PLTGOT16 52

#define R\_390\_TLS\_LDO64 53

#define R\_PPC64\_PLTGOT16\_LO 53

#define R\_390\_TLS\_DTPMOD 54

#define R\_PPC64\_PLTGOT16\_HI 54

#define R\_390\_TLS\_DTPOFF 55

#define R\_PPC64\_PLTGOT16\_HA 55

#define R\_390\_TLS\_TPOFF 56

#define R\_PPC64\_ADDR16\_DS 56

#define R\_390\_20 57

#define R\_PPC64\_ADDR16\_LO\_DS 57

#define R\_390\_GOT20 58

#define R\_PPC64\_GOT16\_DS 58

#define R\_390\_GOTPLT20 59

#define R\_PPC64\_GOT16\_LO\_DS 59

#define DT\_SYMTAB 6

#define EI\_VERSION 6

#define PT\_PHDR 6

#define R\_386\_GLOB\_DAT 6

#define R\_390\_GOT12 6

#define R\_PPC\_ADDR16\_HA 6

#define R\_X86\_64\_GLOB\_DAT 6

#define SHT\_DYNAMIC 6

#define STT\_TLS 6

#define R\_390\_TLS\_GOTIE20 60

#define R\_PPC64\_PLT16\_LO\_DS 60

#define R\_390\_NUM 61

#define R\_PPC64\_SECTOFF\_DS 61

#define EM\_X86\_64 62

#define R\_PPC64\_SECTOFF\_LO\_DS 62

#define R\_PPC64\_TOC16\_DS 63

#define R\_PPC64\_TOC16\_LO\_DS 64

#define R\_PPC64\_PLTGOT16\_DS 65

#define R\_PPC64\_PLTGOT16\_LO\_DS 66

#define R\_PPC64\_TLS 67

#define R\_PPC\_TLS 67

#define R\_PPC64\_DTPMOD64 68

#define R\_PPC\_DTPMOD32 68

#define R\_PPC64\_TPREL16 69

#define R\_PPC\_TPREL16 69

#define DT\_RELA 7

#define EI\_OSABI 7

#define PT\_TLS 7

#define R\_386\_JMP\_SLOT 7

#define R\_390\_GOT32 7

#define R\_PPC\_ADDR14 7

#define R\_X86\_64\_JUMP\_SLOT 7

#define SHT\_NOTE 7

#define STT\_NUM 7

#define R\_PPC64\_TPREL16\_LO 70

#define R\_PPC\_TPREL16\_LO 70

#define R\_PPC64\_TPREL16\_HI 71

#define R\_PPC\_TPREL16\_HI 71

#define R\_PPC64\_TPREL16\_HA 72

#define R\_PPC\_TPREL16\_HA 72

#define R\_PPC64\_TPREL64 73

#define R\_PPC\_TPREL32 73

#define R\_PPC64\_DTPREL16 74

#define R\_PPC\_DTPREL16 74

#define R\_PPC64\_DTPREL16\_LO 75

#define R\_PPC\_DTPREL16\_LO 75

#define R\_PPC64\_DTPREL16\_HI 76

#define R\_PPC\_DTPREL16\_HI 76

#define R\_PPC64\_DTPREL16\_HA 77

#define R\_PPC\_DTPREL16\_HA 77

#define R\_PPC64\_DTPREL64 78

#define R\_PPC\_DTPREL32 78

#define R\_PPC64\_GOT\_TLSGD16 79

#define R\_PPC\_GOT\_TLSGD16 79

#define DT\_RELASZ 8

#define EI\_ABIVERSION 8

#define PT\_NUM 8

#define R\_386\_RELATIVE 8

#define R\_390\_PLT32 8

#define R\_PPC\_ADDR14\_BRTAKEN 8

#define R\_X86\_64\_RELATIVE 8

#define SHT\_NOBITS 8

#define R\_PPC64\_GOT\_TLSGD16\_LO 80

#define R\_PPC\_GOT\_TLSGD16\_LO 80

#define R\_PPC64\_GOT\_TLSGD16\_HI 81

#define R\_PPC\_GOT\_TLSGD16\_HI 81

#define R\_PPC64\_GOT\_TLSGD16\_HA 82

#define R\_PPC\_GOT\_TLSGD16\_HA 82

#define R\_PPC64\_GOT\_TLSLD16 83

#define R\_PPC\_GOT\_TLSLD16 83

#define R\_PPC64\_GOT\_TLSLD16\_LO 84

#define R\_PPC\_GOT\_TLSLD16\_LO 84

#define R\_PPC64\_GOT\_TLSLD16\_HI 85

#define R\_PPC\_GOT\_TLSLD16\_HI 85

#define R\_PPC64\_GOT\_TLSLD16\_HA 86

#define R\_PPC\_GOT\_TLSLD16\_HA 86

#define R\_PPC64\_GOT\_TPREL16\_DS 87

#define R\_PPC\_GOT\_TPREL16 87

#define R\_PPC64\_GOT\_TPREL16\_LO\_DS 88

#define R\_PPC\_GOT\_TPREL16\_LO 88

#define R\_PPC64\_GOT\_TPREL16\_HI 89

#define R\_PPC\_GOT\_TPREL16\_HI 89

#define DT\_RELAENT 9

#define EI\_PAD 9

#define R\_386\_GOTOFF 9

#define R\_390\_COPY 9

#define R\_PPC\_ADDR14\_BRNTAKEN 9

#define R\_X86\_64\_GOTPCREL 9

#define SHT\_REL 9

#define R\_PPC64\_GOT\_TPREL16\_HA 90

#define R\_PPC\_GOT\_TPREL16\_HA 90

#define R\_PPC64\_GOT\_DTPREL16\_DS 91

#define R\_PPC\_GOT\_DTPREL16 91

#define R\_PPC64\_GOT\_DTPREL16\_LO\_DS 92

#define R\_PPC\_GOT\_DTPREL16\_LO 92

#define R\_PPC64\_GOT\_DTPREL16\_HI 93

#define R\_PPC\_GOT\_DTPREL16\_HI 93

#define R\_PPC64\_GOT\_DTPREL16\_HA 94

#define R\_PPC\_GOT\_DTPREL16\_HA 94

#define R\_PPC64\_TPREL16\_DS 95

#define R\_PPC64\_TPREL16\_LO\_DS 96

#define R\_PPC64\_TPREL16\_HIGHER 97

#define R\_PPC64\_TPREL16\_HIGHERA 98

#define R\_PPC64\_TPREL16\_HIGHEST 99

#define ELF64\_M\_INFO(sym, size) ELF32\_M\_INFO (sym, size)

#define ELF64\_M\_SIZE(info) ELF32\_M\_SIZE (info)

#define ELF64\_M\_SYM(info) ELF32\_M\_SYM (info)

#define ELF64\_ST\_BIND(val) ELF32\_ST\_BIND (val)

#define ELF64\_ST\_INFO(bind,type) ELF32\_ST\_INFO ((bind), (type))

#define ELF64\_ST\_TYPE(val) ELF32\_ST\_TYPE (val)

#define ELF64\_ST\_VISIBILITY(o) ELF32\_ST\_VISIBILITY(o)

#define ELF\_NOTE\_GNU "GNU"

#define ELF\_NOTE\_ABI NT\_GNU\_ABI\_TAG

#define R\_PPC64\_ADDR14 R\_PPC\_ADDR14

#define R\_PPC64\_ADDR14\_BRNTAKEN R\_PPC\_ADDR14\_BRNTAKEN

#define R\_PPC64\_ADDR14\_BRTAKEN R\_PPC\_ADDR14\_BRTAKEN

#define R\_PPC64\_ADDR16 R\_PPC\_ADDR16

#define R\_PPC64\_ADDR16\_HA R\_PPC\_ADDR16\_HA

#define R\_PPC64\_ADDR16\_HI R\_PPC\_ADDR16\_HI

#define R\_PPC64\_ADDR16\_LO R\_PPC\_ADDR16\_LO

#define R\_PPC64\_ADDR24 R\_PPC\_ADDR24

#define R\_PPC64\_ADDR32 R\_PPC\_ADDR32

#define R\_PPC64\_COPY R\_PPC\_COPY

#define R\_PPC64\_GLOB\_DAT R\_PPC\_GLOB\_DAT

#define R\_PPC64\_GOT16 R\_PPC\_GOT16

#define R\_PPC64\_GOT16\_HA R\_PPC\_GOT16\_HA

#define R\_PPC64\_GOT16\_HI R\_PPC\_GOT16\_HI

#define R\_PPC64\_GOT16\_LO R\_PPC\_GOT16\_LO

#define R\_PPC64\_JMP\_SLOT R\_PPC\_JMP\_SLOT

#define R\_PPC64\_NONE R\_PPC\_NONE

#define R\_PPC64\_PLT16\_HA R\_PPC\_PLT16\_HA

#define R\_PPC64\_PLT16\_HI R\_PPC\_PLT16\_HI

#define R\_PPC64\_PLT16\_LO R\_PPC\_PLT16\_LO

#define R\_PPC64\_PLT32 R\_PPC\_PLT32

#define R\_PPC64\_PLTREL32 R\_PPC\_PLTREL32

#define R\_PPC64\_REL14 R\_PPC\_REL14

#define R\_PPC64\_REL14\_BRNTAKEN R\_PPC\_REL14\_BRNTAKEN

#define R\_PPC64\_REL14\_BRTAKEN R\_PPC\_REL14\_BRTAKEN

#define R\_PPC64\_REL24 R\_PPC\_REL24

#define R\_PPC64\_REL32 R\_PPC\_REL32

#define R\_PPC64\_RELATIVE R\_PPC\_RELATIVE

#define R\_PPC64\_SECTOFF R\_PPC\_SECTOFF

#define R\_PPC64\_SECTOFF\_HA R\_PPC\_SECTOFF\_HA

#define R\_PPC64\_SECTOFF\_HI R\_PPC\_SECTOFF\_HI

#define R\_PPC64\_SECTOFF\_LO R\_PPC\_SECTOFF\_LO

#define R\_PPC64\_UADDR16 R\_PPC\_UADDR16

#define R\_PPC64\_UADDR32 R\_PPC\_UADDR32

#define ELFMAG "\177ELF"

typedef uint32\_t Elf32\_Addr;

typedef uint64\_t Elf64\_Addr;

typedef uint32\_t Elf32\_Word;

typedef uint32\_t Elf64\_Word;

typedef int32\_t Elf32\_Sword;

typedef int32\_t Elf64\_Sword;

typedef uint64\_t Elf32\_Xword;

typedef uint64\_t Elf64\_Xword;

typedef int64\_t Elf32\_Sxword;

typedef int64\_t Elf64\_Sxword;

typedef uint32\_t Elf32\_Off;

typedef uint64\_t Elf64\_Off;

typedef struct {

Elf32\_Word p\_type; /\* Segment type \*/

Elf32\_Off p\_offset; /\* Segment file offset \*/

Elf32\_Addr p\_vaddr; /\* Segment virtual address \*/

Elf32\_Addr p\_paddr; /\* Segment physical address \*/

Elf32\_Word p\_filesz; /\* Segment size in file \*/

Elf32\_Word p\_memsz; /\* Segment size in memory \*/

Elf32\_Word p\_flags; /\* Segment flags \*/

Elf32\_Word p\_align; /\* Segment alignment \*/

} Elf32\_Phdr;

typedef struct {

Elf64\_Word p\_type; /\* Segment type \*/

Elf64\_Word p\_flags; /\* Segment flags \*/

Elf64\_Off p\_offset; /\* Segment file offset \*/

Elf64\_Addr p\_vaddr; /\* Segment virtual address \*/

Elf64\_Addr p\_paddr; /\* Segment physical address \*/

Elf64\_Xword p\_filesz; /\* Segment size in file \*/

Elf64\_Xword p\_memsz; /\* Segment size in memory \*/

Elf64\_Xword p\_align; /\* Segment alignment \*/

} Elf64\_Phdr;

typedef uint16\_t Elf32\_Half;

typedef uint16\_t Elf64\_Half;

typedef uint16\_t Elf32\_Section;

typedef uint16\_t Elf64\_Section;

typedef struct {

Elf32\_Word n\_namesz;

Elf32\_Word n\_descsz;

Elf32\_Word n\_type;

} Elf32\_Nhdr;

typedef struct {

Elf64\_Word n\_namesz;

Elf64\_Word n\_descsz;

Elf64\_Word n\_type;

} Elf64\_Nhdr;

typedef struct {

Elf64\_Word st\_name;

unsigned char st\_info;

unsigned char st\_other;

Elf64\_Section st\_shndx;

Elf64\_Addr st\_value;

Elf64\_Xword st\_size;

} Elf64\_Sym;

typedef struct {

Elf32\_Word st\_name;

Elf32\_Addr st\_value;

Elf32\_Word st\_size;

unsigned char st\_info;

unsigned char st\_other;

Elf32\_Section st\_shndx;

} Elf32\_Sym;

typedef struct {

Elf64\_Addr r\_offset;

Elf64\_Xword r\_info;

} Elf64\_Rel;

typedef struct {

Elf32\_Addr r\_offset;

Elf32\_Word r\_info;

} Elf32\_Rel;

typedef struct {

Elf64\_Addr r\_offset;

Elf64\_Xword r\_info;

Elf64\_Sxword r\_addend;

} Elf64\_Rela;

typedef struct {

Elf32\_Addr r\_offset;

Elf32\_Word r\_info;

Elf32\_Sword r\_addend;

} Elf32\_Rela;

typedef struct {

Elf32\_Half vd\_version;

Elf32\_Half vd\_flags;

Elf32\_Half vd\_ndx;

Elf32\_Half vd\_cnt;

Elf32\_Word vd\_hash;

Elf32\_Word vd\_aux;

Elf32\_Word vd\_next;

} Elf32\_Verdef;

typedef struct {

Elf64\_Half vd\_version;

Elf64\_Half vd\_flags;

Elf64\_Half vd\_ndx;

Elf64\_Half vd\_cnt;

Elf64\_Word vd\_hash;

Elf64\_Word vd\_aux;

Elf64\_Word vd\_next;

} Elf64\_Verdef;

typedef struct {

Elf64\_Word vda\_name;

Elf64\_Word vda\_next;

} Elf64\_Verdaux;

typedef struct {

Elf32\_Word vda\_name;

Elf32\_Word vda\_next;

} Elf32\_Verdaux;

typedef struct {

Elf32\_Half vn\_version;

Elf32\_Half vn\_cnt;

Elf32\_Word vn\_file;

Elf32\_Word vn\_aux;

Elf32\_Word vn\_next;

} Elf32\_Verneed;

typedef struct {

Elf64\_Half vn\_version;

Elf64\_Half vn\_cnt;

Elf64\_Word vn\_file;

Elf64\_Word vn\_aux;

Elf64\_Word vn\_next;

} Elf64\_Verneed;

typedef struct {

Elf32\_Word vna\_hash;

Elf32\_Half vna\_flags;

Elf32\_Half vna\_other;

Elf32\_Word vna\_name;

Elf32\_Word vna\_next;

} Elf32\_Vernaux;

typedef struct {

Elf64\_Word vna\_hash;

Elf64\_Half vna\_flags;

Elf64\_Half vna\_other;

Elf64\_Word vna\_name;

Elf64\_Word vna\_next;

} Elf64\_Vernaux;

typedef struct {

unsigned char e\_ident[EI\_NIDENT];

Elf64\_Half e\_type;

Elf64\_Half e\_machine;

Elf64\_Word e\_version;

Elf64\_Addr e\_entry;

Elf64\_Off e\_phoff;

Elf64\_Off e\_shoff;

Elf64\_Word e\_flags;

Elf64\_Half e\_ehsize;

Elf64\_Half e\_phentsize;

Elf64\_Half e\_phnum;

Elf64\_Half e\_shentsize;

Elf64\_Half e\_shnum;

Elf64\_Half e\_shstrndx;

} Elf64\_Ehdr;

typedef struct {

unsigned char e\_ident[EI\_NIDENT];

Elf32\_Half e\_type;

Elf32\_Half e\_machine;

Elf32\_Word e\_version;

Elf32\_Addr e\_entry;

Elf32\_Off e\_phoff;

Elf32\_Off e\_shoff;

Elf32\_Word e\_flags;

Elf32\_Half e\_ehsize;

Elf32\_Half e\_phentsize;

Elf32\_Half e\_phnum;

Elf32\_Half e\_shentsize;

Elf32\_Half e\_shnum;

Elf32\_Half e\_shstrndx;

} Elf32\_Ehdr;

typedef struct {

Elf32\_Word sh\_name;

Elf32\_Word sh\_type;

Elf32\_Word sh\_flags;

Elf32\_Addr sh\_addr;

Elf32\_Off sh\_offset;

Elf32\_Word sh\_size;

Elf32\_Word sh\_link;

Elf32\_Word sh\_info;

Elf32\_Word sh\_addralign;

Elf32\_Word sh\_entsize;

} Elf32\_Shdr;

typedef struct {

Elf64\_Word sh\_name;

Elf64\_Word sh\_type;

Elf64\_Xword sh\_flags;

Elf64\_Addr sh\_addr;

Elf64\_Off sh\_offset;

Elf64\_Xword sh\_size;

Elf64\_Word sh\_link;

Elf64\_Word sh\_info;

Elf64\_Xword sh\_addralign;

Elf64\_Xword sh\_entsize;

} Elf64\_Shdr;

typedef struct {

Elf32\_Sword d\_tag;

union {

Elf32\_Word d\_val;

Elf32\_Addr d\_ptr;

} d\_un;

} Elf32\_Dyn;

typedef struct {

Elf64\_Sxword d\_tag;

union {

Elf64\_Xword d\_val;

Elf64\_Addr d\_ptr;

} d\_un;

} Elf64\_Dyn;

### **14.4.8 endian.h**

#define \_\_LITTLE\_ENDIAN 1234

#define \_\_BIG\_ENDIAN 4321

#define BIG\_ENDIAN \_\_BIG\_ENDIAN

#define BYTE\_ORDER \_\_BYTE\_ORDER

#define LITTLE\_ENDIAN \_\_LITTLE\_ENDIAN

### **14.4.9 envz.h**

extern error\_t envz\_add(char \*\*envz, size\_t \* envz\_len, const char \*name,

const char \*value);

extern char envz\_entry(const char \*envz, size\_t envz\_len,

const char \*name);

extern char envz\_get(const char \*envz, size\_t envz\_len, const char \*name);

extern error\_t envz\_merge(char \*\*envz, size\_t \* envz\_len,

const char \*envz2, size\_t envz2\_len,

int override);

extern void envz\_remove(char \*\*envz, size\_t \* envz\_len, const char \*name);

extern void envz\_strip(char \*\*envz, size\_t \* envz\_len);

### **14.4.10 err.h**

extern void err(int eval, const char \*fmt, ...);

extern void errx(int eval, const char \*fmt, ...);

extern void verrx(int eval, const char \*fmt, va\_list args);

extern void warn(const char \*fmt, ...);

extern void warnx(const char \*fmt, ...);

### **14.4.11 errno.h**

#define errno (\*\_\_errno\_location())

#define EPERM 1 /\* Operation not permitted \*/

#define ECHILD 10 /\* No child processes \*/

#define ENETDOWN 100 /\* Network is down \*/

#define ENETUNREACH 101 /\* Network is unreachable \*/

#define ENETRESET 102 /\* Network dropped connection because of reset \*/

#define ECONNABORTED 103 /\* Software caused connection abort \*/

#define ECONNRESET 104 /\* Connection reset by peer \*/

#define ENOBUFS 105 /\* No buffer space available \*/

#define EISCONN 106 /\* Transport endpoint is already connected \*/

#define ENOTCONN 107 /\* Transport endpoint is not connected \*/

#define ESHUTDOWN 108 /\* Cannot send after transport endpoint shutdown \*/

#define ETOOMANYREFS 109 /\* Too many references: cannot splice \*/

#define EAGAIN 11 /\* Try again \*/

#define ETIMEDOUT 110 /\* Connection timed out \*/

#define ECONNREFUSED 111 /\* Connection refused \*/

#define EHOSTDOWN 112 /\* Host is down \*/

#define EHOSTUNREACH 113 /\* No route to host \*/

#define EALREADY 114 /\* Operation already in progress \*/

#define EINPROGRESS 115 /\* Operation now in progress \*/

#define ESTALE 116 /\* Stale NFS file handle \*/

#define EUCLEAN 117 /\* Structure needs cleaning \*/

#define ENOTNAM 118 /\* Not a XENIX named type file \*/

#define ENAVAIL 119 /\* No XENIX semaphores available \*/

#define ENOMEM 12 /\* Out of memory \*/

#define EISNAM 120 /\* Is a named type file \*/

#define EREMOTEIO 121 /\* Remote I/O error \*/

#define EDQUOT 122 /\* Quota exceeded \*/

#define ENOMEDIUM 123 /\* No medium found \*/

#define EMEDIUMTYPE 124 /\* Wrong medium type \*/

#define ECANCELED 125 /\* Operation Canceled \*/

#define EACCES 13 /\* Permission denied \*/

#define EOWNERDEAD 130 /\* Owner died \*/

#define ENOTRECOVERABLE 131 /\* State not recoverable \*/

#define ERFKILL 132 /\* Operation not possible due to RF-kill \*/

#define EFAULT 14 /\* Bad address \*/

#define ENOTBLK 15 /\* Block device required \*/

#define EBUSY 16 /\* Device or resource busy \*/

#define EEXIST 17 /\* File exists \*/

#define EXDEV 18 /\* Cross-device link \*/

#define ENODEV 19 /\* No such device \*/

#define ENOENT 2 /\* No such file or directory \*/

#define ENOTDIR 20 /\* Not a directory \*/

#define EISDIR 21 /\* Is a directory \*/

#define EINVAL 22 /\* Invalid argument \*/

#define ENFILE 23 /\* File table overflow \*/

#define EMFILE 24 /\* Too many open files \*/

#define ENOTTY 25 /\* Not a typewriter \*/

#define ETXTBSY 26 /\* Text file busy \*/

#define EFBIG 27 /\* File too large \*/

#define ENOSPC 28 /\* No space left on device \*/

#define ESPIPE 29 /\* Illegal seek \*/

#define ESRCH 3 /\* No such process \*/

#define EROFS 30 /\* Read-only file system \*/

#define EMLINK 31 /\* Too many links \*/

#define EPIPE 32 /\* Broken pipe \*/

#define EDOM 33 /\* Math argument out of domain of func \*/

#define ERANGE 34 /\* Math result not representable \*/

#define EDEADLK 35 /\* Resource deadlock would occur \*/

#define ENAMETOOLONG 36 /\* File name too long \*/

#define ENOLCK 37 /\* No record locks available \*/

#define ENOSYS 38 /\* Function not implemented \*/

#define ENOTEMPTY 39 /\* Directory not empty \*/

#define EINTR 4 /\* Interrupted system call \*/

#define ELOOP 40 /\* Too many symbolic links encountered \*/

#define ENOMSG 42 /\* No message of desired type \*/

#define EIDRM 43 /\* Identifier removed \*/

#define ECHRNG 44 /\* Channel number out of range \*/

#define EL2NSYNC 45 /\* Level 2 not synchronized \*/

#define EL3HLT 46 /\* Level 3 halted \*/

#define EL3RST 47 /\* Level 3 reset \*/

#define ELNRNG 48 /\* Link number out of range \*/

#define EUNATCH 49 /\* Protocol driver not attached \*/

#define EIO 5 /\* I/O error \*/

#define ENOANO 55 /\* No anode \*/

#define EBADRQC 56 /\* Invalid request code \*/

#define EBADSLT 57 /\* Invalid slot \*/

#define EBFONT 59 /\* Bad font file format \*/

#define ENXIO 6 /\* No such device or address \*/

#define ENOSTR 60 /\* Device not a stream \*/

#define ENODATA 61 /\* No data available \*/

#define ETIME 62 /\* Timer expired \*/

#define ENOSR 63 /\* Out of streams resources \*/

#define ENONET 64 /\* Machine is not on the network \*/

#define ENOPKG 65 /\* Package not installed \*/

#define EREMOTE 66 /\* Object is remote \*/

#define ENOLINK 67 /\* Link has been severed \*/

#define EADV 68 /\* Advertise error \*/

#define ESRMNT 69 /\* Srmount error \*/

#define E2BIG 7 /\* Argument list too long \*/

#define ECOMM 70 /\* Communication error on send \*/

#define EPROTO 71 /\* Protocol error \*/

#define EMULTIHOP 72 /\* Multihop attempted \*/

#define EDOTDOT 73 /\* RFS specific error \*/

#define EBADMSG 74 /\* Not a data message \*/

#define EOVERFLOW 75 /\* Value too large for defined data type \*/

#define ENOTUNIQ 76 /\* Name not unique on network \*/

#define EBADFD 77 /\* File descriptor in bad state \*/

#define EREMCHG 78 /\* Remote address changed \*/

#define ELIBACC 79 /\* Can not access a needed shared library \*/

#define ENOEXEC 8 /\* Exec format error \*/

#define ELIBBAD 80 /\* Accessing a corrupted shared library \*/

#define ELIBSCN 81 /\* .lib section in a.out corrupted \*/

#define ELIBMAX 82 /\* Attempting to link in too many shared libraries \*/

#define ELIBEXEC 83 /\* Cannot exec a shared library directly \*/

#define EILSEQ 84 /\* Illegal byte sequence \*/

#define ERESTART 85 /\* Interrupted system call should be restarted \*/

#define ESTRPIPE 86 /\* Streams pipe error \*/

#define EUSERS 87 /\* Too many users \*/

#define ENOTSOCK 88 /\* Socket operation on non-socket \*/

#define EDESTADDRREQ 89 /\* Destination address required \*/

#define EBADF 9 /\* Bad file number \*/

#define EMSGSIZE 90 /\* Message too long \*/

#define EPROTOTYPE 91 /\* Protocol wrong type for socket \*/

#define ENOPROTOOPT 92 /\* Protocol not available \*/

#define EPROTONOSUPPORT 93 /\* Protocol not supported \*/

#define ESOCKTNOSUPPORT 94 /\* Socket type not supported \*/

#define EOPNOTSUPP 95 /\* Operation not supported on transport endpoint \*/

#define EPFNOSUPPORT 96 /\* Protocol family not supported \*/

#define EAFNOSUPPORT 97 /\* Address family not supported by protocol \*/

#define EADDRINUSE 98 /\* Address already in use \*/

#define EADDRNOTAVAIL 99 /\* Cannot assign requested address \*/

#define EWOULDBLOCK EAGAIN /\* Operation would block \*/

#define ENOTSUP EOPNOTSUPP

extern int \*\_\_errno\_location(void);

### **14.4.12 error.h**

extern void error(int status, int errnum, const char \*format, ...);

### **14.4.13 execinfo.h**

extern int backtrace(void \*\*\_\_array, int \_\_size);

extern char \*\*backtrace\_symbols(void \*const \*\_\_array, int \_\_size);

extern void backtrace\_symbols\_fd(void \*const \*\_\_array, int \_\_size,

int \_\_fd);

### **14.4.14 fcntl.h**

#define AT\_FDCWD -100 /\* Use the current working directory to determine the target of relative file paths. \*/

#define POSIX\_FADV\_NORMAL 0

#define O\_RDONLY 00

#define O\_ACCMODE 0003

#define O\_WRONLY 01

#define O\_CREAT 0100

#define O\_TRUNC 01000

#define O\_DSYNC 010000

#define O\_RDWR 02

#define O\_EXCL 0200

#define O\_APPEND 02000

#define O\_ASYNC 020000

#define O\_CLOEXEC 02000000 /\* The FD\_CLOEXEC flag associated with the new descriptor shall be set to close the file descriptor upon execution of an exec family function. \*/

#define O\_NOCTTY 0400

#define O\_NDELAY 04000

#define O\_NONBLOCK 04000

#define O\_SYNC 04010000

#define AT\_SYMLINK\_NOFOLLOW 0x100 /\* Do not follow symbolic links. \*/

#define AT\_EMPTY\_PATH 0x1000 /\* Allow empty relative pathname. \*/

#define AT\_EACCESS 0x200 /\* Check access using effective user and group ID. \*/

#define AT\_REMOVEDIR 0x200 /\* Remove directory instead of file. \*/

#define AT\_SYMLINK\_FOLLOW 0x400 /\* Follow symbolic link. \*/

#define AT\_NO\_AUTOMOUNT 0x800 /\* Suppress terminal automount traversal. \*/

#define FD\_CLOEXEC 1

#define POSIX\_FADV\_RANDOM 1

#define F\_DUPFD\_CLOEXEC 1030 /\* Duplicate file descriptor with the close-on-exec flag FD\_CLOEXEC set. \*/

#define POSIX\_FADV\_SEQUENTIAL 2

#define POSIX\_FADV\_WILLNEED 3

#define O\_RSYNC O\_SYNC

struct flock {

short l\_type;

short l\_whence;

off\_t l\_start;

off\_t l\_len;

pid\_t l\_pid;

};

struct flock64 {

short l\_type;

short l\_whence;

loff\_t l\_start;

loff\_t l\_len;

pid\_t l\_pid;

};

#define AT\_FDCWD -100

#define AT\_SYMLINK\_NOFOLLOW 0x100

#define AT\_EACCESS 0x200

#define AT\_REMOVEDIR 0x200

#define AT\_SYMLINK\_FOLLOW 0x400

#define F\_DUPFD 0

#define F\_RDLCK 0

#ifndef SEEK\_SET

#define SEEK\_SET 0

#endif

#define F\_GETFD 1

#define F\_WRLCK 1

#ifndef SEEK\_CUR

#define SEEK\_CUR 1

#endif

#define F\_SETSIG 10

#define F\_GETSIG 11

#define F\_SETFD 2

#define F\_UNLCK 2

#ifndef SEEK\_END

#define SEEK\_END 2

#endif

#define F\_GETFL 3

#define F\_SETFL 4

#define F\_GETLK 5

#define F\_SETLK 6

#define F\_SETLKW 7

#define F\_SETOWN 8

#define F\_GETOWN 9

extern int creat(const char \*\_\_file, mode\_t \_\_mode);

extern int creat64(const char \*\_\_file, mode\_t \_\_mode);

extern int fcntl(int \_\_fd, int \_\_cmd, ...);

extern int open(const char \*\_\_file, int \_\_oflag, ...);

extern int open64(const char \*\_\_file, int \_\_oflag, ...);

extern int openat(int \_\_fd, const char \*\_\_file, int \_\_oflag, ...);

extern int openat64(int \_\_fd, const char \*\_\_file, int \_\_oflag, ...);

extern int posix\_fadvise(int \_\_fd, off\_t \_\_offset, off\_t \_\_len,

int \_\_advise);

extern int posix\_fadvise64(int \_\_fd, off64\_t \_\_offset, off64\_t \_\_len,

int \_\_advise);

extern int posix\_fallocate(int \_\_fd, off\_t \_\_offset, off\_t \_\_len);

extern int posix\_fallocate64(int \_\_fd, off64\_t \_\_offset, off64\_t \_\_len);

### **14.4.15 fmtmsg.h**

#define MM\_HARD 1 /\* Source of the condition is hardware. \*/

#define MM\_NRECOV 128 /\* Non-recoverable error. \*/

#define MM\_UTIL 16 /\* Condition detected by utility. \*/

#define MM\_SOFT 2 /\* Source of the condition is software. \*/

#define MM\_PRINT 256 /\* Display message in standard error. \*/

#define MM\_OPSYS 32 /\* Condition detected by operating system. \*/

#define MM\_FIRM 4 /\* Source of the condition is firmware. \*/

#define MM\_CONSOLE 512 /\* Display message on system console. \*/

#define MM\_RECOVER 64 /\* Recoverable error. \*/

#define MM\_APPL 8 /\* Condition detected by application. \*/

#define MM\_NOSEV 0 /\* No severity level provided for the message. \*/

#define MM\_HALT 1 /\* Error causing application to halt. \*/

#define MM\_ERROR 2 /\* Application has encountered a non-fatal fault. \*/

#define MM\_WARNING 3 /\* Application has detected unusual non-error condition. \*/

#define MM\_INFO 4 /\* Informative message. \*/

#define MM\_NULLACT ((char \*) 0)

#define MM\_NULLLBL ((char \*) 0)

#define MM\_NULLTAG ((char \*) 0)

#define MM\_NULLTXT ((char \*) 0)

#define MM\_NULLMC ((long int) 0)

#define MM\_NULLSEV 0

#define MM\_NOTOK -1 /\* The function failed completely. \*/

#define MM\_OK 0 /\* The function succeeded. \*/

#define MM\_NOMSG 1 /\* The function was unable to generate a message on standard error, but otherwise succeeded. \*/

#define MM\_NOCON 4 /\* The function was unable to generate a console message, but otherwise succeeded. \*/

extern int fmtmsg(long int \_\_classification, const char \*\_\_label,

int \_\_severity, const char \*\_\_text, const char \*\_\_action,

const char \*\_\_tag);

### **14.4.16 fnmatch.h**

#define FNM\_CASEFOLD (1<<4)

#define FNM\_FILE\_NAME FNM\_PATHNAME

#define FNM\_PATHNAME (1<<0)

#define FNM\_NOESCAPE (1<<1)

#define FNM\_PERIOD (1<<2)

#define FNM\_NOMATCH 1

extern int fnmatch(const char \*\_\_pattern, const char \*\_\_name, int \_\_flags);

### **14.4.17 ftw.h**

#define FTW\_D FTW\_D

#define FTW\_DNR FTW\_DNR

#define FTW\_DP FTW\_DP

#define FTW\_F FTW\_F

#define FTW\_NS FTW\_NS

#define FTW\_SL FTW\_SL

#define FTW\_SLN FTW\_SLN

enum {

FTW\_F,

FTW\_D,

FTW\_DNR,

FTW\_NS,

FTW\_SL,

FTW\_DP,

FTW\_SLN

};

enum {

FTW\_PHYS = 1,

FTW\_MOUNT = 2,

FTW\_CHDIR = 4,

FTW\_DEPTH = 8

};

struct FTW {

int base;

int level;

};

typedef int (\*\_\_ftw\_func\_t) (const char \*\_\_filename,

const struct stat \* \_\_status, int \_\_flag);

typedef int (\*\_\_ftw64\_func\_t) (const char \*\_\_filename,

const struct stat64 \* \_\_status, int \_\_flag);

typedef int (\*\_\_nftw\_func\_t) (const char \*\_\_filename,

const struct stat \* \_\_status, int \_\_flag,

struct FTW \* \_\_info);

typedef int (\*\_\_nftw64\_func\_t) (const char \*\_\_filename,

const struct stat64 \* \_\_status, int \_\_flag,

struct FTW \* \_\_info);

extern int ftw(const char \*\_\_dir, \_\_ftw\_func\_t \_\_func, int \_\_descriptors);

extern int ftw64(const char \*\_\_dir, \_\_ftw64\_func\_t \_\_func,

int \_\_descriptors);

extern int nftw(const char \*\_\_dir, \_\_nftw\_func\_t \_\_func, int \_\_descriptors,

int \_\_flag);

extern int nftw64(const char \*\_\_dir, \_\_nftw64\_func\_t \_\_func,

int \_\_descriptors, int \_\_flag);

### **14.4.18 getopt.h**

#define no\_argument 0

#define required\_argument 1

#define optional\_argument 2

struct option {

const char \*name;

int has\_arg;

int \*flag;

int val;

};

extern int getopt\_long(int \_\_\_argc, char \*const \_\_\_argv[],

const char \*\_\_shortopts,

const struct option \*\_\_longopts, int \*\_\_longind);

extern int getopt\_long\_only(int \_\_\_argc, char \*const \_\_\_argv[],

const char \*\_\_shortopts,

const struct option \*\_\_longopts,

int \*\_\_longind);

### **14.4.19 glob.h**

#define GLOB\_ERR (1<<0)

#define GLOB\_MARK (1<<1)

#define GLOB\_BRACE (1<<10)

#define GLOB\_NOMAGIC (1<<11)

#define GLOB\_TILDE (1<<12)

#define GLOB\_ONLYDIR (1<<13)

#define GLOB\_TILDE\_CHECK (1<<14)

#define GLOB\_NOSORT (1<<2)

#define GLOB\_DOOFFS (1<<3)

#define GLOB\_NOCHECK (1<<4)

#define GLOB\_APPEND (1<<5)

#define GLOB\_NOESCAPE (1<<6)

#define GLOB\_PERIOD (1<<7)

#define GLOB\_MAGCHAR (1<<8)

#define GLOB\_ALTDIRFUNC (1<<9)

#define GLOB\_NOSPACE 1

#define GLOB\_ABORTED 2

#define GLOB\_NOMATCH 3

#define GLOB\_NOSYS 4

typedef struct {

size\_t gl\_pathc;

char \*\*gl\_pathv;

size\_t gl\_offs;

int gl\_flags;

void (\*gl\_closedir) (void \*);

struct dirent \*(\*gl\_readdir) (void \*);

void \*(\*gl\_opendir) (const char \*);

int (\*gl\_lstat) (const char \*, struct stat \*);

int (\*gl\_stat) (const char \*, struct stat \*);

} glob\_t;

typedef struct {

size\_t gl\_pathc;

char \*\*gl\_pathv;

size\_t gl\_offs;

int gl\_flags;

void (\*gl\_closedir) (void \*);

struct dirent64 \*(\*gl\_readdir) (void \*);

void \*(\*gl\_opendir) (const char \*);

int (\*gl\_lstat) (const char \*, struct stat \*);

int (\*gl\_stat) (const char \*, struct stat \*);

} glob64\_t;

extern int glob(const char \*\_\_pattern, int \_\_flags,

int (\*\_\_errfunc) (const char \*, int), glob\_t \* \_\_pglob);

extern int glob64(const char \*\_\_pattern, int \_\_flags,

int (\*\_\_errfunc) (const char \*, int),

glob64\_t \* \_\_pglob);

extern void globfree(glob\_t \* \_\_pglob);

extern void globfree64(glob64\_t \* \_\_pglob);

### **14.4.20 gnu/libc-version.h**

extern const char \*gnu\_get\_libc\_release(void);

extern const char \*gnu\_get\_libc\_version(void);

### **14.4.21 grp.h**

struct group {

char \*gr\_name;

char \*gr\_passwd;

gid\_t gr\_gid;

char \*\*gr\_mem;

};

extern void endgrent(void);

extern struct group \*getgrent(void);

extern int getgrent\_r(struct group \*\_\_resultbuf, char \*\_\_buffer,

size\_t \_\_buflen, struct group \*\*\_\_result);

extern struct group \*getgrgid(gid\_t \_\_gid);

extern int getgrgid\_r(gid\_t \_\_gid, struct group \*\_\_resultbuf,

char \*\_\_buffer, size\_t \_\_buflen,

struct group \*\*\_\_result);

extern struct group \*getgrnam(const char \*\_\_name);

extern int getgrnam\_r(const char \*\_\_name, struct group \*\_\_resultbuf,

char \*\_\_buffer, size\_t \_\_buflen,

struct group \*\*\_\_result);

extern int getgrouplist(const char \*\_\_user, gid\_t \_\_group,

gid\_t \* \_\_groups, int \*\_\_ngroups);

extern int initgroups(const char \*\_\_user, gid\_t \_\_group);

extern void setgrent(void);

extern int setgroups(size\_t \_\_n, const gid\_t \* \_\_groups);

### **14.4.22 iconv.h**

typedef void \*iconv\_t;

extern size\_t iconv(iconv\_t \_\_cd, char \*\*\_\_inbuf, size\_t \* \_\_inbytesleft,

char \*\*\_\_outbuf, size\_t \* \_\_outbytesleft);

extern int iconv\_close(iconv\_t \_\_cd);

extern iconv\_t iconv\_open(const char \*\_\_tocode, const char \*\_\_fromcode);

### **14.4.23 ifaddrs.h**

#define ifa\_broadaddr ifa\_ifu.ifu\_broadaddr

#define ifa\_dstaddr ifa\_ifu.ifu\_dstaddr

struct ifaddrs {

struct ifaddrs \*ifa\_next;

char \*ifa\_name;

unsigned int ifa\_flags;

struct sockaddr \*ifa\_addr;

struct sockaddr \*ifa\_netmask;

union {

struct sockaddr \*ifu\_broadaddr;

struct sockaddr \*ifu\_dstaddr;

} ifa\_ifu;

void \*ifa\_data;

};

extern void freeifaddrs(struct ifaddrs \*);

extern int getifaddrs(struct ifaddrs \*\*);

### **14.4.24 inttypes.h**

#if !defined \_\_cplusplus || defined \_\_STDC\_FORMAT\_MACROS

#define PRId16 "d"

#define PRId32 "d"

#define PRId8 "d"

#define PRIdFAST8 "d"

#define PRIdLEAST16 "d"

#define PRIdLEAST32 "d"

#define PRIdLEAST8 "d"

#define SCNd32 "d"

#define SCNdLEAST32 "d"

#define SCNd16 "hd"

#define SCNdLEAST16 "hd"

#define SCNd8 "hhd"

#define SCNdFAST8 "hhd"

#define SCNdLEAST8 "hhd"

#define SCNi8 "hhi"

#define SCNiFAST8 "hhi"

#define SCNiLEAST8 "hhi"

#define SCNo8 "hho"

#define SCNoFAST8 "hho"

#define SCNoLEAST8 "hho"

#define SCNu8 "hhu"

#define SCNuFAST8 "hhu"

#define SCNuLEAST8 "hhu"

#define SCNx8 "hhx"

#define SCNxFAST8 "hhx"

#define SCNxLEAST8 "hhx"

#define SCNi16 "hi"

#define SCNiLEAST16 "hi"

#define SCNo16 "ho"

#define SCNoLEAST16 "ho"

#define SCNu16 "hu"

#define SCNuLEAST16 "hu"

#define SCNx16 "hx"

#define SCNxLEAST16 "hx"

#define PRIi16 "i"

#define PRIi32 "i"

#define PRIi8 "i"

#define PRIiFAST8 "i"

#define PRIiLEAST16 "i"

#define PRIiLEAST32 "i"

#define PRIiLEAST8 "i"

#define SCNi32 "i"

#define SCNiLEAST32 "i"

#define PRIo16 "o"

#define PRIo32 "o"

#define PRIo8 "o"

#define PRIoFAST8 "o"

#define PRIoLEAST16 "o"

#define PRIoLEAST32 "o"

#define PRIoLEAST8 "o"

#define SCNo32 "o"

#define SCNoLEAST32 "o"

#define PRIu16 "u"

#define PRIu32 "u"

#define PRIu8 "u"

#define PRIuFAST8 "u"

#define PRIuLEAST16 "u"

#define PRIuLEAST32 "u"

#define PRIuLEAST8 "u"

#define SCNu32 "u"

#define SCNuLEAST32 "u"

#define PRIX16 "X"

#define PRIX32 "X"

#define PRIX8 "X"

#define PRIXFAST8 "X"

#define PRIXLEAST16 "X"

#define PRIXLEAST32 "X"

#define PRIXLEAST8 "X"

#define PRIx16 "x"

#define PRIx32 "x"

#define PRIx8 "x"

#define PRIxFAST8 "x"

#define PRIxLEAST16 "x"

#define PRIxLEAST32 "x"

#define PRIxLEAST8 "x"

#define SCNx32 "x"

#define SCNxLEAST32 "x"

#define PRId64 \_\_PRI64\_PREFIX"d"

#define PRIdFAST64 \_\_PRI64\_PREFIX"d"

#define PRIdLEAST64 \_\_PRI64\_PREFIX"d"

#define PRIdMAX \_\_PRI64\_PREFIX"d"

#define SCNd64 \_\_PRI64\_PREFIX"d"

#define SCNdFAST64 \_\_PRI64\_PREFIX"d"

#define SCNdLEAST64 \_\_PRI64\_PREFIX"d"

#define SCNdMAX \_\_PRI64\_PREFIX"d"

#define PRIi64 \_\_PRI64\_PREFIX"i"

#define PRIiFAST64 \_\_PRI64\_PREFIX"i"

#define PRIiLEAST64 \_\_PRI64\_PREFIX"i"

#define PRIiMAX \_\_PRI64\_PREFIX"i"

#define SCNi64 \_\_PRI64\_PREFIX"i"

#define SCNiFAST64 \_\_PRI64\_PREFIX"i"

#define SCNiLEAST64 \_\_PRI64\_PREFIX"i"

#define SCNiMAX \_\_PRI64\_PREFIX"i"

#define PRIo64 \_\_PRI64\_PREFIX"o"

#define PRIoFAST64 \_\_PRI64\_PREFIX"o"

#define PRIoLEAST64 \_\_PRI64\_PREFIX"o"

#define PRIoMAX \_\_PRI64\_PREFIX"o"

#define SCNo64 \_\_PRI64\_PREFIX"o"

#define SCNoFAST64 \_\_PRI64\_PREFIX"o"

#define SCNoLEAST64 \_\_PRI64\_PREFIX"o"

#define SCNoMAX \_\_PRI64\_PREFIX"o"

#define PRIu64 \_\_PRI64\_PREFIX"u"

#define PRIuFAST64 \_\_PRI64\_PREFIX"u"

#define PRIuLEAST64 \_\_PRI64\_PREFIX"u"

#define PRIuMAX \_\_PRI64\_PREFIX"u"

#define SCNu64 \_\_PRI64\_PREFIX"u"

#define SCNuFAST64 \_\_PRI64\_PREFIX"u"

#define SCNuLEAST64 \_\_PRI64\_PREFIX"u"

#define SCNuMAX \_\_PRI64\_PREFIX"u"

#define PRIX64 \_\_PRI64\_PREFIX"X"

#define PRIXFAST64 \_\_PRI64\_PREFIX"X"

#define PRIXLEAST64 \_\_PRI64\_PREFIX"X"

#define PRIXMAX \_\_PRI64\_PREFIX"X"

#define PRIx64 \_\_PRI64\_PREFIX"x"

#define PRIxFAST64 \_\_PRI64\_PREFIX"x"

#define PRIxLEAST64 \_\_PRI64\_PREFIX"x"

#define PRIxMAX \_\_PRI64\_PREFIX"x"

#define SCNx64 \_\_PRI64\_PREFIX"x"

#define SCNxFAST64 \_\_PRI64\_PREFIX"x"

#define SCNxLEAST64 \_\_PRI64\_PREFIX"x"

#define SCNxMAX \_\_PRI64\_PREFIX"x"

#define PRIdFAST16 \_\_PRIPTR\_PREFIX"d"

#define PRIdFAST32 \_\_PRIPTR\_PREFIX"d"

#define PRIdPTR \_\_PRIPTR\_PREFIX"d"

#define SCNdFAST16 \_\_PRIPTR\_PREFIX"d"

#define SCNdFAST32 \_\_PRIPTR\_PREFIX"d"

#define SCNdPTR \_\_PRIPTR\_PREFIX"d"

#define PRIiFAST16 \_\_PRIPTR\_PREFIX"i"

#define PRIiFAST32 \_\_PRIPTR\_PREFIX"i"

#define PRIiPTR \_\_PRIPTR\_PREFIX"i"

#define SCNiFAST16 \_\_PRIPTR\_PREFIX"i"

#define SCNiFAST32 \_\_PRIPTR\_PREFIX"i"

#define SCNiPTR \_\_PRIPTR\_PREFIX"i"

#define PRIoFAST16 \_\_PRIPTR\_PREFIX"o"

#define PRIoFAST32 \_\_PRIPTR\_PREFIX"o"

#define PRIoPTR \_\_PRIPTR\_PREFIX"o"

#define SCNoFAST16 \_\_PRIPTR\_PREFIX"o"

#define SCNoFAST32 \_\_PRIPTR\_PREFIX"o"

#define SCNoPTR \_\_PRIPTR\_PREFIX"o"

#define PRIuFAST16 \_\_PRIPTR\_PREFIX"u"

#define PRIuFAST32 \_\_PRIPTR\_PREFIX"u"

#define PRIuPTR \_\_PRIPTR\_PREFIX"u"

#define SCNuFAST16 \_\_PRIPTR\_PREFIX"u"

#define SCNuFAST32 \_\_PRIPTR\_PREFIX"u"

#define SCNuPTR \_\_PRIPTR\_PREFIX"u"

#define PRIXFAST16 \_\_PRIPTR\_PREFIX"X"

#define PRIXFAST32 \_\_PRIPTR\_PREFIX"X"

#define PRIXPTR \_\_PRIPTR\_PREFIX"X"

#define PRIxFAST16 \_\_PRIPTR\_PREFIX"x"

#define PRIxFAST32 \_\_PRIPTR\_PREFIX"x"

#define PRIxPTR \_\_PRIPTR\_PREFIX"x"

#define SCNxFAST16 \_\_PRIPTR\_PREFIX"x"

#define SCNxFAST32 \_\_PRIPTR\_PREFIX"x"

#define SCNxPTR \_\_PRIPTR\_PREFIX"x"

#endif

#define \_\_PDP\_ENDIAN 3412

#define PDP\_ENDIAN \_\_PDP\_ENDIAN

extern intmax\_t imaxabs(intmax\_t \_\_n);

extern imaxdiv\_t imaxdiv(intmax\_t \_\_numer, intmax\_t \_\_denom);

extern intmax\_t strtoimax(const char \*\_\_nptr, char \*\*\_\_endptr, int \_\_base);

extern uintmax\_t strtoumax(const char \*\_\_nptr, char \*\*\_\_endptr,

int \_\_base);

extern intmax\_t wcstoimax(const wchar\_t \* \_\_nptr, wchar\_t \* \*\_\_endptr,

int \_\_base);

extern uintmax\_t wcstoumax(const wchar\_t \* \_\_nptr, wchar\_t \* \*\_\_endptr,

int \_\_base);

### **14.4.25 langinfo.h**

#define ABDAY\_1 0x20000 /\* Sun. \*/

#define ABDAY\_2 0x20001

#define ABDAY\_3 0x20002

#define ABDAY\_4 0x20003

#define ABDAY\_5 0x20004

#define ABDAY\_6 0x20005

#define ABDAY\_7 0x20006

#define DAY\_1 0x20007

#define DAY\_2 0x20008

#define DAY\_3 0x20009

#define DAY\_4 0x2000A

#define DAY\_5 0x2000B

#define DAY\_6 0x2000C

#define DAY\_7 0x2000D

#define ABMON\_1 0x2000E

#define ABMON\_2 0x2000F

#define ABMON\_3 0x20010

#define ABMON\_4 0x20011

#define ABMON\_5 0x20012

#define ABMON\_6 0x20013

#define ABMON\_7 0x20014

#define ABMON\_8 0x20015

#define ABMON\_9 0x20016

#define ABMON\_10 0x20017

#define ABMON\_11 0x20018

#define ABMON\_12 0x20019

#define MON\_1 0x2001A

#define MON\_2 0x2001B

#define MON\_3 0x2001C

#define MON\_4 0x2001D

#define MON\_5 0x2001E

#define MON\_6 0x2001F

#define MON\_7 0x20020

#define MON\_8 0x20021

#define MON\_9 0x20022

#define MON\_10 0x20023

#define MON\_11 0x20024

#define MON\_12 0x20025

#define AM\_STR 0x20026

#define PM\_STR 0x20027

#define D\_T\_FMT 0x20028

#define D\_FMT 0x20029

#define T\_FMT 0x2002A

#define T\_FMT\_AMPM 0x2002B

#define ERA 0x2002C

#define ERA\_D\_FMT 0x2002E

#define ALT\_DIGITS 0x2002F

#define ERA\_D\_T\_FMT 0x20030

#define ERA\_T\_FMT 0x20031

#define CODESET 14

#define CRNCYSTR 0x4000F

#define RADIXCHAR 0x10000

#define THOUSEP 0x10001

#define YESEXPR 0x50000

#define NOEXPR 0x50001

#define YESSTR 0x50002

#define NOSTR 0x50003

extern char \*nl\_langinfo(nl\_item \_\_item);

### **14.4.26 libgen.h**

#define basename \_\_xpg\_basename

extern char \*\_\_xpg\_basename(char \*\_\_path);

extern char \*dirname(char \*\_\_path);

### **14.4.27 libintl.h**

extern char \*bind\_textdomain\_codeset(const char \*\_\_domainname,

const char \*\_\_codeset);

extern char \*bindtextdomain(const char \*\_\_domainname,

const char \*\_\_dirname);

extern char \*dcgettext(const char \*\_\_domainname, const char \*\_\_msgid,

int \_\_category);

extern char \*dcngettext(const char \*\_\_domainname, const char \*\_\_msgid1,

const char \*\_\_msgid2, unsigned long int \_\_n,

int \_\_category);

extern char \*dgettext(const char \*\_\_domainname, const char \*\_\_msgid);

extern char \*dngettext(const char \*\_\_domainname, const char \*\_\_msgid1,

const char \*\_\_msgid2, unsigned long int \_\_n);

extern char \*gettext(const char \*\_\_msgid);

extern char \*ngettext(const char \*\_\_msgid1, const char \*\_\_msgid2,

unsigned long int \_\_n);

extern char \*textdomain(const char \*\_\_domainname);

### **14.4.28 limits.h**

#define LLONG\_MIN (-LLONG\_MAX-1LL)

#define \_POSIX\_AIO\_MAX 1

#define \_POSIX\_QLIMIT 1

#define \_POSIX2\_BC\_STRING\_MAX 1000

#define IOV\_MAX 1024

#define \_POSIX2\_CHARCLASS\_NAME\_MAX 14

#define \_POSIX\_NAME\_MAX 14

#define \_POSIX\_UIO\_MAXIOV 16

#define ULLONG\_MAX 18446744073709551615ULL

#define \_POSIX2\_COLL\_WEIGHTS\_MAX 2

#define \_POSIX\_AIO\_LISTIO\_MAX 2

#define \_POSIX\_OPEN\_MAX 20

#define \_POSIX\_CLOCKRES\_MIN 20000000

#define CHARCLASS\_NAME\_MAX 2048

#define LINE\_MAX 2048

#define \_POSIX2\_BC\_DIM\_MAX 2048

#define \_POSIX2\_LINE\_MAX 2048

#define \_POSIX\_CHILD\_MAX 25

#define COLL\_WEIGHTS\_MAX 255

#define NAME\_MAX 255

#define \_POSIX2\_RE\_DUP\_MAX 255

#define \_POSIX\_HOST\_NAME\_MAX 255

#define \_POSIX\_MAX\_CANON 255

#define \_POSIX\_MAX\_INPUT 255

#define \_POSIX\_RE\_DUP\_MAX 255

#define \_POSIX\_SYMLINK\_MAX 255

#define \_POSIX\_PATH\_MAX 256

#define \_POSIX\_SEM\_NSEMS\_MAX 256

#define NGROUPS\_MAX 32

#define WORD\_BIT 32

#define \_POSIX2\_EXPR\_NEST\_MAX 32

#define \_POSIX\_DELAYTIMER\_MAX 32

#define \_POSIX\_MQ\_PRIO\_MAX 32

#define \_POSIX\_SIGQUEUE\_MAX 32

#define \_POSIX\_TIMER\_MAX 32

#define \_POSIX\_SEM\_VALUE\_MAX 32767

#define \_POSIX\_SSIZE\_MAX 32767

#define PATH\_MAX 4096

#define \_POSIX\_ARG\_MAX 4096

#define \_POSIX\_PIPE\_BUF 512

#define \_POSIX\_TZNAME\_MAX 6

#define \_POSIX\_LINK\_MAX 8

#define \_POSIX\_MQ\_OPEN\_MAX 8

#define \_POSIX\_NGROUPS\_MAX 8

#define \_POSIX\_RTSIG\_MAX 8

#define \_POSIX\_STREAM\_MAX 8

#define \_POSIX\_SYMLOOP\_MAX 8

#define \_POSIX\_LOGIN\_NAME\_MAX 9

#define \_POSIX\_TTY\_NAME\_MAX 9

#define LLONG\_MAX 9223372036854775807LL

#define \_POSIX2\_BC\_BASE\_MAX 99

#define \_POSIX2\_BC\_SCALE\_MAX 99

#define NL\_MSGMAX INT\_MAX

#define NL\_SETMAX INT\_MAX

#define NL\_TEXTMAX INT\_MAX

#define SSIZE\_MAX LONG\_MAX /\* Maximum value of an object of type ssize\_t \*/

#define BC\_BASE\_MAX \_POSIX2\_BC\_BASE\_MAX

#define BC\_DIM\_MAX \_POSIX2\_BC\_DIM\_MAX

#define BC\_SCALE\_MAX \_POSIX2\_BC\_SCALE\_MAX

#define BC\_STRING\_MAX \_POSIX2\_BC\_STRING\_MAX

#define EXPR\_NEST\_MAX \_POSIX2\_EXPR\_NEST\_MAX

#define NL\_LANGMAX \_POSIX2\_LINE\_MAX

#define NL\_ARGMAX \_POSIX\_ARG\_MAX

#define \_POSIX\_FD\_SETSIZE \_POSIX\_OPEN\_MAX

#define \_POSIX\_HIWAT \_POSIX\_PIPE\_BUF

#define MB\_LEN\_MAX 16

#define SCHAR\_MIN (-128)

#define SCHAR\_MAX 127

#define UCHAR\_MAX 255

#define CHAR\_BIT 8

#define SHRT\_MIN (-32768)

#define SHRT\_MAX 32767

#define USHRT\_MAX 65535

#define INT\_MIN (-INT\_MAX-1)

#define INT\_MAX 2147483647

#define UINT\_MAX 4294967295U

#define LONG\_MIN (-LONG\_MAX-1L)

#define PTHREAD\_KEYS\_MAX 1024

#define PTHREAD\_THREADS\_MAX 16384

#define PTHREAD\_DESTRUCTOR\_ITERATIONS 4

### **14.4.29 link.h**

extern int

dl\_iterate\_phdr(int (\*callback) (struct dl\_phdr\_info \*, size\_t, void \*),

void \*data);

### **14.4.30 locale.h**

struct lconv {

char \*decimal\_point;

char \*thousands\_sep;

char \*grouping;

char \*int\_curr\_symbol;

char \*currency\_symbol;

char \*mon\_decimal\_point;

char \*mon\_thousands\_sep;

char \*mon\_grouping;

char \*positive\_sign;

char \*negative\_sign;

char int\_frac\_digits;

char frac\_digits;

char p\_cs\_precedes;

char p\_sep\_by\_space;

char n\_cs\_precedes;

char n\_sep\_by\_space;

char p\_sign\_posn;

char n\_sign\_posn;

char int\_p\_cs\_precedes;

char int\_p\_sep\_by\_space;

char int\_n\_cs\_precedes;

char int\_n\_sep\_by\_space;

char int\_p\_sign\_posn;

char int\_n\_sign\_posn;

};

#define LC\_GLOBAL\_LOCALE ((locale\_t) -1L)

#define LC\_CTYPE 0

#define LC\_NUMERIC 1

#define LC\_TELEPHONE 10

#define LC\_MEASUREMENT 11

#define LC\_IDENTIFICATION 12

#define LC\_TIME 2

#define LC\_COLLATE 3

#define LC\_MONETARY 4

#define LC\_MESSAGES 5

#define LC\_ALL 6

#define LC\_PAPER 7

#define LC\_NAME 8

#define LC\_ADDRESS 9

struct \_\_locale\_struct {

struct locale\_data \*\_\_locales[13];

const unsigned short \*\_\_ctype\_b;

const int \*\_\_ctype\_tolower;

const int \*\_\_ctype\_toupper;

const char \*\_\_names[13];

};

typedef struct \_\_locale\_struct \*\_\_locale\_t;

typedef struct \_\_locale\_struct \*locale\_t;

#define LC\_ADDRESS\_MASK (1 << LC\_ADDRESS)

#define LC\_COLLATE\_MASK (1 << LC\_COLLATE)

#define LC\_IDENTIFICATION\_MASK (1 << LC\_IDENTIFICATION)

#define LC\_MEASUREMENT\_MASK (1 << LC\_MEASUREMENT)

#define LC\_MESSAGES\_MASK (1 << LC\_MESSAGES)

#define LC\_MONETARY\_MASK (1 << LC\_MONETARY)

#define LC\_NAME\_MASK (1 << LC\_NAME)

#define LC\_NUMERIC\_MASK (1 << LC\_NUMERIC)

#define LC\_PAPER\_MASK (1 << LC\_PAPER)

#define LC\_TELEPHONE\_MASK (1 << LC\_TELEPHONE)

#define LC\_TIME\_MASK (1 << LC\_TIME)

#define LC\_CTYPE\_MASK (1<<LC\_CTYPE)

#define LC\_ALL\_MASK \

(LC\_CTYPE\_MASK| LC\_NUMERIC\_MASK| LC\_TIME\_MASK| LC\_COLLATE\_MASK| LC\_MONETARY\_MASK|\

LC\_MESSAGES\_MASK| LC\_PAPER\_MASK| LC\_NAME\_MASK| LC\_ADDRESS\_MASK| LC\_TELEPHONE\_MASK|\

LC\_MEASUREMENT\_MASK| LC\_IDENTIFICATION\_MASK)

extern locale\_t duplocale(locale\_t \_\_dataset);

extern void freelocale(locale\_t \_\_dataset);

extern struct lconv \*localeconv(void);

extern locale\_t newlocale(int \_\_category\_mask, const char \*\_\_locale,

locale\_t \_\_base);

extern char \*setlocale(int \_\_category, const char \*\_\_locale);

extern locale\_t uselocale(locale\_t \_\_dataset);

### **14.4.31 lsb/time.h**

struct timeval {

time\_t tv\_sec;

suseconds\_t tv\_usec;

};

### **14.4.32 lsb/types.h**

/\*

\* This header is architecture dependent

\* Please refer to the specific architecture specification for details

\*/

### **14.4.33 lsb/wchar.h**

typedef unsigned int wint\_t;

typedef struct {

int count;

wint\_t value;

} \_\_mbstate\_t;

typedef \_\_mbstate\_t mbstate\_t;

### **14.4.34 monetary.h**

extern ssize\_t strfmon(char \*\_\_s, size\_t \_\_maxsize, const char \*\_\_format,

...);

extern ssize\_t strfmon\_l(char \*s, size\_t maxsize, locale\_t locale,

const char \*format, ...);

### **14.4.35 net/if.h**

#define IF\_NAMESIZE 16

#define IFF\_UP 0x01 /\* Interface is up. \*/

#define IFF\_BROADCAST 0x02 /\* Broadcast address valid. \*/

#define IFF\_DEBUG 0x04 /\* Turn on debugging. \*/

#define IFF\_LOOPBACK 0x08 /\* Is a loopback net. \*/

#define IFF\_POINTOPOINT 0x10 /\* Interface is point-to-point link. \*/

#define IFF\_PROMISC 0x100 /\* Receive all packets. \*/

#define IFF\_MULTICAST 0x1000 /\* Supports multicast. \*/

#define IFF\_NOTRAILERS 0x20 /\* Avoid use of trailers. \*/

#define IFF\_RUNNING 0x40 /\* Resources allocated. \*/

#define IFF\_NOARP 0x80 /\* No address resolution protocol. \*/

struct if\_nameindex {

unsigned int if\_index; /\* 1, 2, ... \*/

char \*if\_name; /\* null terminated name: \*/

};

struct ifaddr {

struct sockaddr ifa\_addr; /\* Address of interface. \*/

union {

struct sockaddr ifu\_broadaddr;

struct sockaddr ifu\_dstaddr;

} ifa\_ifu;

void \*ifa\_ifp;

void \*ifa\_next;

};

#define ifr\_name ifr\_ifrn.ifrn\_name /\* interface name \*/

#define ifr\_addr ifr\_ifru.ifru\_addr /\* address \*/

#define ifr\_broadaddr ifr\_ifru.ifru\_broadaddr /\* broadcast address \*/

#define ifr\_data ifr\_ifru.ifru\_data /\* for use by interface \*/

#define ifr\_dstaddr ifr\_ifru.ifru\_dstaddr /\* other end of p-p lnk \*/

#define ifr\_flags ifr\_ifru.ifru\_flags /\* flags \*/

#define ifr\_hwaddr ifr\_ifru.ifru\_hwaddr /\* interface name \*/

#define ifr\_bandwidth ifr\_ifru.ifru\_ivalue /\* link bandwidth \*/

#define ifr\_ifindex ifr\_ifru.ifru\_ivalue /\* interface index \*/

#define ifr\_metric ifr\_ifru.ifru\_ivalue /\* metric \*/

#define ifr\_qlen ifr\_ifru.ifru\_ivalue /\* queue length \*/

#define ifr\_mtu ifr\_ifru.ifru\_mtu /\* mtu \*/

#define ifr\_netmask ifr\_ifru.ifru\_netmask /\* interface net mask \*/

#define ifr\_slave ifr\_ifru.ifru\_slave /\* slave device \*/

#define IFNAMSIZ IF\_NAMESIZE

struct ifreq {

union {

char ifrn\_name[IFNAMSIZ];

} ifr\_ifrn;

union {

struct sockaddr ifru\_addr;

struct sockaddr ifru\_dstaddr;

struct sockaddr ifru\_broadaddr;

struct sockaddr ifru\_netmask;

struct sockaddr ifru\_hwaddr;

short ifru\_flags;

int ifru\_ivalue;

int ifru\_mtu;

char ifru\_slave[IFNAMSIZ];

char ifru\_newname[IFNAMSIZ];

caddr\_t ifru\_data;

struct ifmap ifru\_map;

} ifr\_ifru;

};

#define ifc\_buf ifc\_ifcu.ifcu\_buf /\* Buffer address. \*/

#define ifc\_req ifc\_ifcu.ifcu\_req /\* Array of structures. \*/

struct ifconf {

int ifc\_len;

union {

caddr\_t ifcu\_buf;

struct ifreq \*ifcu\_req;

} ifc\_ifcu;

};

extern void if\_freenameindex(struct if\_nameindex \*\_\_ptr);

extern char \*if\_indextoname(unsigned int \_\_ifindex, char \*\_\_ifname);

extern struct if\_nameindex \*if\_nameindex(void);

extern unsigned int if\_nametoindex(const char \*\_\_ifname);

### **14.4.36 netdb.h**

#define h\_errno (\*\_\_h\_errno\_location ())

#define NETDB\_INTERNAL -1 /\* See errno. \*/

#define NETDB\_SUCCESS 0 /\* No problem. \*/

#define HOST\_NOT\_FOUND 1 /\* Authoritative Answer Host not found. \*/

#define IPPORT\_RESERVED 1024

#define NI\_MAXHOST 1025

#define TRY\_AGAIN 2 /\* Non-Authoritative Host not found, or SERVERFAIL. \*/

#define NO\_RECOVERY 3 /\* Non recoverable errors, FORMERR, REFUSED, NOTIMP. \*/

#define NI\_MAXSERV 32

#define NO\_DATA 4 /\* Valid name, no data record of requested type. \*/

#define h\_addr h\_addr\_list[0]

#define NO\_ADDRESS NO\_DATA /\* No address, look for MX record. \*/

struct servent {

char \*s\_name;

char \*\*s\_aliases;

int s\_port;

char \*s\_proto;

};

struct hostent {

char \*h\_name;

char \*\*h\_aliases;

int h\_addrtype;

int h\_length;

char \*\*h\_addr\_list;

};

struct protoent {

char \*p\_name;

char \*\*p\_aliases;

int p\_proto;

};

struct netent {

char \*n\_name;

char \*\*n\_aliases;

int n\_addrtype;

unsigned int n\_net;

};

#define AI\_PASSIVE 0x0001 /\* Socket address is intended for `bind' \*/

#define AI\_CANONNAME 0x0002 /\* Request for canonical name \*/

#define AI\_NUMERICHOST 0x0004 /\* Don't use name resolution \*/

#define AI\_V4MAPPED 0x0008 /\* IPv4 mapped addresses are acceptable. \*/

#define AI\_ALL 0x0010 /\* Return IPv4 mapped and IPv6 addresses. \*/

#define AI\_ADDRCONFIG 0x0020 /\* Use configuration of this host to choose returned address type.. \*/

#define AI\_NUMERICSERV 0x0400 /\* Don't use name resolution \*/

struct addrinfo {

int ai\_flags;

int ai\_family;

int ai\_socktype;

int ai\_protocol;

socklen\_t ai\_addrlen;

struct sockaddr \*ai\_addr;

char \*ai\_canonname;

struct addrinfo \*ai\_next;

};

#define NI\_NUMERICHOST 1

#define NI\_DGRAM 16

#define NI\_NUMERICSERV 2

#define NI\_NOFQDN 4

#define NI\_NAMEREQD 8

#define EAI\_BADFLAGS -1 /\* Invalid value for `ai\_flags' field. \*/

#define EAI\_MEMORY -10 /\* Memory allocation failure. \*/

#define EAI\_SYSTEM -11 /\* System error returned in `errno'. \*/

#define EAI\_NONAME -2 /\* NAME or SERVICE is unknown. \*/

#define EAI\_AGAIN -3 /\* Temporary failure in name resolution. \*/

#define EAI\_FAIL -4 /\* Non-recoverable failure in name res. \*/

#define EAI\_NODATA -5 /\* No address associated with NAME. \*/

#define EAI\_FAMILY -6 /\* `ai\_family' not supported. \*/

#define EAI\_SOCKTYPE -7 /\* `ai\_family' not supported. \*/

#define EAI\_SERVICE -8 /\* SERVICE not supported for `ai\_socktype'. \*/

#define EAI\_ADDRFAMILY -9 /\* Address family for NAME not supported. \*/

extern int \*\_\_h\_errno\_location(void);

extern void endprotoent(void);

extern void endservent(void);

extern void freeaddrinfo(struct addrinfo \*\_\_ai);

extern const char \*gai\_strerror(int \_\_ecode);

extern int getaddrinfo(const char \*\_\_name, const char \*\_\_service,

const struct addrinfo \*\_\_req,

struct addrinfo \*\*\_\_pai);

extern struct hostent \*gethostbyaddr(const void \*\_\_addr, socklen\_t \_\_len,

int \_\_type);

extern int gethostbyaddr\_r(const void \*\_\_addr, socklen\_t \_\_len, int \_\_type,

struct hostent \*\_\_result\_buf, char \*\_\_buf,

size\_t \_\_buflen, struct hostent \*\*\_\_result,

int \*\_\_h\_errnop);

extern struct hostent \*gethostbyname(const char \*\_\_name);

extern struct hostent \*gethostbyname2(const char \*\_\_name, int \_\_af);

extern int gethostbyname2\_r(const char \*\_\_name, int \_\_af,

struct hostent \*\_\_result\_buf, char \*\_\_buf,

size\_t \_\_buflen, struct hostent \*\*\_\_result,

int \*\_\_h\_errnop);

extern int gethostbyname\_r(const char \*\_\_name,

struct hostent \*\_\_result\_buf, char \*\_\_buf,

size\_t \_\_buflen, struct hostent \*\*\_\_result,

int \*\_\_h\_errnop);

extern struct protoent \*getprotobyname(const char \*\_\_name);

extern int getprotobyname\_r(const char \*\_\_name,

struct protoent \*\_\_result\_buf, char \*\_\_buf,

size\_t \_\_buflen, struct protoent \*\*\_\_result);

extern struct protoent \*getprotobynumber(int \_\_proto);

extern int getprotobynumber\_r(int \_\_proto, struct protoent \*\_\_result\_buf,

char \*\_\_buf, size\_t \_\_buflen,

struct protoent \*\*\_\_result);

extern struct protoent \*getprotoent(void);

extern int getprotoent\_r(struct protoent \*\_\_result\_buf, char \*\_\_buf,

size\_t \_\_buflen, struct protoent \*\*\_\_result);

extern struct servent \*getservbyname(const char \*\_\_name,

const char \*\_\_proto);

extern int getservbyname\_r(const char \*\_\_name, const char \*\_\_proto,

struct servent \*\_\_result\_buf, char \*\_\_buf,

size\_t \_\_buflen, struct servent \*\*\_\_result);

extern struct servent \*getservbyport(int \_\_port, const char \*\_\_proto);

extern int getservbyport\_r(int \_\_port, const char \*\_\_proto,

struct servent \*\_\_result\_buf, char \*\_\_buf,

size\_t \_\_buflen, struct servent \*\*\_\_result);

extern struct servent \*getservent(void);

extern int getservent\_r(struct servent \*\_\_result\_buf, char \*\_\_buf,

size\_t \_\_buflen, struct servent \*\*\_\_result);

extern void setprotoent(int \_\_stay\_open);

extern void setservent(int \_\_stay\_open);

### **14.4.37 netinet/icmp6.h**

#define ICMP6\_FILTER\_WILLBLOCK(type,filterp) ((((filterp)->icmp6\_filt[(type) >> 5]) & (1 << ((type) & 31))) != 0)

#define ICMP6\_FILTER\_WILLPASS(type,filterp) ((((filterp)->icmp6\_filt[(type) >> 5]) & (1 << ((type) & 31))) == 0)

#define ICMP6\_FILTER\_SETPASS(type,filterp) ((((filterp)->icmp6\_filt[(type) >> 5]) &= ~(1 << ((type) & 31))))

#define ICMP6\_FILTER\_SETBLOCK(type,filterp) ((((filterp)->icmp6\_filt[(type) >> 5]) |= (1 << ((type) & 31))))

#define ICMP6\_DST\_UNREACH\_NOROUTE 0

#define ICMP6\_PARAMPROB\_HEADER 0

#define ICMP6\_TIME\_EXCEED\_TRANSIT 0

#define ICMP6\_RR\_FLAGS\_PREVDONE 0x08

#define ICMP6\_RR\_FLAGS\_SPECSITE 0x10

#define ICMP6\_RR\_PCOUSE\_RAFLAGS\_AUTO 0x10

#define ICMP6\_RR\_FLAGS\_FORCEAPPLY 0x20

#define ICMP6\_RR\_PCOUSE\_RAFLAGS\_ONLINK 0x20

#define ND\_OPT\_PI\_FLAG\_RADDR 0x20

#define ND\_RA\_FLAG\_HOME\_AGENT 0x20

#define ICMP6\_RR\_FLAGS\_REQRESULT 0x40

#define ND\_OPT\_PI\_FLAG\_AUTO 0x40

#define ND\_RA\_FLAG\_OTHER 0x40

#define ICMP6\_INFOMSG\_MASK 0x80

#define ICMP6\_RR\_FLAGS\_TEST 0x80

#define ND\_OPT\_PI\_FLAG\_ONLINK 0x80

#define ND\_RA\_FLAG\_MANAGED 0x80

#define ICMP6\_DST\_UNREACH 1

#define ICMP6\_DST\_UNREACH\_ADMIN 1

#define ICMP6\_FILTER 1

#define ICMP6\_FILTER\_BLOCK 1

#define ICMP6\_PARAMPROB\_NEXTHEADER 1

#define ICMP6\_TIME\_EXCEED\_REASSEMBLY 1

#define ND\_OPT\_SOURCE\_LINKADDR 1

#define RPM\_PCO\_ADD 1

#define ICMP6\_ECHO\_REQUEST 128

#define ICMP6\_ECHO\_REPLY 129

#define MLD\_LISTENER\_QUERY 130

#define MLD\_LISTENER\_REPORT 131

#define MLD\_LISTENER\_REDUCTION 132

#define ND\_ROUTER\_SOLICIT 133

#define ND\_ROUTER\_ADVERT 134

#define ND\_NEIGHBOR\_SOLICIT 135

#define ND\_NEIGHBOR\_ADVERT 136

#define ND\_REDIRECT 137

#define ICMP6\_ROUTER\_RENUMBERING 138

#define ICMP6\_DST\_UNREACH\_BEYONDSCOPE 2

#define ICMP6\_FILTER\_PASS 2

#define ICMP6\_PACKET\_TOO\_BIG 2

#define ICMP6\_PARAMPROB\_OPTION 2

#define ND\_OPT\_TARGET\_LINKADDR 2

#define RPM\_PCO\_CHANGE 2

#define ICMP6\_DST\_UNREACH\_ADDR 3

#define ICMP6\_FILTER\_BLOCKOTHERS 3

#define ICMP6\_TIME\_EXCEEDED 3

#define ND\_OPT\_PREFIX\_INFORMATION 3

#define RPM\_PCO\_SETGLOBAL 3

#define ICMP6\_DST\_UNREACH\_NOPORT 4

#define ICMP6\_FILTER\_PASSONLY 4

#define ICMP6\_PARAM\_PROB 4

#define ND\_OPT\_REDIRECTED\_HEADER 4

#define ND\_OPT\_MTU 5

#define ND\_OPT\_RTR\_ADV\_INTERVAL 7

#define ND\_OPT\_HOME\_AGENT\_INFO 8

#define icmp6\_id icmp6\_data16[0]

#define icmp6\_maxdelay icmp6\_data16[0]

#define icmp6\_seq icmp6\_data16[1]

#define icmp6\_mtu icmp6\_data32[0]

#define icmp6\_pptr icmp6\_data32[0]

#define icmp6\_data16 icmp6\_dataun.icmp6\_un\_data16

#define icmp6\_data32 icmp6\_dataun.icmp6\_un\_data32

#define icmp6\_data8 icmp6\_dataun.icmp6\_un\_data8

#define ICMP6\_FILTER\_SETPASSALL(filterp) memset (filterp, 0, sizeof (struct icmp6\_filter));

#define ICMP6\_FILTER\_SETBLOCKALL(filterp) memset (filterp, 0xFF, sizeof (struct icmp6\_filter));

#define mld\_cksum mld\_icmp6\_hdr.icmp6\_cksum

#define mld\_code mld\_icmp6\_hdr.icmp6\_code

#define mld\_maxdelay mld\_icmp6\_hdr.icmp6\_data16[0]

#define mld\_reserved mld\_icmp6\_hdr.icmp6\_data16[1]

#define mld\_type mld\_icmp6\_hdr.icmp6\_type

#define nd\_na\_cksum nd\_na\_hdr.icmp6\_cksum

#define nd\_na\_code nd\_na\_hdr.icmp6\_code

#define nd\_na\_flags\_reserved nd\_na\_hdr.icmp6\_data32[0]

#define nd\_na\_type nd\_na\_hdr.icmp6\_type

#define nd\_ns\_cksum nd\_ns\_hdr.icmp6\_cksum

#define nd\_ns\_code nd\_ns\_hdr.icmp6\_code

#define nd\_ns\_reserved nd\_ns\_hdr.icmp6\_data32[0]

#define nd\_ns\_type nd\_ns\_hdr.icmp6\_type

#define nd\_ra\_cksum nd\_ra\_hdr.icmp6\_cksum

#define nd\_ra\_code nd\_ra\_hdr.icmp6\_code

#define nd\_ra\_router\_lifetime nd\_ra\_hdr.icmp6\_data16[1]

#define nd\_ra\_curhoplimit nd\_ra\_hdr.icmp6\_data8[0]

#define nd\_ra\_flags\_reserved nd\_ra\_hdr.icmp6\_data8[1]

#define nd\_ra\_type nd\_ra\_hdr.icmp6\_type

#define nd\_rd\_cksum nd\_rd\_hdr.icmp6\_cksum

#define nd\_rd\_code nd\_rd\_hdr.icmp6\_code

#define nd\_rd\_reserved nd\_rd\_hdr.icmp6\_data32[0]

#define nd\_rd\_type nd\_rd\_hdr.icmp6\_type

#define nd\_rs\_cksum nd\_rs\_hdr.icmp6\_cksum

#define nd\_rs\_code nd\_rs\_hdr.icmp6\_code

#define nd\_rs\_reserved nd\_rs\_hdr.icmp6\_data32[0]

#define nd\_rs\_type nd\_rs\_hdr.icmp6\_type

#define rr\_cksum rr\_hdr.icmp6\_cksum

#define rr\_code rr\_hdr.icmp6\_code

#define rr\_seqnum rr\_hdr.icmp6\_data32[0]

#define rr\_type rr\_hdr.icmp6\_type

struct icmp6\_filter {

uint32\_t icmp6\_filt[8];

};

struct icmp6\_hdr {

uint8\_t icmp6\_type;

uint8\_t icmp6\_code;

uint16\_t icmp6\_cksum;

union {

uint32\_t icmp6\_un\_data32[1];

uint16\_t icmp6\_un\_data16[2];

uint8\_t icmp6\_un\_data8[4];

} icmp6\_dataun;

};

struct nd\_router\_solicit {

struct icmp6\_hdr nd\_rs\_hdr;

};

struct nd\_router\_advert {

struct icmp6\_hdr nd\_ra\_hdr;

uint32\_t nd\_ra\_reachable;

uint32\_t nd\_ra\_retransmit;

};

struct nd\_neighbor\_solicit {

struct icmp6\_hdr nd\_ns\_hdr;

struct in6\_addr nd\_ns\_target;

};

struct nd\_neighbor\_advert {

struct icmp6\_hdr nd\_na\_hdr;

struct in6\_addr nd\_na\_target;

};

struct nd\_redirect {

struct icmp6\_hdr nd\_rd\_hdr;

struct in6\_addr nd\_rd\_target;

struct in6\_addr nd\_rd\_dst;

};

struct nd\_opt\_hdr {

uint8\_t nd\_opt\_type;

uint8\_t nd\_opt\_len;

};

struct nd\_opt\_prefix\_info {

uint8\_t nd\_opt\_pi\_type;

uint8\_t nd\_opt\_pi\_len;

uint8\_t nd\_opt\_pi\_prefix\_len;

uint8\_t nd\_opt\_pi\_flags\_reserved;

uint32\_t nd\_opt\_pi\_valid\_time;

uint32\_t nd\_opt\_pi\_preferred\_time;

uint32\_t nd\_opt\_pi\_reserved2;

struct in6\_addr nd\_opt\_pi\_prefix;

};

struct nd\_opt\_rd\_hdr {

uint8\_t nd\_opt\_rh\_type;

uint8\_t nd\_opt\_rh\_len;

uint16\_t nd\_opt\_rh\_reserved1;

uint32\_t nd\_opt\_rh\_reserved2;

};

struct nd\_opt\_mtu {

uint8\_t nd\_opt\_mtu\_type;

uint8\_t nd\_opt\_mtu\_len;

uint16\_t nd\_opt\_mtu\_reserved;

uint32\_t nd\_opt\_mtu\_mtu;

};

struct mld\_hdr {

struct icmp6\_hdr mld\_icmp6\_hdr;

struct in6\_addr mld\_addr;

};

struct icmp6\_router\_renum {

struct icmp6\_hdr rr\_hdr;

uint8\_t rr\_segnum;

uint8\_t rr\_flags;

uint16\_t rr\_maxdelay;

uint32\_t rr\_reserved;

};

struct rr\_pco\_match {

uint8\_t rpm\_code;

uint8\_t rpm\_len;

uint8\_t rpm\_ordinal;

uint8\_t rpm\_matchlen;

uint8\_t rpm\_minlen;

uint8\_t rpm\_maxlen;

uint16\_t rpm\_reserved;

struct in6\_addr rpm\_prefix;

};

struct rr\_pco\_use {

uint8\_t rpu\_uselen;

uint8\_t rpu\_keeplen;

uint8\_t rpu\_ramask;

uint8\_t rpu\_raflags;

uint32\_t rpu\_vltime;

uint32\_t rpu\_pltime;

uint32\_t rpu\_flags;

struct in6\_addr rpu\_prefix;

};

struct rr\_result {

uint16\_t rrr\_flags;

uint8\_t rrr\_ordinal;

uint8\_t rrr\_matchedlen;

uint32\_t rrr\_ifid;

struct in6\_addr rrr\_prefix;

};

struct nd\_opt\_adv\_interval {

uint8\_t nd\_opt\_adv\_interval\_type;

uint8\_t nd\_opt\_adv\_interval\_len;

uint16\_t nd\_opt\_adv\_interval\_reserved;

uint32\_t nd\_opt\_adv\_interval\_ival;

};

struct nd\_opt\_home\_agent\_info {

uint8\_t nd\_opt\_home\_agent\_info\_type;

uint8\_t nd\_opt\_home\_agent\_info\_len;

uint16\_t nd\_opt\_home\_agent\_info\_reserved;

int16\_t nd\_opt\_home\_agent\_info\_preference;

uint16\_t nd\_opt\_home\_agent\_info\_lifetime;

};

### **14.4.38 netinet/igmp.h**

#define IGMP\_MEMBERSHIP\_QUERY 0x11

#define IGMP\_V1\_MEMBERSHIP\_REPORT 0x12

#define IGMP\_DVMRP 0x13

#define IGMP\_PIM 0x14

#define IGMP\_TRACE 0x15

#define IGMP\_V2\_MEMBERSHIP\_REPORT 0x16

#define IGMP\_V2\_LEAVE\_GROUP 0x17

#define IGMP\_MTRACE\_RESP 0x1e

#define IGMP\_MTRACE 0x1f

#define IGMP\_DELAYING\_MEMBER 1

#define IGMP\_v1\_ROUTER 1

#define IGMP\_MAX\_HOST\_REPORT\_DELAY 10

#define IGMP\_TIMER\_SCALE 10

#define IGMP\_IDLE\_MEMBER 2

#define IGMP\_v2\_ROUTER 2

#define IGMP\_LAZY\_MEMBER 3

#define IGMP\_SLEEPING\_MEMBER 4

#define IGMP\_AWAKENING\_MEMBER 5

#define IGMP\_MINLEN 8

#define IGMP\_HOST\_MEMBERSHIP\_QUERY IGMP\_MEMBERSHIP\_QUERY

#define IGMP\_HOST\_MEMBERSHIP\_REPORT IGMP\_V1\_MEMBERSHIP\_REPORT

#define IGMP\_HOST\_LEAVE\_MESSAGE IGMP\_V2\_LEAVE\_GROUP

#define IGMP\_HOST\_NEW\_MEMBERSHIP\_REPORT IGMP\_V2\_MEMBERSHIP\_REPORT

struct igmp {

u\_int8\_t igmp\_type;

u\_int8\_t igmp\_code;

u\_int16\_t igmp\_cksum;

struct in\_addr igmp\_group;

};

### **14.4.39 netinet/in.h**

#define IPPROTO\_IP 0

#define IPPROTO\_ICMP 1

#define IPPROTO\_UDP 17

#define IPPROTO\_IGMP 2

#define IPPROTO\_RAW 255

#define IPPROTO\_IPV6 41

#define IPPROTO\_ICMPV6 58

#define IPPROTO\_TCP 6

typedef uint16\_t in\_port\_t;

struct in\_addr {

uint32\_t s\_addr;

};

typedef uint32\_t in\_addr\_t;

#define INADDR\_NONE ((in\_addr\_t) 0xffffffff)

#define INADDR\_BROADCAST (0xffffffff)

#define INADDR\_ANY 0

#define INADDR\_LOOPBACK 0x7f000001 /\* 127.0.0.1 \*/

#define s6\_addr16 in6\_u.u6\_addr16

#define s6\_addr32 in6\_u.u6\_addr32

#define s6\_addr in6\_u.u6\_addr8

struct in6\_addr {

union {

uint8\_t u6\_addr8[16];

uint16\_t u6\_addr16[8];

uint32\_t u6\_addr32[4];

} in6\_u;

};

#define IN6ADDR\_ANY\_INIT { { { 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 } } }

#define IN6ADDR\_LOOPBACK\_INIT { { { 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1 } } }

#define IN\_MULTICAST(a) ((((in\_addr\_t)(a))&0xf0000000)==0xe0000000)

#define INET\_ADDRSTRLEN 16

struct sockaddr\_in {

sa\_family\_t sin\_family;

unsigned short sin\_port;

struct in\_addr sin\_addr;

unsigned char sin\_zero[8];

};

#define IN6\_IS\_ADDR\_LINKLOCAL(a) ((((const uint32\_t \*) (a))[0] & htonl (0xffc00000)) == htonl (0xfe800000))

#define IN6\_IS\_ADDR\_SITELOCAL(a) ((((const uint32\_t \*) (a))[0] & htonl (0xffc00000)) == htonl (0xfec00000))

#define IN6\_ARE\_ADDR\_EQUAL(a,b) ((((const uint32\_t \*) (a))[0] == ((const uint32\_t \*) (b))[0]) && (((const uint32\_t \*) (a))[1] == ((const uint32\_t \*) (b))[1]) && (((const uint32\_t \*) (a))[2] == ((const uint32\_t \*) (b))[2]) && (((const uint32\_t \*) (a))[3] == ((const uint32\_t \*) (b))[3]))

#define IN6\_IS\_ADDR\_V4COMPAT(a) ((((const uint32\_t \*) (a))[0] == 0) && (((const uint32\_t \*) (a))[1] == 0) && (((const uint32\_t \*) (a))[2] == 0) && (ntohl (((const uint32\_t \*) (a))[3]) > 1))

#define IN6\_IS\_ADDR\_V4MAPPED(a) ((((const uint32\_t \*) (a))[0] == 0) && (((const uint32\_t \*) (a))[1] == 0) && (((const uint32\_t \*) (a))[2] == htonl (0xffff)))

#define IN6\_IS\_ADDR\_UNSPECIFIED(a) (((const uint32\_t \*) (a))[0] == 0 && ((const uint32\_t \*) (a))[1] == 0 && ((const uint32\_t \*) (a))[2] == 0 && ((const uint32\_t \*) (a))[3] == 0)

#define IN6\_IS\_ADDR\_LOOPBACK(a) (((const uint32\_t \*) (a))[0] == 0 && ((const uint32\_t \*) (a))[1] == 0 && ((const uint32\_t \*) (a))[2] == 0 && ((const uint32\_t \*) (a))[3] == htonl (1))

#define IN6\_IS\_ADDR\_MULTICAST(a) (((const uint8\_t \*) (a))[0] == 0xff)

#define IN6\_IS\_ADDR\_MC\_NODELOCAL(a) (IN6\_IS\_ADDR\_MULTICAST(a) && ((((const uint8\_t \*) (a))[1] & 0xf) == 0x1))

#define IN6\_IS\_ADDR\_MC\_LINKLOCAL(a) (IN6\_IS\_ADDR\_MULTICAST(a) && ((((const uint8\_t \*) (a))[1] & 0xf) == 0x2))

#define IN6\_IS\_ADDR\_MC\_SITELOCAL(a) (IN6\_IS\_ADDR\_MULTICAST(a) && ((((const uint8\_t \*) (a))[1] & 0xf) == 0x5))

#define IN6\_IS\_ADDR\_MC\_ORGLOCAL(a) (IN6\_IS\_ADDR\_MULTICAST(a) && ((((const uint8\_t \*) (a))[1] & 0xf) == 0x8))

#define IN6\_IS\_ADDR\_MC\_GLOBAL(a) (IN6\_IS\_ADDR\_MULTICAST(a) && ((((const uint8\_t \*) (a))[1] & 0xf) == 0xe))

#define INET6\_ADDRSTRLEN 46

struct sockaddr\_in6 {

unsigned short sin6\_family; /\* AF\_INET6 \*/

uint16\_t sin6\_port; /\* Transport layer port # \*/

uint32\_t sin6\_flowinfo; /\* IPv6 flow information \*/

struct in6\_addr sin6\_addr; /\* IPv6 address \*/

uint32\_t sin6\_scope\_id; /\* scope id (new in RFC2553) \*/

};

#define SOL\_IP 0

#define IP\_TOS 1 /\* IP type of service and precedence \*/

#define IPV6\_UNICAST\_HOPS 16

#define IPV6\_MULTICAST\_IF 17

#define IPV6\_MULTICAST\_HOPS 18

#define IPV6\_MULTICAST\_LOOP 19

#define IP\_TTL 2 /\* IP time to live \*/

#define IPV6\_JOIN\_GROUP 20

#define IPV6\_LEAVE\_GROUP 21

#define IPV6\_V6ONLY 26

#define IP\_MULTICAST\_IF 32 /\* set/get IP multicast i/f \*/

#define IP\_MULTICAST\_TTL 33 /\* set/get IP multicast ttl \*/

#define IP\_MULTICAST\_LOOP 34 /\* set/get IP multicast loopback \*/

#define IP\_ADD\_MEMBERSHIP 35 /\* add an IP group membership \*/

#define IP\_DROP\_MEMBERSHIP 36 /\* drop an IP group membership \*/

#define IP\_OPTIONS 4 /\* IP per-packet options \*/

#define IPV6\_ADD\_MEMBERSHIP IPV6\_JOIN\_GROUP

#define IPV6\_DROP\_MEMBERSHIP IPV6\_LEAVE\_GROUP

struct ipv6\_mreq {

struct in6\_addr ipv6mr\_multiaddr; /\* IPv6 multicast address of group \*/

int ipv6mr\_interface; /\* local IPv6 address of interface \*/

};

struct ip\_mreq {

struct in\_addr imr\_multiaddr; /\* IP multicast address of group \*/

struct in\_addr imr\_interface; /\* local IP address of interface \*/

};

extern int bindresvport(int, struct sockaddr\_in \*);

extern const struct in6\_addr in6addr\_any;

extern const struct in6\_addr in6addr\_loopback;

### **14.4.40 netinet/in\_systm.h**

typedef u\_int16\_t n\_short;

typedef u\_int32\_t n\_long;

typedef u\_int32\_t n\_time;

### **14.4.41 netinet/ip.h**

#define IPOPT\_CLASS(o) ((o) & IPOPT\_CLASS\_MASK)

#define IPOPT\_COPIED(o) ((o) & IPOPT\_COPY)

#define IPOPT\_NUMBER(o) ((o) & IPOPT\_NUMBER\_MASK)

#define IPOPT\_EOL 0

#define IPOPT\_OPTVAL 0

#define IPOPT\_TS\_TSONLY 0

#define IPOPT\_CONTROL 0x00

#define IPOPT\_SECUR\_UNCLASS 0x0000

#define IPOPT\_NUMBER\_MASK 0x1f

#define IP\_OFFMASK 0x1fff

#define IPOPT\_RESERVED1 0x20

#define IP\_MF 0x2000

#define IPOPT\_DEBMEAS 0x40

#define IP\_DF 0x4000

#define IPOPT\_CLASS\_MASK 0x60

#define IPOPT\_RESERVED2 0x60

#define IPOPT\_SECUR\_TOPSECRET 0x6bc5

#define IPOPT\_SECUR\_EFTO 0x789a

#define IPOPT\_COPY 0x80

#define IP\_RF 0x8000

#define IPOPT\_SECUR\_RESTR 0xaf13

#define IPOPT\_SECUR\_MMMM 0xbc4d

#define IPOPT\_SECUR\_SECRET 0xd788

#define IPOPT\_SECUR\_CONFID 0xf135

#define IPOPT\_NOP 1

#define IPOPT\_OLEN 1

#define IPOPT\_TS\_TSANDADDR 1

#define IPTTLDEC 1

#define IPOPT\_SECURITY 130

#define IPOPT\_LSRR 131

#define IPOPT\_SATID 136

#define IPOPT\_SSRR 137

#define IPOPT\_RA 148

#define IPOPT\_OFFSET 2

#define MAXTTL 255

#define IPOPT\_TS\_PRESPEC 3

#define IPOPT\_MINOFF 4

#define IPVERSION 4

#define MAX\_IPOPTLEN 40

#define IP\_MSS 576

#define IPFRAGTTL 60

#define IPDEFTTL 64

#define IP\_MAXPACKET 65535

#define IPOPT\_TS 68

#define IPOPT\_RR 7

#define IPOPT\_MEASUREMENT IPOPT\_DEBMEAS

#define IPOPT\_END IPOPT\_EOL

#define IPOPT\_NOOP IPOPT\_NOP

#define IPOPT\_SID IPOPT\_SATID

#define IPOPT\_SEC IPOPT\_SECURITY

#define IPOPT\_TIMESTAMP IPOPT\_TS

#define IPTOS\_TOS(tos) ((tos) & IPTOS\_TOS\_MASK)

#define IPTOS\_LOWCOST 0x02

#define IPTOS\_RELIABILITY 0x04

#define IPTOS\_THROUGHPUT 0x08

#define IPTOS\_LOWDELAY 0x10

#define IPTOS\_TOS\_MASK 0x1e

#define IPTOS\_MINCOST IPTOS\_LOWCOST

#define IPTOS\_PREC(tos) ((tos) & IPTOS\_PREC\_MASK)

#define IPTOS\_PREC\_MASK 0xe0

### **14.4.42 netinet/ip6.h**

#define IP6OPT\_TYPE(o) ((o) & 0xc0)

#define IP6OPT\_PAD1 0

#define IP6OPT\_TYPE\_SKIP 0x00

#define IP6OPT\_TUNNEL\_LIMIT 0x04

#define IP6OPT\_ROUTER\_ALERT 0x05

#define IP6OPT\_TYPE\_MUTABLE 0x20

#define IP6OPT\_TYPE\_DISCARD 0x40

#define IP6OPT\_TYPE\_FORCEICMP 0x80

#define IP6OPT\_TYPE\_ICMP 0xc0

#define IP6OPT\_JUMBO 0xc2

#define IP6OPT\_NSAP\_ADDR 0xc3

#define IP6OPT\_PADN 1

#define IP6OPT\_JUMBO\_LEN 6

#define ip6\_flow ip6\_ctlun.ip6\_un1.ip6\_un1\_flow

#define ip6\_hlim ip6\_ctlun.ip6\_un1.ip6\_un1\_hlim

#define ip6\_hops ip6\_ctlun.ip6\_un1.ip6\_un1\_hlim

#define ip6\_nxt ip6\_ctlun.ip6\_un1.ip6\_un1\_nxt

#define ip6\_plen ip6\_ctlun.ip6\_un1.ip6\_un1\_plen

#define ip6\_vfc ip6\_ctlun.ip6\_un2\_vfc

struct ip6\_hdrctl {

uint32\_t ip6\_un1\_flow;

uint16\_t ip6\_un1\_plen;

uint8\_t ip6\_un1\_nxt;

uint8\_t ip6\_un1\_hlim;

};

struct ip6\_hdr {

struct in6\_addr ip6\_src;

struct in6\_addr ip6\_dst;

};

struct ip6\_ext {

uint8\_t ip6e\_nxt;

uint8\_t ip6e\_len;

};

struct ip6\_hbh {

uint8\_t ip6h\_nxt;

uint8\_t ip6h\_len;

};

struct ip6\_dest {

uint8\_t ip6d\_nxt;

uint8\_t ip6d\_len;

};

struct ip6\_rthdr {

uint8\_t ip6r\_nxt;

uint8\_t ip6r\_len;

uint8\_t ip6r\_type;

uint8\_t ip6r\_segleft;

};

struct ip6\_frag {

uint8\_t ip6f\_nxt;

uint8\_t ip6f\_reserved;

uint16\_t ip6f\_offlg;

uint32\_t ip6f\_ident;

};

struct ip6\_opt {

uint8\_t ip6o\_type;

uint8\_t ip6o\_len;

};

struct ip6\_opt\_jumbo {

uint8\_t ip6oj\_type;

uint8\_t ip6oj\_len;

uint8\_t ip6oj\_jumbo\_len[4];

};

struct ip6\_opt\_nsap {

uint8\_t ip6on\_type;

uint8\_t ip6on\_len;

uint8\_t ip6on\_src\_nsap\_len;

uint8\_t ip6on\_dst\_nsap\_len;

};

struct ip6\_opt\_tunnel {

uint8\_t ip6ot\_type;

uint8\_t ip6ot\_len;

uint8\_t ip6ot\_encap\_limit;

};

struct ip6\_opt\_router {

uint8\_t ip6or\_type;

uint8\_t ip6or\_len;

uint8\_t ip6or\_value[2];

};

### **14.4.43 netinet/ip\_icmp.h**

#define ICMP\_INFOTYPE(type) ((type) == ICMP\_ECHOREPLY || (type) == ICMP\_ECHO || (type) == ICMP\_ROUTERADVERT || (type) == ICMP\_ROUTERSOLICIT || (type) == ICMP\_TSTAMP || (type) == ICMP\_TSTAMPREPLY || (type) == ICMP\_IREQ || (type) == ICMP\_IREQREPLY || (type) == ICMP\_MASKREQ || (type) == ICMP\_MASKREPLY)

#define ICMP\_ADVLEN(p) (8 + ((p)->icmp\_ip.ip\_hl << 2) + 8)

#define ICMP\_TSLEN (8 + 3 \* sizeof (n\_time))

#define ICMP\_ADVLENMIN (8 + sizeof (struct ip) + 8)

#define ICMP\_ECHOREPLY 0

#define ICMP\_EXC\_TTL 0

#define ICMP\_NET\_UNREACH 0

#define ICMP\_REDIRECT\_NET 0

#define ICMP\_REDIR\_NET 0

#define ICMP\_TIMXCEED\_INTRANS 0

#define ICMP\_UNREACH\_NET 0

#define ICMP\_EXC\_FRAGTIME 1

#define ICMP\_HOST\_UNREACH 1

#define ICMP\_PARAMPROB\_OPTABSENT 1

#define ICMP\_REDIRECT\_HOST 1

#define ICMP\_REDIR\_HOST 1

#define ICMP\_TIMXCEED\_REASS 1

#define ICMP\_UNREACH\_HOST 1

#define ICMP\_HOST\_ANO 10

#define ICMP\_ROUTERSOLICIT 10

#define ICMP\_UNREACH\_HOST\_PROHIB 10

#define ICMP\_NET\_UNR\_TOS 11

#define ICMP\_TIME\_EXCEEDED 11

#define ICMP\_TIMXCEED 11

#define ICMP\_UNREACH\_TOSNET 11

#define ICMP\_HOST\_UNR\_TOS 12

#define ICMP\_MASKLEN 12

#define ICMP\_PARAMETERPROB 12

#define ICMP\_PARAMPROB 12

#define ICMP\_UNREACH\_TOSHOST 12

#define ICMP\_PKT\_FILTERED 13

#define ICMP\_TIMESTAMP 13

#define ICMP\_TSTAMP 13

#define ICMP\_UNREACH\_FILTER\_PROHIB 13

#define ICMP\_PREC\_VIOLATION 14

#define ICMP\_TIMESTAMPREPLY 14

#define ICMP\_TSTAMPREPLY 14

#define ICMP\_UNREACH\_HOST\_PRECEDENCE 14

#define ICMP\_INFO\_REQUEST 15

#define ICMP\_IREQ 15

#define ICMP\_PREC\_CUTOFF 15

#define ICMP\_UNREACH\_PRECEDENCE\_CUTOFF 15

#define NR\_ICMP\_UNREACH 15

#define ICMP\_INFO\_REPLY 16

#define ICMP\_IREQREPLY 16

#define ICMP\_ADDRESS 17

#define ICMP\_MASKREQ 17

#define ICMP\_ADDRESSREPLY 18

#define ICMP\_MASKREPLY 18

#define ICMP\_MAXTYPE 18

#define NR\_ICMP\_TYPES 18

#define ICMP\_PROT\_UNREACH 2

#define ICMP\_REDIRECT\_TOSNET 2

#define ICMP\_REDIR\_NETTOS 2

#define ICMP\_UNREACH\_PROTOCOL 2

#define ICMP\_DEST\_UNREACH 3

#define ICMP\_PORT\_UNREACH 3

#define ICMP\_REDIRECT\_TOSHOST 3

#define ICMP\_REDIR\_HOSTTOS 3

#define ICMP\_UNREACH 3

#define ICMP\_UNREACH\_PORT 3

#define ICMP\_FRAG\_NEEDED 4

#define ICMP\_SOURCEQUENCH 4

#define ICMP\_SOURCE\_QUENCH 4

#define ICMP\_UNREACH\_NEEDFRAG 4

#define ICMP\_REDIRECT 5

#define ICMP\_SR\_FAILED 5

#define ICMP\_UNREACH\_SRCFAIL 5

#define ICMP\_NET\_UNKNOWN 6

#define ICMP\_UNREACH\_NET\_UNKNOWN 6

#define ICMP\_HOST\_UNKNOWN 7

#define ICMP\_UNREACH\_HOST\_UNKNOWN 7

#define ICMP\_ECHO 8

#define ICMP\_HOST\_ISOLATED 8

#define ICMP\_MINLEN 8

#define ICMP\_UNREACH\_ISOLATED 8

#define ICMP\_NET\_ANO 9

#define ICMP\_ROUTERADVERT 9

#define ICMP\_UNREACH\_NET\_PROHIB 9

#define icmp\_data icmp\_dun.id\_data

#define icmp\_ip icmp\_dun.id\_ip.idi\_ip

#define icmp\_mask icmp\_dun.id\_mask

#define icmp\_radv icmp\_dun.id\_radv

#define icmp\_otime icmp\_dun.id\_ts.its\_otime

#define icmp\_rtime icmp\_dun.id\_ts.its\_rtime

#define icmp\_ttime icmp\_dun.id\_ts.its\_ttime

#define icmp\_gwaddr icmp\_hun.ih\_gwaddr

#define icmp\_id icmp\_hun.ih\_idseq.icd\_id

#define icmp\_seq icmp\_hun.ih\_idseq.icd\_seq

#define icmp\_nextmtu icmp\_hun.ih\_pmtu.ipm\_nextmtu

#define icmp\_pmvoid icmp\_hun.ih\_pmtu.ipm\_void

#define icmp\_pptr icmp\_hun.ih\_pptr

#define icmp\_lifetime icmp\_hun.ih\_rtradv.irt\_lifetime

#define icmp\_num\_addrs icmp\_hun.ih\_rtradv.irt\_num\_addrs

#define icmp\_wpa icmp\_hun.ih\_rtradv.irt\_wpa

#define icmp\_void icmp\_hun.ih\_void

struct icmphdr {

u\_int8\_t type;

u\_int8\_t code;

u\_int16\_t checksum;

union {

struct {

u\_int16\_t id;

u\_int16\_t sequence;

} echo;

u\_int32\_t gateway;

struct {

u\_int16\_t \_\_unused;

u\_int16\_t mtu;

} frag;

} un;

};

struct icmp\_ra\_addr {

u\_int32\_t ira\_addr;

u\_int32\_t ira\_preference;

};

struct ih\_idseq {

u\_int16\_t icd\_id;

u\_int16\_t icd\_seq;

};

struct ih\_pmtu {

u\_int16\_t ipm\_void;

u\_int16\_t ipm\_nextmtu;

};

struct ih\_rtradv {

u\_int8\_t irt\_num\_addrs;

u\_int8\_t irt\_wpa;

u\_int16\_t irt\_lifetime;

};

struct icmp {

u\_int8\_t icmp\_type;

u\_int8\_t icmp\_code;

u\_int16\_t icmp\_cksum;

union {

u\_int16\_t ih\_pptr;

struct in\_addr ih\_gwaddr;

struct ih\_idseq ih\_idseq;

u\_int32\_t ih\_void;

struct ih\_pmtu ih\_pmtu;

struct ih\_rtradv ih\_rtradv;

} icmp\_hun;

union {

struct {

u\_int32\_t its\_otime;

u\_int32\_t its\_rtime;

u\_int32\_t its\_ttime;

} id\_ts;

struct {

struct ip idi\_ip;

} id\_ip;

struct icmp\_ra\_addr id\_radv;

u\_int32\_t id\_mask;

u\_int8\_t id\_data[1];

} icmp\_dun;

};

### **14.4.44 netinet/tcp.h**

#define TCPOLEN\_TSTAMP\_APPA (TCPOLEN\_TIMESTAMP+2)

#define TCPOPT\_TSTAMP\_HDR (TCPOPT\_NOP<<24|TCPOPT\_NOP<<16|TCPOPT\_TIMESTAMP<<8|TCPOLEN\_TIMESTAMP)

#define TCPOPT\_EOL 0

#define TCPI\_OPT\_TIMESTAMPS 1

#define TCPOPT\_NOP 1

#define TCP\_NODELAY 1

#define TCPOLEN\_TIMESTAMP 10

#define TCP\_WINDOW\_CLAMP 10

#define TCP\_INFO 11

#define TCP\_QUICKACK 12

#define TCP\_CONGESTION 13

#define TCP\_MAX\_WINSHIFT 14

#define TCPI\_OPT\_SACK 2

#define TCPOLEN\_SACK\_PERMITTED 2

#define TCPOPT\_MAXSEG 2

#define TCP\_MAXSEG 2

#define TCPOLEN\_WINDOW 3

#define TCPOPT\_WINDOW 3

#define TCP\_CORK 3

#define TCPI\_OPT\_WSCALE 4

#define TCPOLEN\_MAXSEG 4

#define TCPOPT\_SACK\_PERMITTED 4

#define TCP\_KEEPIDLE 4

#define TCPOPT\_SACK 5

#define TCP\_KEEPINTVL 5

#define TCP\_MSS 512

#define SOL\_TCP 6

#define TCP\_KEEPCNT 6

#define TCP\_MAXWIN 65535

#define TCP\_SYNCNT 7

#define TCPI\_OPT\_ECN 8

#define TCPOPT\_TIMESTAMP 8

#define TCP\_LINGER2 8

#define TCP\_DEFER\_ACCEPT 9

enum tcp\_ca\_state {

TCP\_CA\_Open,

TCP\_CA\_Disorder,

TCP\_CA\_CWR,

TCP\_CA\_Recovery,

TCP\_CA\_Loss

};

struct tcp\_info {

uint8\_t tcpi\_state;

uint8\_t tcpi\_ca\_state;

uint8\_t tcpi\_retransmits;

uint8\_t tcpi\_probes;

uint8\_t tcpi\_backoff;

uint8\_t tcpi\_options;

uint8\_t tcpi\_snd\_wscale:4;

uint8\_t tcpi\_rcv\_wscale:4;

uint32\_t tcpi\_rto;

uint32\_t tcpi\_ato;

uint32\_t tcpi\_snd\_mss;

uint32\_t tcpi\_rcv\_mss;

uint32\_t tcpi\_unacked;

uint32\_t tcpi\_sacked;

uint32\_t tcpi\_lost;

uint32\_t tcpi\_retrans;

uint32\_t tcpi\_fackets;

uint32\_t tcpi\_last\_data\_sent;

uint32\_t tcpi\_last\_ack\_sent;

uint32\_t tcpi\_last\_data\_recv;

uint32\_t tcpi\_last\_ack\_recv;

uint32\_t tcpi\_pmtu;

uint32\_t tcpi\_rcv\_ssthresh;

uint32\_t tcpi\_rtt;

uint32\_t tcpi\_rttvar;

uint32\_t tcpi\_snd\_ssthresh;

uint32\_t tcpi\_snd\_cwnd;

uint32\_t tcpi\_advmss;

uint32\_t tcpi\_reordering;

};

enum {

TCP\_ESTABLISHED = 1,

TCP\_SYN\_SENT = 2,

TCP\_SYN\_RECV = 3,

TCP\_FIN\_WAIT1 = 4,

TCP\_FIN\_WAIT2 = 5,

TCP\_TIME\_WAIT = 6,

TCP\_CLOSE = 7,

TCP\_CLOSE\_WAIT = 8,

TCP\_LAST\_ACK = 9,

TCP\_LISTEN = 10,

TCP\_CLOSING = 11

};

### **14.4.45 netinet/udp.h**

#define SOL\_UDP 17

struct udphdr {

u\_int16\_t source;

u\_int16\_t dest;

u\_int16\_t len;

u\_int16\_t check;

};

### **14.4.46 nl\_types.h**

#define NL\_CAT\_LOCALE 1

#define NL\_SETD 1

typedef void \*nl\_catd;

typedef int nl\_item;

extern int catclose(nl\_catd \_\_catalog);

extern char \*catgets(nl\_catd \_\_catalog, int \_\_set, int \_\_number,

const char \*\_\_string);

extern nl\_catd catopen(const char \*\_\_cat\_name, int \_\_flag);

### **14.4.47 poll.h**

extern int poll(struct pollfd \*\_\_fds, nfds\_t \_\_nfds, int \_\_timeout);

### **14.4.48 pwd.h**

struct passwd {

char \*pw\_name;

char \*pw\_passwd;

uid\_t pw\_uid;

gid\_t pw\_gid;

char \*pw\_gecos;

char \*pw\_dir;

char \*pw\_shell;

};

extern void endpwent(void);

extern struct passwd \*getpwent(void);

extern int getpwent\_r(struct passwd \*\_\_resultbuf, char \*\_\_buffer,

size\_t \_\_buflen, struct passwd \*\*\_\_result);

extern struct passwd \*getpwnam(const char \*\_\_name);

extern int getpwnam\_r(const char \*\_\_name, struct passwd \*\_\_resultbuf,

char \*\_\_buffer, size\_t \_\_buflen,

struct passwd \*\*\_\_result);

extern struct passwd \*getpwuid(uid\_t \_\_uid);

extern int getpwuid\_r(uid\_t \_\_uid, struct passwd \*\_\_resultbuf,

char \*\_\_buffer, size\_t \_\_buflen,

struct passwd \*\*\_\_result);

extern void setpwent(void);

### **14.4.49 regex.h**

#define RE\_DUP\_MAX (0x7fff)

typedef unsigned long int reg\_syntax\_t;

typedef struct re\_pattern\_buffer {

unsigned char \*buffer;

unsigned long int allocated;

unsigned long int used;

reg\_syntax\_t syntax;

char \*fastmap;

char \*translate;

size\_t re\_nsub;

unsigned int can\_be\_null:1;

unsigned int regs\_allocated:2;

unsigned int fastmap\_accurate:1;

unsigned int no\_sub:1;

unsigned int not\_bol:1;

unsigned int not\_eol:1;

unsigned int newline\_anchor:1;

} regex\_t;

typedef int regoff\_t;

typedef struct {

regoff\_t rm\_so;

regoff\_t rm\_eo;

} regmatch\_t;

#define REG\_ICASE (REG\_EXTENDED<<1)

#define REG\_NEWLINE (REG\_ICASE<<1)

#define REG\_NOSUB (REG\_NEWLINE<<1)

#define REG\_EXTENDED 1

#define REG\_NOTEOL (1<<1)

#define REG\_NOTBOL 1

typedef enum {

REG\_ENOSYS = -1,

REG\_NOERROR = 0,

REG\_NOMATCH = 1,

REG\_BADPAT = 2,

REG\_ECOLLATE = 3,

REG\_ECTYPE = 4,

REG\_EESCAPE = 5,

REG\_ESUBREG = 6,

REG\_EBRACK = 7,

REG\_EPAREN = 8,

REG\_EBRACE = 9,

REG\_BADBR = 10,

REG\_ERANGE = 11,

REG\_ESPACE = 12,

REG\_BADRPT = 13,

REG\_EEND = 14,

REG\_ESIZE = 15,

REG\_ERPAREN = 16

} reg\_errcode\_t;

extern int regcomp(regex\_t \* \_\_preg, const char \*\_\_pattern, int \_\_cflags);

extern size\_t regerror(int \_\_errcode, const regex\_t \* \_\_preg,

char \*\_\_errbuf, size\_t \_\_errbuf\_size);

extern int regexec(const regex\_t \* \_\_preg, const char \*\_\_string,

size\_t \_\_nmatch, regmatch\_t \_\_pmatch[], int \_\_eflags);

extern void regfree(regex\_t \* \_\_preg);

### **14.4.50 rpc/auth.h**

#define auth\_destroy(auth) ((\*((auth)->ah\_ops->ah\_destroy))(auth))

enum auth\_stat {

AUTH\_OK = 0,

AUTH\_BADCRED = 1, /\* bogus credentials (seal broken) \*/

AUTH\_REJECTEDCRED = 2, /\* client should begin new session \*/

AUTH\_BADVERF = 3, /\* bogus verifier (seal broken) \*/

AUTH\_REJECTEDVERF = 4, /\* verifier expired or was replayed \*/

AUTH\_TOOWEAK = 5, /\* Rpc calls return an enum clnt\_stat. \*/

AUTH\_INVALIDRESP = 6, /\* bogus response verifier \*/

AUTH\_FAILED = 7 /\* some unknown reason \*/

};

union des\_block {

struct {

u\_int32\_t high;

u\_int32\_t low;

} key;

char c[8];

};

struct opaque\_auth {

enum\_t oa\_flavor; /\* flavor of auth \*/

caddr\_t oa\_base; /\* address of more auth stuff \*/

u\_int oa\_length; /\* not to exceed MAX\_AUTH\_BYTES \*/

};

typedef struct AUTH {

struct opaque\_auth ah\_cred;

struct opaque\_auth ah\_verf;

union des\_block ah\_key;

struct auth\_ops \*ah\_ops;

caddr\_t ah\_private;

} AUTH;

struct auth\_ops {

void (\*ah\_nextverf) (struct AUTH \*);

int (\*ah\_marshal) (struct AUTH \*, XDR \*); /\* nextverf & serialize \*/

int (\*ah\_validate) (struct AUTH \*, struct opaque\_auth \*); /\* validate verifier \*/

int (\*ah\_refresh) (struct AUTH \*); /\* refresh credentials \*/

void (\*ah\_destroy) (struct AUTH \*); /\* Rpc calls return an enum clnt\_stat. \*/

};

extern struct AUTH \*authnone\_create(void);

extern int key\_decryptsession(char \*, union des\_block \*);

extern bool\_t xdr\_opaque\_auth(XDR \*, struct opaque\_auth \*);

### **14.4.51 rpc/clnt.h**

#define clnt\_control(cl,rq,in) ((\*(cl)->cl\_ops->cl\_control)(cl,rq,in))

#define clnt\_abort(rh) ((\*(rh)->cl\_ops->cl\_abort)(rh))

#define clnt\_destroy(rh) ((\*(rh)->cl\_ops->cl\_destroy)(rh))

#define clnt\_freeres(rh,xres,resp) ((\*(rh)->cl\_ops->cl\_freeres)(rh,xres,resp))

#define clnt\_geterr(rh,errp) ((\*(rh)->cl\_ops->cl\_geterr)(rh, errp))

#define NULLPROC ((u\_long)0) /\* By convention, procedure 0 takes null arguments and returns \*/

#define CLSET\_TIMEOUT 1 /\* set timeout (timeval) \*/

#define CLGET\_XID 10 /\* Get xid \*/

#define CLSET\_XID 11 /\* Set xid \*/

#define CLGET\_VERS 12 /\* Get version number \*/

#define CLSET\_VERS 13 /\* Set version number \*/

#define CLGET\_PROG 14 /\* Get program number \*/

#define CLSET\_PROG 15 /\* Set program number \*/

#define CLGET\_TIMEOUT 2 /\* get timeout (timeval) \*/

#define CLGET\_SERVER\_ADDR 3 /\* get server's address (sockaddr) \*/

#define CLSET\_RETRY\_TIMEOUT 4 /\* set retry timeout (timeval) \*/

#define CLGET\_RETRY\_TIMEOUT 5 /\* get retry timeout (timeval) \*/

#define CLGET\_FD 6 /\* get connections file descriptor \*/

#define CLGET\_SVC\_ADDR 7 /\* get server's address (netbuf) \*/

#define CLSET\_FD\_CLOSE 8 /\* close fd while clnt\_destroy \*/

#define CLSET\_FD\_NCLOSE 9 /\* Do not close fd while clnt\_destroy \*/

#define clnt\_call(rh, proc, xargs, argsp, xres, resp, secs) \

((\*(rh)->cl\_ops->cl\_call)(rh, proc, xargs, argsp, xres, resp, secs))

enum clnt\_stat {

RPC\_SUCCESS = 0, /\* call succeeded \*/

RPC\_CANTENCODEARGS = 1, /\* can't encode arguments \*/

RPC\_CANTDECODERES = 2, /\* can't decode results \*/

RPC\_CANTSEND = 3, /\* failure in sending call \*/

RPC\_CANTRECV = 4, /\* failure in receiving result \*/

RPC\_TIMEDOUT = 5, /\* call timed out \*/

RPC\_VERSMISMATCH = 6, /\* rpc versions not compatible \*/

RPC\_AUTHERROR = 7, /\* authentication error \*/

RPC\_PROGUNAVAIL = 8, /\* program not available \*/

RPC\_PROGVERSMISMATCH = 9, /\* program version mismatched \*/

RPC\_PROCUNAVAIL = 10, /\* procedure unavailable \*/

RPC\_CANTDECODEARGS = 11, /\* decode arguments error \*/

RPC\_SYSTEMERROR = 12, /\* generic "other problem" \*/

RPC\_NOBROADCAST = 21, /\* Broadcasting not supported \*/

RPC\_UNKNOWNHOST = 13, /\* unknown host name \*/

RPC\_UNKNOWNPROTO = 17, /\* unknown protocol \*/

RPC\_UNKNOWNADDR = 19, /\* Remote address unknown \*/

RPC\_RPCBFAILURE = 14, /\* portmapper failed in its call \*/

RPC\_PROGNOTREGISTERED = 15, /\* remote program is not registered \*/

RPC\_N2AXLATEFAILURE = 22, /\* Name to addr translation failed \*/

RPC\_FAILED = 16,

RPC\_INTR = 18,

RPC\_TLIERROR = 20,

RPC\_UDERROR = 23,

RPC\_INPROGRESS = 24,

RPC\_STALERACHANDLE = 25

};

struct rpc\_err {

enum clnt\_stat re\_status;

union {

int RE\_errno;

enum auth\_stat RE\_why;

struct {

u\_long low;

u\_long high;

} RE\_vers;

struct {

long int s1;

long int s2;

} RE\_lb;

} ru;

};

typedef struct CLIENT {

struct AUTH \*cl\_auth;

struct clnt\_ops \*cl\_ops;

caddr\_t cl\_private;

} CLIENT;

struct clnt\_ops {

enum clnt\_stat (\*cl\_call) (struct CLIENT \*, u\_long, xdrproc\_t, caddr\_t,

xdrproc\_t, caddr\_t, struct timeval);

void (\*cl\_abort) (void);

void (\*cl\_geterr) (struct CLIENT \*, struct rpc\_err \*);

bool\_t(\*cl\_freeres) (struct CLIENT \*, xdrproc\_t, caddr\_t);

void (\*cl\_destroy) (struct CLIENT \*);

bool\_t(\*cl\_control) (struct CLIENT \*, int, char \*);

};

extern int callrpc(const char \*\_\_host, const u\_long \_\_prognum,

const u\_long \_\_versnum, const u\_long \_\_procnum,

const xdrproc\_t \_\_inproc, const char \*\_\_in,

const xdrproc\_t \_\_outproc, char \*\_\_out);

extern struct CLIENT \*clnt\_create(const char \*\_\_host, const u\_long \_\_prog,

const u\_long \_\_vers, const char \*\_\_prot);

extern void clnt\_pcreateerror(const char \*\_\_msg);

extern void clnt\_perrno(enum clnt\_stat \_\_num);

extern void clnt\_perror(struct CLIENT \*\_\_clnt, const char \*\_\_msg);

extern char \*clnt\_spcreateerror(const char \*\_\_msg);

extern char \*clnt\_sperrno(enum clnt\_stat \_\_num);

extern char \*clnt\_sperror(struct CLIENT \*\_\_clnt, const char \*\_\_msg);

extern struct CLIENT \*clntraw\_create(u\_long \_\_prog, u\_long \_\_vers);

extern struct CLIENT \*clnttcp\_create(struct sockaddr\_in \*\_\_raddr,

u\_long \_\_prog, u\_long \_\_version,

int \*\_\_sockp, u\_int \_\_sendsz,

u\_int \_\_recvsz);

extern struct CLIENT \*clntudp\_bufcreate(struct sockaddr\_in \*\_\_raddr,

u\_long \_\_program, u\_long \_\_version,

struct timeval \_\_wait\_resend,

int \*\_\_sockp, u\_int \_\_sendsz,

u\_int \_\_recvsz);

extern struct CLIENT \*clntudp\_create(struct sockaddr\_in \*\_\_raddr,

u\_long \_\_program, u\_long \_\_version,

struct timeval \_\_wait\_resend,

int \*\_\_sockp);

### **14.4.52 rpc/pmap\_clnt.h**

extern u\_short pmap\_getport(struct sockaddr\_in \*\_\_address,

const u\_long \_\_program, const u\_long \_\_version,

u\_int \_\_protocol);

extern bool\_t pmap\_set(const u\_long \_\_program, const u\_long \_\_vers,

int \_\_protocol, u\_short \_\_port);

extern bool\_t pmap\_unset(u\_long \_\_program, u\_long \_\_vers);

### **14.4.53 rpc/rpc\_msg.h**

enum msg\_type {

CALL = 0,

REPLY = 1

};

enum reply\_stat {

MSG\_ACCEPTED = 0,

MSG\_DENIED = 1

};

enum accept\_stat {

SUCCESS = 0,

PROG\_UNAVAIL = 1,

PROG\_MISMATCH = 2,

PROC\_UNAVAIL = 3,

GARBAGE\_ARGS = 4,

SYSTEM\_ERR = 5

};

enum reject\_stat {

RPC\_MISMATCH = 0,

AUTH\_ERROR = 1

};

#define ar\_results ru.AR\_results

#define ar\_vers ru.AR\_versions

struct accepted\_reply {

struct opaque\_auth ar\_verf;

enum accept\_stat ar\_stat;

union {

struct {

unsigned long int low;

unsigned long int high;

} AR\_versions;

struct {

caddr\_t where;

xdrproc\_t proc;

} AR\_results;

} ru;

};

#define rj\_vers ru.RJ\_versions

#define rj\_why ru.RJ\_why

struct rejected\_reply {

enum reject\_stat rj\_stat;

union {

struct {

unsigned long int low;

unsigned long int high;

} RJ\_versions;

enum auth\_stat RJ\_why; /\* why authentication did not work \*/

} ru;

};

#define rp\_acpt ru.RP\_ar

#define rp\_rjct ru.RP\_dr

struct reply\_body {

enum reply\_stat rp\_stat;

union {

struct accepted\_reply RP\_ar;

struct rejected\_reply RP\_dr;

} ru;

};

struct call\_body {

unsigned long int cb\_rpcvers; /\* must be equal to two \*/

unsigned long int cb\_prog;

unsigned long int cb\_vers;

unsigned long int cb\_proc;

struct opaque\_auth cb\_cred;

struct opaque\_auth cb\_verf; /\* protocol specific - provided by client \*/

};

#define rm\_call ru.RM\_cmb

#define rm\_reply ru.RM\_rmb

#define acpted\_rply ru.RM\_rmb.ru.RP\_ar

#define rjcted\_rply ru.RM\_rmb.ru.RP\_dr

struct rpc\_msg {

unsigned long int rm\_xid;

enum msg\_type rm\_direction;

union {

struct call\_body RM\_cmb;

struct reply\_body RM\_rmb;

} ru;

};

extern bool\_t xdr\_accepted\_reply(XDR \*, struct accepted\_reply \*);

extern bool\_t xdr\_callhdr(XDR \* \_\_xdrs, struct rpc\_msg \*\_\_cmsg);

extern bool\_t xdr\_callmsg(XDR \* \_\_xdrs, struct rpc\_msg \*\_\_cmsg);

extern bool\_t xdr\_rejected\_reply(XDR \*, struct rejected\_reply \*);

extern bool\_t xdr\_replymsg(XDR \* \_\_xdrs, struct rpc\_msg \*\_\_rmsg);

### **14.4.54 rpc/svc.h**

#define svc\_getcaller(x) (&(x)->xp\_raddr)

#define svc\_destroy(xprt) (\*(xprt)->xp\_ops->xp\_destroy)(xprt)

#define svc\_recv(xprt,msg) (\*(xprt)->xp\_ops->xp\_recv)((xprt), (msg))

#define svc\_reply(xprt,msg) (\*(xprt)->xp\_ops->xp\_reply) ((xprt), (msg))

#define svc\_stat(xprt) (\*(xprt)->xp\_ops->xp\_stat)(xprt)

#define RPC\_ANYSOCK -1

#define svc\_freeargs(xprt,xargs, argsp) \

(\*(xprt)->xp\_ops->xp\_freeargs)((xprt), (xargs), (argsp))

#define svc\_getargs(xprt,xargs, argsp) \

(\*(xprt)->xp\_ops->xp\_getargs)((xprt), (xargs), (argsp))

enum xprt\_stat {

XPRT\_DIED,

XPRT\_MOREREQS,

XPRT\_IDLE

};

typedef struct SVCXPRT {

int xp\_sock;

u\_short xp\_port;

struct xp\_ops \*xp\_ops;

int xp\_addrlen;

struct sockaddr\_in xp\_raddr;

struct opaque\_auth xp\_verf;

caddr\_t xp\_p1;

caddr\_t xp\_p2;

char xp\_pad[256];

} SVCXPRT;

struct svc\_req {

rpcprog\_t rq\_prog;

rpcvers\_t rq\_vers;

rpcproc\_t rq\_proc;

struct opaque\_auth rq\_cred;

caddr\_t rq\_clntcred;

SVCXPRT \*rq\_xprt;

};

typedef void (\*\_\_dispatch\_fn\_t) (struct svc\_req \*, SVCXPRT \*);

struct xp\_ops {

bool\_t(\*xp\_recv) (SVCXPRT \* \_\_xprt, struct rpc\_msg \* \_\_msg);

enum xprt\_stat (\*xp\_stat) (SVCXPRT \* \_\_xprt);

bool\_t(\*xp\_getargs) (SVCXPRT \* \_\_xprt, xdrproc\_t \_\_xdr\_args,

caddr\_t args\_ptr);

bool\_t(\*xp\_reply) (SVCXPRT \* \_\_xprt, struct rpc\_msg \* \_\_msg);

bool\_t(\*xp\_freeargs) (SVCXPRT \* \_\_xprt, xdrproc\_t \_\_xdr\_args,

caddr\_t args\_ptr);

void (\*xp\_destroy) (SVCXPRT \* \_\_xprt);

};

extern void svc\_getreqset(fd\_set \* \_\_readfds);

extern bool\_t svc\_register(SVCXPRT \* \_\_xprt, rpcprog\_t \_\_prog,

rpcvers\_t \_\_vers, \_\_dispatch\_fn\_t \_\_dispatch,

rpcprot\_t \_\_protocol);

extern void svc\_run(void);

extern bool\_t svc\_sendreply(SVCXPRT \* xprt, xdrproc\_t \_\_xdr\_results,

caddr\_t \_\_xdr\_location);

extern void svcerr\_auth(SVCXPRT \* \_\_xprt, enum auth\_stat \_\_why);

extern void svcerr\_decode(SVCXPRT \* \_\_xprt);

extern void svcerr\_noproc(SVCXPRT \* \_\_xprt);

extern void svcerr\_noprog(SVCXPRT \* \_\_xprt);

extern void svcerr\_progvers(SVCXPRT \* \_\_xprt, rpcvers\_t \_\_low\_vers,

rpcvers\_t \_\_high\_vers);

extern void svcerr\_systemerr(SVCXPRT \* \_\_xprt);

extern void svcerr\_weakauth(SVCXPRT \* \_\_xprt);

extern SVCXPRT \*svcfd\_create(int, unsigned int, unsigned int);

extern SVCXPRT \*svcraw\_create(void);

extern SVCXPRT \*svctcp\_create(int \_\_sock, u\_int \_\_sendsize,

u\_int \_\_recvsize);

extern SVCXPRT \*svcudp\_create(int \_\_sock);

### **14.4.55 rpc/types.h**

typedef int bool\_t;

typedef int enum\_t;

typedef unsigned long int rpcprog\_t;

typedef unsigned long int rpcvers\_t;

typedef unsigned long int rpcproc\_t;

typedef unsigned long int rpcprot\_t;

### **14.4.56 rpc/xdr.h**

#define XDR\_DESTROY(xdrs) \

do { if ((xdrs)->x\_ops->x\_destroy) (\*(xdrs)->x\_ops->x\_destroy)(xdrs); \

} while (0)

#define xdr\_destroy(xdrs) \

do { if ((xdrs)->x\_ops->x\_destroy) (\*(xdrs)->x\_ops->x\_destroy)(xdrs); \

} while (0)

#define XDR\_GETBYTES(xdrs,addr,len) (\*(xdrs)->x\_ops->x\_getbytes)(xdrs, addr, len)

#define xdr\_getbytes(xdrs,addr,len) (\*(xdrs)->x\_ops->x\_getbytes)(xdrs, addr, len)

#define XDR\_GETINT32(xdrs,int32p) (\*(xdrs)->x\_ops->x\_getint32)(xdrs, int32p)

#define xdr\_getint32(xdrs,int32p) (\*(xdrs)->x\_ops->x\_getint32)(xdrs, int32p)

#define XDR\_GETLONG(xdrs,longp) (\*(xdrs)->x\_ops->x\_getlong)(xdrs, longp)

#define xdr\_getlong(xdrs,longp) (\*(xdrs)->x\_ops->x\_getlong)(xdrs, longp)

#define XDR\_GETPOS(xdrs) (\*(xdrs)->x\_ops->x\_getpostn)(xdrs)

#define xdr\_getpos(xdrs) (\*(xdrs)->x\_ops->x\_getpostn)(xdrs)

#define XDR\_INLINE(xdrs,len) (\*(xdrs)->x\_ops->x\_inline)(xdrs, len)

#define xdr\_inline(xdrs,len) (\*(xdrs)->x\_ops->x\_inline)(xdrs, len)

#define XDR\_PUTBYTES(xdrs,addr,len) (\*(xdrs)->x\_ops->x\_putbytes)(xdrs, addr, len)

#define xdr\_putbytes(xdrs,addr,len) (\*(xdrs)->x\_ops->x\_putbytes)(xdrs, addr, len)

#define XDR\_PUTINT32(xdrs,int32p) (\*(xdrs)->x\_ops->x\_putint32)(xdrs, int32p)

#define xdr\_putint32(xdrs,int32p) (\*(xdrs)->x\_ops->x\_putint32)(xdrs, int32p)

#define XDR\_PUTLONG(xdrs,longp) (\*(xdrs)->x\_ops->x\_putlong)(xdrs, longp)

#define xdr\_putlong(xdrs,longp) (\*(xdrs)->x\_ops->x\_putlong)(xdrs, longp)

#define XDR\_SETPOS(xdrs,pos) (\*(xdrs)->x\_ops->x\_setpostn)(xdrs, pos)

#define xdr\_setpos(xdrs,pos) (\*(xdrs)->x\_ops->x\_setpostn)(xdrs, pos)

enum xdr\_op {

XDR\_ENCODE,

XDR\_DECODE,

XDR\_FREE

};

typedef struct XDR {

enum xdr\_op x\_op;

struct xdr\_ops \*x\_ops;

caddr\_t x\_public;

caddr\_t x\_private;

caddr\_t x\_base;

int x\_handy;

} XDR;

struct xdr\_ops {

bool\_t(\*x\_getlong) (XDR \* \_\_xdrs, long int \*\_\_lp);

bool\_t(\*x\_putlong) (XDR \* \_\_xdrs, long int \*\_\_lp);

bool\_t(\*x\_getbytes) (XDR \* \_\_xdrs, caddr\_t \_\_addr, u\_int \_\_len);

bool\_t(\*x\_putbytes) (XDR \* \_\_xdrs, char \*\_\_addr, u\_int \_\_len);

u\_int(\*x\_getpostn) (XDR \* \_\_xdrs);

bool\_t(\*x\_setpostn) (XDR \* \_\_xdrs, u\_int \_\_pos);

int32\_t \*(\*x\_inline) (XDR \* \_\_xdrs, int \_\_len);

void (\*x\_destroy) (XDR \* \_\_xdrs);

bool\_t(\*x\_getint32) (XDR \* \_\_xdrs, int32\_t \* \_\_ip);

bool\_t(\*x\_putint32) (XDR \* \_\_xdrs, int32\_t \* \_\_ip);

};

typedef bool\_t(\*xdrproc\_t) (XDR \*, void \*, ...);

struct xdr\_discrim {

int value;

xdrproc\_t proc;

};

extern bool\_t xdr\_array(XDR \* \_xdrs, caddr\_t \* \_\_addrp, u\_int \* \_\_sizep,

u\_int \_\_maxsize, u\_int \_\_elsize,

xdrproc\_t \_\_elproc);

extern bool\_t xdr\_bool(XDR \* \_\_xdrs, bool\_t \* \_\_bp);

extern bool\_t xdr\_bytes(XDR \* \_\_xdrs, char \*\*\_\_cpp, u\_int \* \_\_sizep,

u\_int \_\_maxsize);

extern bool\_t xdr\_char(XDR \* \_\_xdrs, char \*\_\_cp);

extern bool\_t xdr\_double(XDR \* \_\_xdrs, double \*\_\_dp);

extern bool\_t xdr\_enum(XDR \* \_\_xdrs, enum\_t \* \_\_ep);

extern bool\_t xdr\_float(XDR \* \_\_xdrs, float \*\_\_fp);

extern void xdr\_free(xdrproc\_t \_\_proc, char \*\_\_objp);

extern bool\_t xdr\_int(XDR \* \_\_xdrs, int \*\_\_ip);

extern bool\_t xdr\_long(XDR \* \_\_xdrs, long int \*\_\_lp);

extern bool\_t xdr\_opaque(XDR \* \_\_xdrs, caddr\_t \_\_cp, u\_int \_\_cnt);

extern bool\_t xdr\_pointer(XDR \* \_\_xdrs, char \*\*\_\_objpp, u\_int \_\_obj\_size,

xdrproc\_t \_\_xdr\_obj);

extern bool\_t xdr\_reference(XDR \* \_\_xdrs, caddr\_t \* \_\_xpp, u\_int \_\_size,

xdrproc\_t \_\_proc);

extern bool\_t xdr\_short(XDR \* \_\_xdrs, short \*\_\_sp);

extern bool\_t xdr\_string(XDR \* \_\_xdrs, char \*\*\_\_cpp, u\_int \_\_maxsize);

extern bool\_t xdr\_u\_char(XDR \* \_\_xdrs, u\_char \* \_\_cp);

extern bool\_t xdr\_u\_int(XDR \* \_\_xdrs, u\_int \* \_\_up);

extern bool\_t xdr\_u\_long(XDR \* \_\_xdrs, u\_long \* \_\_ulp);

extern bool\_t xdr\_u\_short(XDR \* \_\_xdrs, u\_short \* \_\_usp);

extern bool\_t xdr\_union(XDR \* \_\_xdrs, enum\_t \* \_\_dscmp, char \*\_\_unp,

const struct xdr\_discrim \*\_\_choices,

xdrproc\_t dfault);

extern bool\_t xdr\_vector(XDR \* \_\_xdrs, char \*\_\_basep, u\_int \_\_nelem,

u\_int \_\_elemsize, xdrproc\_t \_\_xdr\_elem);

extern bool\_t xdr\_void(void);

extern bool\_t xdr\_wrapstring(XDR \* \_\_xdrs, char \*\*\_\_cpp);

extern void xdrmem\_create(XDR \* \_\_xdrs, caddr\_t \_\_addr, u\_int \_\_size,

enum xdr\_op \_\_xop);

extern void xdrrec\_create(XDR \* \_\_xdrs, u\_int \_\_sendsize, u\_int \_\_recvsize,

caddr\_t \_\_tcp\_handle, int (\*\_\_readit) (char \*,

char \*,

int),

int (\*\_\_writeit) (char \*, char \*, int));

extern bool\_t xdrrec\_endofrecord(XDR \* \_\_xdrs, bool\_t \_\_sendnow);

extern bool\_t xdrrec\_eof(XDR \* \_\_xdrs);

extern bool\_t xdrrec\_skiprecord(XDR \* \_\_xdrs);

extern void xdrstdio\_create(XDR \* \_\_xdrs, FILE \* \_\_file,

enum xdr\_op \_\_xop);

### **14.4.57 sched.h**

#define \_\_CPU\_ALLOC\_SIZE(count) ((((count) + \_\_NCPUBITS - 1) / \_\_NCPUBITS) \* 8)

#define \_\_CPUELT(cpu) ((cpu) / \_\_NCPUBITS)

#define \_\_CPUMASK(cpu) ((\_\_cpu\_mask) 1 << ((cpu) % \_\_NCPUBITS))

#define \_\_NCPUBITS (8 \* sizeof (\_\_cpu\_mask))

#define SCHED\_OTHER 0

#define SCHED\_FIFO 1

#define \_\_CPU\_SETSIZE 1024

#define SCHED\_RR 2

#define \_\_CPU\_OP\_S(setsize, destset, srcset1, srcset2, op) \

(\_\_extension\_\_\

({ cpu\_set\_t \*\_\_dest = (destset); \

cpu\_set\_t \*\_\_arr1 = (srcset1); \

cpu\_set\_t \*\_\_arr2 = (srcset2); \

size\_t \_\_imax = (setsize) / sizeof (\_\_cpu\_mask); \

size\_t \_\_i; \

for (\_\_i = 0; \_\_i < \_\_imax; ++\_\_i)\

\_\_dest->\_\_bits[\_\_i] = \_\_arr1->\_\_bits[\_\_i] op \_\_arr2->\_\_bits[\_\_i]; \

\_\_dest; }))

#define \_\_CPU\_SET\_S(cpu, setsize, cpusetp) \

(\_\_extension\_\_\

({ size\_t \_\_cpu = (cpu); \

\_\_cpu < 8 \* (setsize) \

? ((cpusetp)->\_\_bits[\_\_CPUELT (\_\_cpu)] |= \_\_CPUMASK (\_\_cpu)) : 0; }))

#define \_\_CPU\_ISSET\_S(cpu, setsize, cpusetp) \

(\_\_extension\_\_\

({ size\_t \_\_cpu = (cpu); \

\_\_cpu < 8 \* (setsize)\

? (((cpusetp)->\_\_bits[\_\_CPUELT (\_\_cpu)] & \_\_CPUMASK (\_\_cpu))) != 0 \

: 0; }))

#define \_\_CPU\_CLR\_S(cpu, setsize, cpusetp) \

(\_\_extension\_\_\

({ size\_t \_\_cpu = (cpu); \

\_\_cpu < 8 \* (setsize)\

? ((cpusetp)->\_\_bits[\_\_CPUELT (\_\_cpu)] &= ~\_\_CPUMASK (\_\_cpu)) : 0; }))

#define \_\_CPU\_ZERO\_S(setsize, cpusetp) \

do {\

size\_t \_\_i; \

size\_t \_\_imax = (setsize) / sizeof (\_\_cpu\_mask); \

cpu\_set\_t \*\_\_arr = (cpusetp); \

for (\_\_i = 0; \_\_i < \_\_imax; ++\_\_i)\

\_\_arr->\_\_bits[\_\_i] = 0; \

} while (0)

#define CPU\_ALLOC\_SIZE(count) \_\_CPU\_ALLOC\_SIZE (count)

#define CPU\_CLR(cpu, cpusetp) \_\_CPU\_CLR\_S (cpu, sizeof (cpu\_set\_t), cpusetp)

#define CPU\_ISSET(cpu, cpusetp) \_\_CPU\_ISSET\_S (cpu, sizeof (cpu\_set\_t), cpusetp)

#define CPU\_AND\_S(setsize, destset, srcset1, srcset2) \_\_CPU\_OP\_S (setsize, destset, srcset1, srcset2, &)

#define CPU\_XOR\_S(setsize, destset, srcset1, srcset2) \_\_CPU\_OP\_S (setsize, destset, srcset1, srcset2, ^)

#define CPU\_OR\_S(setsize, destset, srcset1, srcset2) \_\_CPU\_OP\_S (setsize, destset, srcset1, srcset2, |)

#define CPU\_AND(destset, srcset1, srcset2) \_\_CPU\_OP\_S (sizeof (cpu\_set\_t), destset, srcset1, srcset2, &)

#define CPU\_XOR(destset, srcset1, srcset2) \_\_CPU\_OP\_S (sizeof (cpu\_set\_t), destset, srcset1, srcset2, ^)

#define CPU\_OR(destset, srcset1, srcset2) \_\_CPU\_OP\_S (sizeof (cpu\_set\_t), destset, srcset1, srcset2, |)

#define CPU\_SETSIZE \_\_CPU\_SETSIZE

#define CPU\_SET(cpu, cpusetp) \_\_CPU\_SET\_S (cpu, sizeof (cpu\_set\_t), cpusetp)

#define CPU\_ZERO(cpusetp) \_\_CPU\_ZERO\_S (sizeof (cpu\_set\_t), cpusetp)

struct sched\_param {

int sched\_priority;

};

typedef unsigned long int \_\_cpu\_mask;

typedef struct {

\_\_cpu\_mask \_\_bits[\_\_CPU\_SETSIZE / \_\_NCPUBITS];

} cpu\_set\_t;

extern int sched\_get\_priority\_max(int \_\_algorithm);

extern int sched\_get\_priority\_min(int \_\_algorithm);

extern int sched\_getaffinity(pid\_t \_\_pid, size\_t \_\_cpusetsize,

cpu\_set\_t \* \_\_cpuset);

extern int sched\_getparam(pid\_t \_\_pid, struct sched\_param \*\_\_param);

extern int sched\_getscheduler(pid\_t \_\_pid);

extern int sched\_rr\_get\_interval(pid\_t \_\_pid, struct timespec \*\_\_t);

extern int sched\_setaffinity(pid\_t \_\_pid, size\_t \_\_cpusetsize,

const cpu\_set\_t \* \_\_cpuset);

extern int sched\_setparam(pid\_t \_\_pid, const struct sched\_param \*\_\_param);

extern int sched\_setscheduler(pid\_t \_\_pid, int \_\_policy,

const struct sched\_param \*\_\_param);

extern int sched\_yield(void);

### **14.4.58 search.h**

typedef struct entry {

char \*key;

void \*data;

} ENTRY;

typedef enum {

FIND,

ENTER

} ACTION;

struct \_ENTRY;

typedef enum {

preorder,

postorder,

endorder,

leaf

} VISIT;

struct hsearch\_data {

struct \_ENTRY \*table;

unsigned int size;

unsigned int filled;

};

typedef void (\*\_\_action\_fn\_t) (const void \*\_\_nodep, VISIT \_\_value,

int \_\_level);

extern int hcreate(size\_t \_\_nel);

extern int hcreate\_r(size\_t \_\_nel, struct hsearch\_data \*\_\_htab);

extern void hdestroy(void);

extern void hdestroy\_r(struct hsearch\_data \*\_\_htab);

extern ENTRY \*hsearch(ENTRY \_\_item, ACTION \_\_action);

extern int hsearch\_r(ENTRY \_\_item, ACTION \_\_action, ENTRY \* \*\_\_retval,

struct hsearch\_data \*\_\_htab);

extern void insque(void \*\_\_elem, void \*\_\_prev);

extern void \*lfind(const void \*\_\_key, const void \*\_\_base, size\_t \* \_\_nmemb,

size\_t \_\_size, \_\_compar\_fn\_t \_\_compar);

extern void \*lsearch(const void \*\_\_key, void \*\_\_base, size\_t \* \_\_nmemb,

size\_t \_\_size, \_\_compar\_fn\_t \_\_compar);

extern void remque(void \*\_\_elem);

extern void \*tdelete(const void \*\_\_key, void \*\*\_\_rootp,

\_\_compar\_fn\_t \_\_compar);

extern void \*tfind(const void \*\_\_key, void \*const \*\_\_rootp,

\_\_compar\_fn\_t \_\_compar);

extern void \*tsearch(const void \*\_\_key, void \*\*\_\_rootp,

\_\_compar\_fn\_t \_\_compar);

extern void twalk(const void \*\_\_root, \_\_action\_fn\_t \_\_action);

### **14.4.59 setjmp.h**

#define setjmp(env) \_setjmp(env)

#define sigsetjmp(a,b) \_\_sigsetjmp(a,b)

struct \_\_jmp\_buf\_tag {

\_\_jmp\_buf \_\_jmpbuf;

int \_\_mask\_was\_saved;

sigset\_t \_\_saved\_mask;

};

typedef struct \_\_jmp\_buf\_tag jmp\_buf[1];

typedef jmp\_buf sigjmp\_buf;

extern int \_\_sigsetjmp(jmp\_buf \_\_env, int \_\_savemask);

extern void \_longjmp(jmp\_buf \_\_env, int \_\_val);

extern int \_setjmp(jmp\_buf \_\_env);

extern void longjmp(jmp\_buf \_\_env, int \_\_val);

extern void siglongjmp(sigjmp\_buf \_\_env, int \_\_val);

### **14.4.60 signal.h**

#define sigpause \_\_xpg\_sigpause

#define \_SIGSET\_NWORDS (1024/(8\*sizeof(unsigned long)))

#define SIGRTMAX (\_\_libc\_current\_sigrtmax ())

#define SIGRTMIN (\_\_libc\_current\_sigrtmin ())

#define NSIG 65

#define SIG\_BLOCK 0 /\* Block signals. \*/

#define SIG\_UNBLOCK 1 /\* Unblock signals. \*/

#define SIG\_SETMASK 2 /\* Set the set of blocked signals. \*/

typedef int sig\_atomic\_t;

typedef void (\*sighandler\_t) (int);

#define SIG\_HOLD ((sighandler\_t) 2) /\* Request that signal be held. \*/

#define SIG\_DFL ((sighandler\_t)0) /\* Request for default signal handling. \*/

#define SIG\_IGN ((sighandler\_t)1) /\* Request that signal be ignored. \*/

#define SIG\_ERR ((sighandler\_t)-1) /\* Return value from signal() in case of error. \*/

#define SIGHUP 1 /\* Hangup. \*/

#define SIGINT 2 /\* Terminal interrupt signal. \*/

#define SIGQUIT 3 /\* Terminal quit signal. \*/

#define SIGILL 4 /\* Illegal instruction. \*/

#define SIGTRAP 5 /\* Trace/breakpoint trap. \*/

#define SIGABRT 6 /\* Process abort signal. \*/

#define SIGIOT 6 /\* IOT trap \*/

#define SIGBUS 7 /\* Access to an undefined portion of a memory object. \*/

#define SIGFPE 8 /\* Erroneous arithmetic operation. \*/

#define SIGKILL 9 /\* Kill (cannot be caught or ignored). \*/

#define SIGUSR1 10 /\* User-defined signal 1. \*/

#define SIGSEGV 11 /\* Invalid memory reference. \*/

#define SIGUSR2 12 /\* User-defined signal 2. \*/

#define SIGPIPE 13 /\* Write on a pipe with no one to read it. \*/

#define SIGALRM 14 /\* Alarm clock. \*/

#define SIGTERM 15 /\* Termination signal. \*/

#define SIGSTKFLT 16 /\* Stack fault. \*/

#define SIGCHLD 17 /\* Child process terminated, stopped, or continued. \*/

#define SIGCLD SIGCHLD /\* Same as SIGCHLD \*/

#define SIGCONT 18 /\* Continue executing, if stopped. \*/

#define SIGSTOP 19 /\* Stop executing (cannot be caught or ignored). \*/

#define SIGTSTP 20 /\* Terminal stop signal. \*/

#define SIGTTIN 21 /\* Background process attempting read. \*/

#define SIGTTOU 22 /\* Background process attempting write. \*/

#define SIGURG 23 /\* High bandwidth data is available at a socket. \*/

#define SIGXCPU 24 /\* CPU time limit exceeded. \*/

#define SIGXFSZ 25 /\* File size limit exceeded. \*/

#define SIGVTALRM 26 /\* Virtual timer expired. \*/

#define SIGPROF 27 /\* Profiling timer expired. \*/

#define SIGWINCH 28 /\* Window size change. \*/

#define SIGIO 29 /\* I/O now possible. \*/

#define SIGPOLL SIGIO /\* Pollable event. \*/

#define SIGPWR 30 /\* Power failure restart \*/

#define SIGSYS 31 /\* Bad system call. \*/

#define SIGUNUSED 31

#define SV\_ONSTACK (1<<0) /\* Take the signal on the signal stack. \*/

#define SV\_INTERRUPT (1<<1) /\* Do not restart system calls. \*/

#define SV\_RESETHAND (1<<2) /\* Reset handler to SIG\_DFL on receipt. \*/

typedef union sigval {

int sival\_int;

void \*sival\_ptr;

} sigval\_t;

#define SIGEV\_SIGNAL 0 /\* Notify via signal. \*/

#define SIGEV\_NONE 1 /\* Other notification: meaningless. \*/

#define SIGEV\_THREAD 2 /\* Deliver via thread creation. \*/

#define SIGEV\_MAX\_SIZE 64

typedef struct sigevent {

sigval\_t sigev\_value;

int sigev\_signo;

int sigev\_notify;

union {

int \_pad[SIGEV\_PAD\_SIZE];

struct {

void (\*\_function) (sigval\_t);

void \*\_attribute;

} \_sigev\_thread;

} \_sigev\_un;

} sigevent\_t;

#define SI\_MAX\_SIZE 128

#define si\_pid \_sifields.\_kill.\_pid

#define si\_uid \_sifields.\_kill.\_uid

#define si\_value \_sifields.\_rt.\_sigval

#define si\_int \_sifields.\_rt.\_sigval.sival\_int

#define si\_ptr \_sifields.\_rt.\_sigval.sival\_ptr

#define si\_status \_sifields.\_sigchld.\_status

#define si\_stime \_sifields.\_sigchld.\_stime

#define si\_utime \_sifields.\_sigchld.\_utime

#define si\_addr \_sifields.\_sigfault.\_addr

#define si\_band \_sifields.\_sigpoll.\_band

#define si\_fd \_sifields.\_sigpoll.\_fd

#define si\_timer1 \_sifields.\_timer.\_timer1

#define si\_timer2 \_sifields.\_timer.\_timer2

#define sigev\_notify\_attributes \_sigev\_un.\_sigev\_thread.\_attribute

#define sigev\_notify\_function \_sigev\_un.\_sigev\_thread.\_function

typedef struct siginfo {

int si\_signo; /\* Signal number. \*/

int si\_errno;

int si\_code; /\* Signal code. \*/

union {

int \_pad[SI\_PAD\_SIZE];

struct {

pid\_t \_pid;

uid\_t \_uid;

} \_kill;

struct {

unsigned int \_timer1;

unsigned int \_timer2;

} \_timer;

struct {

pid\_t \_pid;

uid\_t \_uid;

sigval\_t \_sigval;

} \_rt;

struct {

pid\_t \_pid;

uid\_t \_uid;

int \_status;

clock\_t \_utime;

clock\_t \_stime;

} \_sigchld;

struct {

void \*\_addr;

} \_sigfault;

struct {

int \_band;

int \_fd;

} \_sigpoll;

} \_sifields;

} siginfo\_t;

#define SI\_QUEUE -1 /\* Sent by sigqueue. \*/

#define SI\_TIMER -2 /\* Sent by timer expiration. \*/

#define SI\_MESGQ -3 /\* Sent by real time mesq state change. \*/

#define SI\_ASYNCIO -4 /\* Sent by AIO completion. \*/

#define SI\_SIGIO -5 /\* Sent by queued SIGIO. \*/

#define SI\_TKILL -6 /\* Sent by tkill. \*/

#define SI\_ASYNCNL -60 /\* Sent by asynch name lookup completion. \*/

#define SI\_USER 0 /\* Sent by kill, sigsend, raise. \*/

#define SI\_KERNEL 0x80 /\* Sent by kernel. \*/

#define ILL\_ILLOPC 1 /\* Illegal opcode. \*/

#define ILL\_ILLOPN 2 /\* Illegal operand. \*/

#define ILL\_ILLADR 3 /\* Illegal addressing mode. \*/

#define ILL\_ILLTRP 4 /\* Illegal trap. \*/

#define ILL\_PRVOPC 5 /\* Privileged opcode. \*/

#define ILL\_PRVREG 6 /\* Privileged register. \*/

#define ILL\_COPROC 7 /\* Coprocessor error. \*/

#define ILL\_BADSTK 8 /\* Internal stack error. \*/

#define FPE\_INTDIV 1 /\* Integer divide by zero. \*/

#define FPE\_INTOVF 2 /\* Integer overflow. \*/

#define FPE\_FLTDIV 3 /\* Floating-point divide by zero. \*/

#define FPE\_FLTOVF 4 /\* Floating-point overflow. \*/

#define FPE\_FLTUND 5 /\* Floating-point underflow. \*/

#define FPE\_FLTRES 6 /\* Floating-point inexact result. \*/

#define FPE\_FLTINV 7 /\* Invalid floating-point operation. \*/

#define FPE\_FLTSUB 8 /\* Subscript out of range. \*/

#define SEGV\_MAPERR 1 /\* Address not mapped to object. \*/

#define SEGV\_ACCERR 2 /\* Invalid permissions for mapped object. \*/

#define BUS\_ADRALN 1 /\* Invalid address alignment. \*/

#define BUS\_ADRERR 2 /\* Nonexistent physical address. \*/

#define BUS\_OBJERR 3 /\* Object-specific hardware error. \*/

#define TRAP\_BRKPT 1 /\* Process breakpoint. \*/

#define TRAP\_TRACE 2 /\* Process trace trap. \*/

#define CLD\_EXITED 1 /\* Child has exited. \*/

#define CLD\_KILLED 2 /\* Child has terminated abnormally and did not create a core fi \*/

#define CLD\_DUMPED 3 /\* Child has terminated abnormally and created a core file. \*/

#define CLD\_TRAPPED 4 /\* Traced child has trapped. \*/

#define CLD\_STOPPED 5 /\* Child has stopped. \*/

#define CLD\_CONTINUED 6 /\* Stopped child has continued. \*/

#define POLL\_IN 1 /\* Data input available. \*/

#define POLL\_OUT 2 /\* Output buffers available. \*/

#define POLL\_MSG 3 /\* Input message available. \*/

#define POLL\_ERR 4 /\* I/O error. \*/

#define POLL\_PRI 5 /\* High priority input available. \*/

#define POLL\_HUP 6 /\* Device disconnected. \*/

typedef struct {

unsigned long int sig[\_SIGSET\_NWORDS];

} sigset\_t;

#define SA\_INTERRUPT 0x20000000

#define sa\_handler \_\_sigaction\_handler.\_sa\_handler

#define sa\_sigaction \_\_sigaction\_handler.\_sa\_sigaction

#define SA\_ONSTACK 0x08000000 /\* Use signal stack by using `sa\_restorer`. \*/

#define SA\_RESETHAND 0x80000000 /\* Reset to SIG\_DFL on entry to handler. \*/

#define SA\_NOCLDSTOP 0x00000001 /\* Don't send SIGCHLD when children stop. \*/

#define SA\_SIGINFO 0x00000004 /\* Invoke signal-catching function with three arguments instead of one. \*/

#define SA\_NODEFER 0x40000000 /\* Don't automatically block the signal when its handler is being executed. \*/

#define SA\_RESTART 0x10000000 /\* Restart syscall on signal return. \*/

#define SA\_NOCLDWAIT 0x00000002 /\* Don't create zombie on child death. \*/

#define SA\_NOMASK SA\_NODEFER

#define SA\_ONESHOT SA\_RESETHAND

typedef struct sigaltstack {

void \*ss\_sp;

int ss\_flags;

size\_t ss\_size;

} stack\_t;

#define SS\_ONSTACK 1

#define SS\_DISABLE 2

extern int \_\_libc\_current\_sigrtmax(void);

extern int \_\_libc\_current\_sigrtmin(void);

extern sighandler\_t \_\_sysv\_signal(int \_\_sig, sighandler\_t \_\_handler);

extern int \_\_xpg\_sigpause(int);

extern char \*const \_sys\_siglist[];

extern sighandler\_t bsd\_signal(int \_\_sig, sighandler\_t \_\_handler);

extern int kill(pid\_t \_\_pid, int \_\_sig);

extern int killpg(pid\_t \_\_pgrp, int \_\_sig);

extern void psiginfo(const siginfo\_t \* pinfo, const char \*message);

extern void psignal(int \_\_sig, const char \*\_\_s);

extern int pthread\_kill(pthread\_t, int);

extern int pthread\_sigmask(int, const sigset\_t \*, sigset\_t \*);

extern int raise(int \_\_sig);

extern int sigaction(int \_\_sig, const struct sigaction \*\_\_act,

struct sigaction \*\_\_oact);

extern int sigaddset(sigset\_t \* \_\_set, int \_\_signo);

extern int sigaltstack(const struct sigaltstack \*\_\_ss,

struct sigaltstack \*\_\_oss);

extern int sigandset(sigset\_t \* \_\_set, const sigset\_t \* \_\_left,

const sigset\_t \* \_\_right);

extern int sigdelset(sigset\_t \* \_\_set, int \_\_signo);

extern int sigemptyset(sigset\_t \* \_\_set);

extern int sigfillset(sigset\_t \* \_\_set);

extern int sighold(int \_\_sig);

extern int sigignore(int \_\_sig);

extern int siginterrupt(int \_\_sig, int \_\_interrupt);

extern int sigisemptyset(const sigset\_t \* \_\_set);

extern int sigismember(const sigset\_t \* \_\_set, int \_\_signo);

extern sighandler\_t signal(int \_\_sig, sighandler\_t \_\_handler);

extern int sigorset(sigset\_t \* \_\_set, const sigset\_t \* \_\_left,

const sigset\_t \* \_\_right);

extern int sigpending(sigset\_t \* \_\_set);

extern int sigprocmask(int \_\_how, const sigset\_t \* \_\_set,

sigset\_t \* \_\_oset);

extern int sigqueue(pid\_t \_\_pid, int \_\_sig, const union sigval \_\_val);

extern int sigrelse(int \_\_sig);

extern int sigreturn(struct sigcontext \*\_\_scp);

extern sighandler\_t sigset(int \_\_sig, sighandler\_t \_\_disp);

extern int sigsuspend(const sigset\_t \* \_\_set);

extern int sigtimedwait(const sigset\_t \* \_\_set, siginfo\_t \* \_\_info,

const struct timespec \*\_\_timeout);

extern int sigwait(const sigset\_t \* \_\_set, int \*\_\_sig);

extern int sigwaitinfo(const sigset\_t \* \_\_set, siginfo\_t \* \_\_info);

### **14.4.61 spawn.h**

#define POSIX\_SPAWN\_RESETIDS 0x01

#define POSIX\_SPAWN\_SETPGROUP 0x02

#define POSIX\_SPAWN\_SETSIGDEF 0x04

#define POSIX\_SPAWN\_SETSIGMASK 0x08

#define POSIX\_SPAWN\_SETSCHEDPARAM 0x10

#define POSIX\_SPAWN\_SETSCHEDULER 0x20

typedef struct {

int \_\_allocated;

int \_\_used;

struct \_\_spawn\_action \*\_\_actions;

int \_\_pad[16];

} posix\_spawn\_file\_actions\_t;

typedef struct {

short \_\_flags;

pid\_t \_\_pgrp;

sigset\_t \_\_sd;

sigset\_t \_\_ss;

struct sched\_param \_\_sp;

int \_\_policy;

int \_\_pad[16];

} posix\_spawnattr\_t;

extern int posix\_spawn(pid\_t \* \_\_pid, const char \*\_\_path,

const posix\_spawn\_file\_actions\_t \* \_\_file\_actions,

const posix\_spawnattr\_t \* \_\_attrp,

char \*const argv[], char \*const envp[]);

extern int posix\_spawn\_file\_actions\_addclose(posix\_spawn\_file\_actions\_t \*

\_\_file\_actions, int \_\_fd);

extern int posix\_spawn\_file\_actions\_adddup2(posix\_spawn\_file\_actions\_t \*

\_\_file\_actions, int \_\_fd,

int \_\_newfd);

extern int posix\_spawn\_file\_actions\_addopen(posix\_spawn\_file\_actions\_t \*

\_\_file\_actions, int \_\_fd,

const char \*\_\_path,

int \_\_oflag, mode\_t \_\_mode);

extern int posix\_spawn\_file\_actions\_destroy(posix\_spawn\_file\_actions\_t \*

\_\_file\_actions);

extern int posix\_spawn\_file\_actions\_init(posix\_spawn\_file\_actions\_t \*

\_\_file\_actions);

extern int posix\_spawnattr\_destroy(posix\_spawnattr\_t \* \_\_attr);

extern int posix\_spawnattr\_getflags(const posix\_spawnattr\_t \* \_\_attr,

short int \*\_\_flags);

extern int posix\_spawnattr\_getpgroup(const posix\_spawnattr\_t \* \_\_attr,

pid\_t \* \_\_pgroup);

extern int posix\_spawnattr\_getschedparam(const posix\_spawnattr\_t \* \_\_attr,

struct sched\_param \*\_\_schedparam);

extern int posix\_spawnattr\_getschedpolicy(const posix\_spawnattr\_t \* \_\_attr,

int \*\_\_schedpolicy);

extern int posix\_spawnattr\_getsigdefault(const posix\_spawnattr\_t \* \_\_attr,

sigset\_t \* \_\_sigdefault);

extern int posix\_spawnattr\_getsigmask(const posix\_spawnattr\_t \* \_\_attr,

sigset\_t \* \_\_sigmask);

extern int posix\_spawnattr\_init(posix\_spawnattr\_t \* \_\_attr);

extern int posix\_spawnattr\_setflags(posix\_spawnattr\_t \* \_attr,

short int \_\_flags);

extern int posix\_spawnattr\_setpgroup(posix\_spawnattr\_t \* \_\_attr,

pid\_t \_\_pgroup);

extern int posix\_spawnattr\_setschedparam(posix\_spawnattr\_t \* \_\_attr,

const struct sched\_param

\*\_\_schedparam);

extern int posix\_spawnattr\_setschedpolicy(posix\_spawnattr\_t \* \_\_attr,

int \_\_schedpolicy);

extern int posix\_spawnattr\_setsigdefault(posix\_spawnattr\_t \* \_\_attr,

const sigset\_t \* \_\_sigdefault);

extern int posix\_spawnattr\_setsigmask(posix\_spawnattr\_t \* \_\_attr,

const sigset\_t \* \_\_sigmask);

extern int posix\_spawnp(pid\_t \* \_\_pid, const char \*\_\_file,

const posix\_spawn\_file\_actions\_t \* \_\_file\_actions,

const posix\_spawnattr\_t \* \_\_attrp,

char \*const argv[], char \*const envp[]);

### **14.4.62 stddef.h**

#if !defined(\_\_GNUC\_\_)

#define \_\_builtin\_offsetof (TYPE, MEMBER) ((size\_t)&((TYPE\*)0)->MEMBER)

#endif

#ifndef NULL

# ifdef \_\_cplusplus

# define NULL (0L)

# else

# define NULL ((void\*) 0)

# endif

#endif

#define offsetof(TYPE,MEMBER) \_\_builtin\_offsetof (TYPE, MEMBER)

### **14.4.63 stdint.h**

#define INT16\_C(c) c

#define INT32\_C(c) c

#define INT8\_C(c) c

#define UINT16\_C(c) c

#define UINT8\_C(c) c

#define UINT32\_C(c) c ## U

#define INT8\_MIN (-128)

#define INT\_FAST8\_MIN (-128)

#define INT\_LEAST8\_MIN (-128)

#define INT32\_MIN (-2147483647-1)

#define INT\_LEAST32\_MIN (-2147483647-1)

#define SIG\_ATOMIC\_MIN (-2147483647-1)

#define INT16\_MIN (-32767-1)

#define INT\_LEAST16\_MIN (-32767-1)

#define INT64\_MIN (-\_\_INT64\_C(9223372036854775807)-1)

#define INTMAX\_MIN (-\_\_INT64\_C(9223372036854775807)-1)

#define INT\_FAST64\_MIN (-\_\_INT64\_C(9223372036854775807)-1)

#define INT\_LEAST64\_MIN (-\_\_INT64\_C(9223372036854775807)-1)

#define WINT\_MIN (0u)

#define INT8\_MAX (127)

#define INT\_FAST8\_MAX (127)

#define INT\_LEAST8\_MAX (127)

#define INT32\_MAX (2147483647)

#define INT\_LEAST32\_MAX (2147483647)

#define SIG\_ATOMIC\_MAX (2147483647)

#define UINT8\_MAX (255)

#define UINT\_FAST8\_MAX (255)

#define UINT\_LEAST8\_MAX (255)

#define INT16\_MAX (32767)

#define INT\_LEAST16\_MAX (32767)

#define UINT32\_MAX (4294967295U)

#define UINT\_LEAST32\_MAX (4294967295U)

#define WINT\_MAX (4294967295u)

#define UINT16\_MAX (65535)

#define UINT\_LEAST16\_MAX (65535)

#define INT64\_MAX (\_\_INT64\_C(9223372036854775807))

#define INTMAX\_MAX (\_\_INT64\_C(9223372036854775807))

#define INT\_FAST64\_MAX (\_\_INT64\_C(9223372036854775807))

#define INT\_LEAST64\_MAX (\_\_INT64\_C(9223372036854775807))

#define UINT64\_MAX (\_\_UINT64\_C(18446744073709551615))

#define UINTMAX\_MAX (\_\_UINT64\_C(18446744073709551615))

#define UINT\_FAST64\_MAX (\_\_UINT64\_C(18446744073709551615))

#define UINT\_LEAST64\_MAX (\_\_UINT64\_C(18446744073709551615))

typedef signed char int8\_t;

typedef short int16\_t;

typedef int int32\_t;

typedef unsigned char uint8\_t;

typedef unsigned short uint16\_t;

typedef unsigned int uint32\_t;

typedef signed char int\_least8\_t;

typedef short int int\_least16\_t;

typedef int int\_least32\_t;

typedef unsigned char uint\_least8\_t;

typedef unsigned short uint\_least16\_t;

typedef unsigned int uint\_least32\_t;

typedef signed char int\_fast8\_t;

typedef unsigned char uint\_fast8\_t;

### **14.4.64 stdio.h**

#define EOF (-1)

#define P\_tmpdir "/tmp"

#ifndef SEEK\_SET

#define SEEK\_SET 0

#endif

#ifndef SEEK\_CUR

#define SEEK\_CUR 1

#endif

#define FOPEN\_MAX 16

#ifndef SEEK\_END

#define SEEK\_END 2

#endif

#define L\_tmpnam 20

#define TMP\_MAX 238328

#define FILENAME\_MAX 4096

#define BUFSIZ 8192

#define L\_ctermid 9

#define L\_cuserid 9

typedef struct {

off\_t \_\_pos;

mbstate\_t \_\_state;

} fpos\_t;

typedef struct {

off64\_t \_\_pos;

mbstate\_t \_\_state;

} fpos64\_t;

typedef struct \_IO\_FILE FILE;

#define \_IOFBF 0

#define \_IOLBF 1

#define \_IONBF 2

extern char \*\_\_fgets\_chk(char \*, size\_t, int, FILE \*);

extern char \*\_\_fgets\_unlocked\_chk(char \*, size\_t, int, FILE \*);

extern size\_t \_\_fpending(FILE \*);

extern int \_\_fprintf\_chk(FILE \*, int, const char \*, ...);

extern int \_\_printf\_chk(int, const char \*, ...);

extern int \_\_snprintf\_chk(char \*, size\_t, int, size\_t, const char \*, ...);

extern int \_\_sprintf\_chk(char \*, int, size\_t, const char \*, ...);

extern int \_\_vfprintf\_chk(FILE \*, int, const char \*, va\_list);

extern int \_\_vprintf\_chk(int, const char \*, va\_list);

extern int \_\_vsnprintf\_chk(char \*, size\_t, int, size\_t, const char \*,

va\_list);

extern int \_\_vsprintf\_chk(char \*, int, size\_t, const char \*, va\_list);

extern char \*const \_sys\_errlist[];

extern int asprintf(char \*\*\_\_ptr, const char \*\_\_fmt, ...);

extern void clearerr(FILE \* \_\_stream);

extern void clearerr\_unlocked(FILE \* \_\_stream);

extern int dprintf(int \_\_fd, const char \*\_\_fmt, ...);

extern int fclose(FILE \* \_\_stream);

extern FILE \*fdopen(int \_\_fd, const char \*\_\_modes);

extern int feof(FILE \* \_\_stream);

extern int feof\_unlocked(FILE \* \_\_stream);

extern int ferror(FILE \* \_\_stream);

extern int ferror\_unlocked(FILE \* \_\_stream);

extern int fflush(FILE \* \_\_stream);

extern int fflush\_unlocked(FILE \* \_\_stream);

extern int fgetc(FILE \* \_\_stream);

extern int fgetc\_unlocked(FILE \* \_\_stream);

extern int fgetpos(FILE \* \_\_stream, fpos\_t \* \_\_pos);

extern int fgetpos64(FILE \* \_\_stream, fpos64\_t \* \_\_pos);

extern char \*fgets(char \*\_\_s, int \_\_n, FILE \* \_\_stream);

extern char \*fgets\_unlocked(char \*\_\_s, int \_\_n, FILE \* \_\_stream);

extern int fileno(FILE \* \_\_stream);

extern int fileno\_unlocked(FILE \* \_\_stream);

extern void flockfile(FILE \* \_\_stream);

extern FILE \*fmemopen(void \*\_\_s, size\_t \_\_len, const char \*\_\_modes);

extern FILE \*fopen(const char \*\_\_filename, const char \*\_\_modes);

extern FILE \*fopen64(const char \*\_\_filename, const char \*\_\_modes);

extern int fprintf(FILE \* \_\_stream, const char \*\_\_format, ...);

extern int fputc(int \_\_c, FILE \* \_\_stream);

extern int fputc\_unlocked(int \_\_c, FILE \* \_\_stream);

extern int fputs(const char \*\_\_s, FILE \* \_\_stream);

extern int fputs\_unlocked(const char \*\_\_s, FILE \* \_\_stream);

extern size\_t fread(void \*\_\_ptr, size\_t \_\_size, size\_t \_\_n,

FILE \* \_\_stream);

extern size\_t fread\_unlocked(void \*\_\_ptr, size\_t \_\_size, size\_t \_\_n,

FILE \* \_\_stream);

extern FILE \*freopen(const char \*\_\_filename, const char \*\_\_modes,

FILE \* \_\_stream);

extern FILE \*freopen64(const char \*\_\_filename, const char \*\_\_modes,

FILE \* \_\_stream);

extern int fscanf(FILE \* \_\_stream, const char \*\_\_format, ...);

extern int fseek(FILE \* \_\_stream, long int \_\_off, int \_\_whence);

extern int fseeko(FILE \* \_\_stream, off\_t \_\_off, int \_\_whence);

extern int fseeko64(FILE \* \_\_stream, loff\_t \_\_off, int \_\_whence);

extern int fsetpos(FILE \* \_\_stream, const fpos\_t \* \_\_pos);

extern int fsetpos64(FILE \* \_\_stream, const fpos64\_t \* \_\_pos);

extern long int ftell(FILE \* \_\_stream);

extern off\_t ftello(FILE \* \_\_stream);

extern loff\_t ftello64(FILE \* \_\_stream);

extern int ftrylockfile(FILE \* \_\_stream);

extern void funlockfile(FILE \* \_\_stream);

extern size\_t fwrite(const void \*\_\_ptr, size\_t \_\_size, size\_t \_\_n,

FILE \* \_\_s);

extern size\_t fwrite\_unlocked(const void \*\_\_ptr, size\_t \_\_size, size\_t \_\_n,

FILE \* \_\_stream);

extern int getc(FILE \* \_\_stream);

extern int getc\_unlocked(FILE \* \_\_stream);

extern int getchar(void);

extern int getchar\_unlocked(void);

extern ssize\_t getdelim(char \*\*\_\_lineptr, size\_t \* \_\_n, int \_\_delimiter,

FILE \* \_\_stream);

extern ssize\_t getline(char \*\*\_\_lineptr, size\_t \* \_\_n, FILE \* \_\_stream);

extern int getw(FILE \* \_\_stream);

extern FILE \*open\_memstream(char \*\*\_\_bufloc, size\_t \* \_\_sizeloc);

extern int pclose(FILE \* \_\_stream);

extern void perror(const char \*\_\_s);

extern FILE \*popen(const char \*\_\_command, const char \*\_\_modes);

extern int printf(const char \*\_\_format, ...);

extern int putc(int \_\_c, FILE \* \_\_stream);

extern int putc\_unlocked(int \_\_c, FILE \* \_\_stream);

extern int putchar(int \_\_c);

extern int putchar\_unlocked(int \_\_c);

extern int puts(const char \*\_\_s);

extern int putw(int \_\_w, FILE \* \_\_stream);

extern int remove(const char \*\_\_filename);

extern int rename(const char \*\_\_old, const char \*\_\_new);

extern int renameat(int \_\_oldfd, const char \*\_\_old, int \_\_newfd,

const char \*\_\_new);

extern void rewind(FILE \* \_\_stream);

extern int scanf(const char \*\_\_format, ...);

extern void setbuf(FILE \* \_\_stream, char \*\_\_buf);

extern void setbuffer(FILE \* \_\_stream, char \*\_\_buf, size\_t \_\_size);

extern int setvbuf(FILE \* \_\_stream, char \*\_\_buf, int \_\_modes, size\_t \_\_n);

extern int snprintf(char \*\_\_s, size\_t \_\_maxlen, const char \*\_\_format, ...);

extern int sprintf(char \*\_\_s, const char \*\_\_format, ...);

extern int sscanf(const char \*\_\_s, const char \*\_\_format, ...);

extern FILE \*stderr;

extern FILE \*stdin;

extern FILE \*stdout;

extern char \*tempnam(const char \*\_\_dir, const char \*\_\_pfx);

extern FILE \*tmpfile(void);

extern FILE \*tmpfile64(void);

extern char \*tmpnam(char \*\_\_s);

extern int ungetc(int \_\_c, FILE \* \_\_stream);

extern int vasprintf(char \*\*\_\_ptr, const char \*\_\_f, va\_list \_\_arg);

extern int vdprintf(int \_\_fd, const char \*\_\_fmt, va\_list \_\_arg);

extern int vfprintf(FILE \* \_\_s, const char \*\_\_format, va\_list \_\_arg);

extern int vfscanf(FILE \* \_\_s, const char \*\_\_format, va\_list \_\_arg);

extern int vprintf(const char \*\_\_format, va\_list \_\_arg);

extern int vscanf(const char \*\_\_format, va\_list \_\_arg);

extern int vsnprintf(char \*\_\_s, size\_t \_\_maxlen, const char \*\_\_format,

va\_list \_\_arg);

extern int vsprintf(char \*\_\_s, const char \*\_\_format, va\_list \_\_arg);

extern int vsscanf(const char \*\_\_s, const char \*\_\_format, va\_list \_\_arg);

### **14.4.65 stdlib.h**

#define MB\_CUR\_MAX (\_\_ctype\_get\_mb\_cur\_max())

#define EXIT\_SUCCESS 0

#define EXIT\_FAILURE 1

#define RAND\_MAX 2147483647

struct drand48\_data {

unsigned short \_\_x[3];

unsigned short \_\_old\_x[3];

unsigned short \_\_c;

unsigned short \_\_init;

unsigned long long int \_\_a;

};

typedef int (\*\_\_compar\_fn\_t) (const void \*, const void \*);

struct random\_data {

int32\_t \*fptr; /\* Front pointer. \*/

int32\_t \*rptr; /\* Rear pointer. \*/

int32\_t \*state; /\* Array of state values. \*/

int rand\_type; /\* Type of random number generator. \*/

int rand\_deg; /\* Degree of random number generator. \*/

int rand\_sep; /\* Distance between front and rear. \*/

int32\_t \*end\_ptr; /\* Pointer behind state table. \*/

};

typedef struct {

int quot;

int rem;

} div\_t;

typedef struct {

long int quot;

long int rem;

} ldiv\_t;

typedef struct {

long long int quot;

long long int rem;

} lldiv\_t;

extern void \_Exit(int \_\_status);

extern size\_t \_\_ctype\_get\_mb\_cur\_max(void);

extern size\_t \_\_mbstowcs\_chk(wchar\_t \*, const char \*, size\_t, size\_t);

extern char \*\_\_realpath\_chk(const char \*, char \*, size\_t);

extern double \_\_strtod\_internal(const char \*, char \*\*, int);

extern float \_\_strtof\_internal(const char \*, char \*\*, int);

extern long int \_\_strtol\_internal(const char \*, char \*\*, int, int);

extern long double \_\_strtold\_internal(const char \*, char \*\*, int);

extern long long int \_\_strtoll\_internal(const char \*, char \*\*, int, int);

extern unsigned long int \_\_strtoul\_internal(const char \*, char \*\*, int,

int);

extern unsigned long long int \_\_strtoull\_internal(const char \*, char \*\*,

int, int);

extern size\_t \_\_wcstombs\_chk(char \*, const wchar\_t \*, size\_t, size\_t);

extern int \_\_wctomb\_chk(char \*, wchar\_t, size\_t);

extern long int a64l(const char \*\_\_s);

extern void abort(void);

extern int abs(int \_\_x);

extern int atexit(void (\*\_\_func) (void));

extern double atof(const char \*\_\_nptr);

extern int atoi(const char \*\_\_nptr);

extern long int atol(const char \*\_\_nptr);

extern long long int atoll(const char \*\_\_nptr);

extern void \*bsearch(const void \*\_\_key, const void \*\_\_base, size\_t \_\_nmemb,

size\_t \_\_size, \_\_compar\_fn\_t \_\_compar);

extern void \*calloc(size\_t \_\_nmemb, size\_t \_\_size);

extern div\_t div(int \_\_numer, int \_\_denom);

extern double drand48(void);

extern int drand48\_r(struct drand48\_data \*\_\_buffer, double \*\_\_result);

extern char \*ecvt(double \_\_value, int \_\_ndigit, int \*\_\_decpt, int \*\_\_sign);

extern char \*\*environ;

extern double erand48(unsigned short \_\_xsubi[3]);

extern int erand48\_r(unsigned short \_\_xsubi[3],

struct drand48\_data \*\_\_buffer, double \*\_\_result);

extern void exit(int \_\_status);

extern char \*fcvt(double \_\_value, int \_\_ndigit, int \*\_\_decpt, int \*\_\_sign);

extern void free(void \*\_\_ptr);

extern char \*gcvt(double \_\_value, int \_\_ndigit, char \*\_\_buf);

extern char \*getenv(const char \*\_\_name);

extern int getloadavg(double \_\_loadavg[], int \_\_nelem);

extern int getsubopt(char \*\*\_\_optionp, char \*const \*\_\_tokens,

char \*\*\_\_valuep);

extern int grantpt(int \_\_fd);

extern char \*initstate(unsigned int \_\_seed, char \*\_\_statebuf,

size\_t \_\_statelen);

extern int initstate\_r(unsigned int \_\_seed, char \*\_\_statebuf,

size\_t \_\_statelen, struct random\_data \*\_\_buf);

extern long int jrand48(unsigned short \_\_xsubi[3]);

extern int jrand48\_r(unsigned short \_\_xsubi[3],

struct drand48\_data \*\_\_buffer, long int \*\_\_result);

extern char \*l64a(long int \_\_n);

extern long int labs(long int \_\_x);

extern void lcong48(unsigned short \_\_param[7]);

extern int lcong48\_r(unsigned short \_\_param[7],

struct drand48\_data \*\_\_buffer);

extern ldiv\_t ldiv(long int \_\_numer, long int \_\_denom);

extern long long int llabs(long long int \_\_x);

extern lldiv\_t lldiv(long long int \_\_numer, long long int \_\_denom);

extern long int lrand48(void);

extern int lrand48\_r(struct drand48\_data \*\_\_buffer, long int \*\_\_result);

extern void \*malloc(size\_t \_\_size);

extern int mblen(const char \*\_\_s, size\_t \_\_n);

extern size\_t mbstowcs(wchar\_t \* \_\_pwcs, const char \*\_\_s, size\_t \_\_n);

extern int mbtowc(wchar\_t \* \_\_pwc, const char \*\_\_s, size\_t \_\_n);

extern char \*mkdtemp(char \*\_\_template);

extern int mkstemp(char \*\_\_template);

extern int mkstemp64(char \*\_\_template);

extern char \*mktemp(char \*\_\_template);

extern long int mrand48(void);

extern int mrand48\_r(struct drand48\_data \*\_\_buffer, long int \*\_\_result);

extern long int nrand48(unsigned short \_\_xsubi[3]);

extern int nrand48\_r(unsigned short \_\_xsubi[3],

struct drand48\_data \*\_\_buffer, long int \*\_\_result);

extern int posix\_memalign(void \*\*\_\_memptr, size\_t \_\_alignment,

size\_t \_\_size);

extern int posix\_openpt(int \_\_oflag);

extern char \*ptsname(int \_\_fd);

extern int putenv(char \*\_\_string);

extern void qsort(void \*\_\_base, size\_t \_\_nmemb, size\_t \_\_size,

const \_\_compar\_fn\_t \_\_compar);

extern int rand(void);

extern int rand\_r(unsigned int \*\_\_seed);

extern long int random(void);

extern int random\_r(struct random\_data \*\_\_buf, int32\_t \* \_\_result);

extern void \*realloc(void \*\_\_ptr, size\_t \_\_size);

extern char \*realpath(const char \*\_\_name, char \*\_\_resolved);

extern unsigned short \*seed48(unsigned short \_\_seed16v[3]);

extern int seed48\_r(unsigned short \_\_seed16v[3],

struct drand48\_data \*\_\_buffer);

extern int setenv(const char \*\_\_name, const char \*\_\_value, int \_\_replace);

extern char \*setstate(char \*\_\_statebuf);

extern int setstate\_r(char \*\_\_statebuf, struct random\_data \*\_\_buf);

extern void srand(unsigned int \_\_seed);

extern void srand48(long int \_\_seedval);

extern int srand48\_r(long int \_\_seedval, struct drand48\_data \*\_\_buffer);

extern void srandom(unsigned int \_\_seed);

extern int srandom\_r(unsigned int \_\_seed, struct random\_data \*\_\_buf);

extern double strtod(const char \*\_\_nptr, char \*\*\_\_endptr);

extern float strtof(const char \*\_\_nptr, char \*\*\_\_endptr);

extern long int strtol(const char \*\_\_nptr, char \*\*\_\_endptr, int \_\_base);

extern long double strtold(const char \*\_\_nptr, char \*\*\_\_endptr);

extern long long int strtoll(const char \*\_\_nptr, char \*\*\_\_endptr,

int \_\_base);

extern long long int strtoq(const char \*\_\_nptr, char \*\*\_\_endptr,

int \_\_base);

extern unsigned long int strtoul(const char \*\_\_nptr, char \*\*\_\_endptr,

int \_\_base);

extern unsigned long long int strtoull(const char \*\_\_nptr, char \*\*\_\_endptr,

int \_\_base);

extern unsigned long long int strtouq(const char \*\_\_nptr, char \*\*\_\_endptr,

int \_\_base);

extern int system(const char \*\_\_command);

extern int unlockpt(int \_\_fd);

extern int unsetenv(const char \*\_\_name);

extern size\_t wcstombs(char \*\_\_s, const wchar\_t \* \_\_pwcs, size\_t \_\_n);

extern int wctomb(char \*\_\_s, wchar\_t \_\_wchar);

### **14.4.66 string.h**

#define strerror\_r \_\_xpg\_strerror\_r

#define bzero(s,n) memset(s,0,n)

extern void \*\_\_memcpy\_chk(void \*, const void \*, size\_t, size\_t);

extern void \*\_\_memmove\_chk(void \*, const void \*, size\_t, size\_t);

extern void \*\_\_mempcpy(void \*\_\_dest, const void \*\_\_src, size\_t \_\_n);

extern void \*\_\_mempcpy\_chk(void \*, const void \*, size\_t, size\_t);

extern void \*\_\_memset\_chk(void \*, int, size\_t, size\_t);

extern char \*\_\_stpcpy(char \*\_\_dest, const char \*\_\_src);

extern char \*\_\_stpcpy\_chk(char \*, const char \*, size\_t);

extern char \*\_\_stpncpy\_chk(char \*, const char \*, size\_t, size\_t);

extern char \*\_\_strcat\_chk(char \*, const char \*, size\_t);

extern char \*\_\_strcpy\_chk(char \*, const char \*, size\_t);

extern char \*\_\_strncat\_chk(char \*, const char \*, size\_t, size\_t);

extern char \*\_\_strncpy\_chk(char \*, const char \*, size\_t, size\_t);

extern char \*\_\_strtok\_r(char \*\_\_s, const char \*\_\_delim, char \*\*\_\_save\_ptr);

extern int \_\_xpg\_strerror\_r(int, char \*, size\_t);

extern void \*memccpy(void \*\_\_dest, const void \*\_\_src, int \_\_c, size\_t \_\_n);

extern void \*memchr(const void \*\_\_s, int \_\_c, size\_t \_\_n);

extern int memcmp(const void \*\_\_s1, const void \*\_\_s2, size\_t \_\_n);

extern void \*memcpy(void \*\_\_dest, const void \*\_\_src, size\_t \_\_n);

extern void \*memmem(const void \*\_\_haystack, size\_t \_\_haystacklen,

const void \*\_\_needle, size\_t \_\_needlelen);

extern void \*memmove(void \*\_\_dest, const void \*\_\_src, size\_t \_\_n);

extern void \*memrchr(const void \*\_\_s, int \_\_c, size\_t \_\_n);

extern void \*memset(void \*\_\_s, int \_\_c, size\_t \_\_n);

extern char \*stpcpy(char \*\_\_dest, const char \*\_\_src);

extern char \*stpncpy(char \*\_\_dest, const char \*\_\_src, size\_t \_\_n);

extern char \*strcasestr(const char \*\_\_haystack, const char \*\_\_needle);

extern char \*strcat(char \*\_\_dest, const char \*\_\_src);

extern char \*strchr(const char \*\_\_s, int \_\_c);

extern int strcmp(const char \*\_\_s1, const char \*\_\_s2);

extern int strcoll(const char \*\_\_s1, const char \*\_\_s2);

extern int strcoll\_l(const char \*s1, const char \*s2, locale\_t locale);

extern char \*strcpy(char \*\_\_dest, const char \*\_\_src);

extern size\_t strcspn(const char \*\_\_s, const char \*\_\_reject);

extern char \*strdup(const char \*\_\_s);

extern char \*strerror(int \_\_errnum);

extern char \*strerror\_l(int errnum, locale\_t locale);

extern size\_t strlen(const char \*\_\_s);

extern char \*strncat(char \*\_\_dest, const char \*\_\_src, size\_t \_\_n);

extern int strncmp(const char \*\_\_s1, const char \*\_\_s2, size\_t \_\_n);

extern char \*strncpy(char \*\_\_dest, const char \*\_\_src, size\_t \_\_n);

extern char \*strndup(const char \*\_\_string, size\_t \_\_n);

extern size\_t strnlen(const char \*\_\_string, size\_t \_\_maxlen);

extern char \*strpbrk(const char \*\_\_s, const char \*\_\_accept);

extern char \*strrchr(const char \*\_\_s, int \_\_c);

extern char \*strsep(char \*\*\_\_stringp, const char \*\_\_delim);

extern char \*strsignal(int \_\_sig);

extern size\_t strspn(const char \*\_\_s, const char \*\_\_accept);

extern char \*strstr(const char \*\_\_haystack, const char \*\_\_needle);

extern char \*strtok(char \*\_\_s, const char \*\_\_delim);

extern char \*strtok\_r(char \*\_\_s, const char \*\_\_delim, char \*\*\_\_save\_ptr);

extern size\_t strxfrm(char \*\_\_dest, const char \*\_\_src, size\_t \_\_n);

extern size\_t strxfrm\_l(char \*s1, const char \*s2, size\_t n,

locale\_t locale);

### **14.4.67 strings.h**

extern int bcmp(const void \*\_\_s1, const void \*\_\_s2, size\_t \_\_n);

extern void bcopy(const void \*\_\_src, void \*\_\_dest, size\_t \_\_n);

extern void bzero(void \*\_\_s, size\_t \_\_n);

extern int ffs(int \_\_i);

extern char \*index(const char \*\_\_s, int \_\_c);

extern char \*rindex(const char \*\_\_s, int \_\_c);

extern int strcasecmp(const char \*\_\_s1, const char \*\_\_s2);

extern int strcasecmp\_l(const char \*s1, const char \*s2, locale\_t locale);

extern int strncasecmp(const char \*\_\_s1, const char \*\_\_s2, size\_t \_\_n);

extern int strncasecmp\_l(const char \*s1, const char \*s2, size\_t n,

locale\_t locale);

### **14.4.68 sys/epoll.h**

#define EPOLL\_CTL\_ADD 1 /\* Add a file decriptor to the interface. \*/

#define EPOLL\_CTL\_DEL 2 /\* Remove a file decriptor from the interface. \*/

#define EPOLL\_CTL\_MOD 3 /\* Change file decriptor epoll\_event structure. \*/

#define EPOLLIN 1

#define EPOLLPRI 2

#define EPOLLOUT 4

#define EPOLLERR 8

#define EPOLLHUP 16

#define EPOLLRDHUP 0x2000

#define EPOLLONESHOT (1 << 30)

#define EPOLLET (1 << 31)

typedef union epoll\_data {

void \*ptr;

int fd;

uint32\_t u32;

uint64\_t u64;

} epoll\_data\_t;

struct epoll\_event {

uint32\_t events;

epoll\_data\_t data;

};

extern int epoll\_create(int \_\_size);

extern int epoll\_ctl(int \_\_epfd, int \_\_op, int \_\_fd,

struct epoll\_event \*\_\_event);

extern int epoll\_wait(int \_\_epfd, struct epoll\_event \*\_\_events,

int \_\_maxevents, int \_\_timeout);

### **14.4.69 sys/file.h**

#define LOCK\_SH 1

#define LOCK\_EX 2

#define LOCK\_NB 4

#define LOCK\_UN 8

extern int flock(int \_\_fd, int \_\_operation);

### **14.4.70 sys/inotify.h**

#define IN\_ACCESS 0x00000001

#define IN\_MODIFY 0x00000002

#define IN\_ATTRIB 0x00000004

#define IN\_CLOSE\_WRITE 0x00000008

#define IN\_CLOSE\_NOWRITE 0x00000010

#define IN\_OPEN 0x00000020

#define IN\_MOVED\_FROM 0x00000040

#define IN\_MOVED\_TO 0x00000080

#define IN\_CREATE 0x00000100

#define IN\_DELETE 0x00000200

#define IN\_DELETE\_SELF 0x00000400

#define IN\_MOVE\_SELF 0x00000800

#define IN\_UNMOUNT 0x00002000

#define IN\_Q\_OVERFLOW 0x00004000

#define IN\_IGNORED 0x00008000

#define IN\_ISDIR 0x40000000

#define IN\_ONESHOT 0x80000000

#define IN\_CLOSE (IN\_CLOSE\_WRITE | IN\_CLOSE\_NOWRITE)

#define IN\_MOVE (IN\_MOVED\_FROM | IN\_MOVED\_TO)

#define IN\_ALL\_EVENTS \

(IN\_ACCESS | IN\_MODIFY | IN\_ATTRIB | IN\_CLOSE\_WRITE | \

IN\_CLOSE\_NOWRITE | IN\_OPEN | IN\_MOVED\_FROM | IN\_MOVED\_TO | IN\_CREATE | \

IN\_DELETE | IN\_DELETE\_SELF | IN\_MOVE\_SELF)

struct inotify\_event {

int wd;

uint32\_t mask;

uint32\_t cookie;

uint32\_t len;

char name[];

};

extern int inotify\_add\_watch(int \_\_fd, const char \*\_\_name,

uint32\_t \_\_mask);

extern int inotify\_init(void);

extern int inotify\_rm\_watch(int \_\_fd, int \_\_wd);

### **14.4.71 sys/ioctl.h**

#define \_IOC(dir,type,nr,size) (((dir) << \_IOC\_DIRSHIFT) | ((type) << \_IOC\_TYPESHIFT) | ((nr) << \_IOC\_NRSHIFT) | ((size) << \_IOC\_SIZESHIFT))

#define \_IOC\_DIR(nr) (((nr) >> \_IOC\_DIRSHIFT) & \_IOC\_DIRMASK)

#define \_IOC\_NR(nr) (((nr) >> \_IOC\_NRSHIFT) & \_IOC\_NRMASK)

#define \_IOC\_SIZE(nr) (((nr) >> \_IOC\_SIZESHIFT) & \_IOC\_SIZEMASK)

#define \_IOC\_TYPE(nr) (((nr) >> \_IOC\_TYPESHIFT) & \_IOC\_TYPEMASK)

#define \_IOC\_DIRMASK ((1 << \_IOC\_DIRBITS)-1)

#define \_IOC\_NRMASK ((1 << \_IOC\_NRBITS)-1)

#define \_IOC\_SIZEMASK ((1 << \_IOC\_SIZEBITS)-1)

#define \_IOC\_TYPEMASK ((1 << \_IOC\_TYPEBITS)-1)

#define IOC\_INOUT ((\_IOC\_WRITE|\_IOC\_READ) << \_IOC\_DIRSHIFT)

#define \_IOC\_TYPECHECK(t) (sizeof(t))

#define \_IOC\_TYPESHIFT (\_IOC\_NRSHIFT+\_IOC\_NRBITS)

#define IOC\_OUT (\_IOC\_READ << \_IOC\_DIRSHIFT)

#define IOCSIZE\_MASK (\_IOC\_SIZEMASK << \_IOC\_SIZESHIFT)

#define IOCSIZE\_SHIFT (\_IOC\_SIZESHIFT)

#define \_IOC\_DIRSHIFT (\_IOC\_SIZESHIFT+\_IOC\_SIZEBITS)

#define \_IOC\_SIZESHIFT (\_IOC\_TYPESHIFT+\_IOC\_TYPEBITS)

#define IOC\_IN (\_IOC\_WRITE << \_IOC\_DIRSHIFT)

#define \_IOC\_NRSHIFT 0

#define \_IOC\_NONE 0U

#define \_IOC\_SIZEBITS 14

#define \_IOC\_WRITE 1U

#define \_IOC\_DIRBITS 2

#define \_IOC\_READ 2U

#define \_IOC\_NRBITS 8

#define \_IOC\_TYPEBITS 8

#define \_IO(type,nr) \_IOC(\_IOC\_NONE,(type),(nr),0)

#define \_IOR(type,nr,size) \_IOC(\_IOC\_READ,(type),(nr),(\_IOC\_TYPECHECK(size)))

#define \_IOR\_BAD(type,nr,size) \_IOC(\_IOC\_READ,(type),(nr),sizeof(size))

#define \_IOWR(type,nr,size) \_IOC(\_IOC\_READ|\_IOC\_WRITE,(type),(nr),(\_IOC\_TYPECHECK(size)))

#define \_IOWR\_BAD(type,nr,size) \_IOC(\_IOC\_READ|\_IOC\_WRITE,(type),(nr),sizeof(size))

#define \_IOW(type,nr,size) \_IOC(\_IOC\_WRITE,(type),(nr),(\_IOC\_TYPECHECK(size)))

#define \_IOW\_BAD(type,nr,size) \_IOC(\_IOC\_WRITE,(type),(nr),sizeof(size))

struct winsize {

unsigned short ws\_row; /\* Rows, in characters. \*/

unsigned short ws\_col; /\* Columns, in characters. \*/

unsigned short ws\_xpixel; /\* Horizontal pixels. \*/

unsigned short ws\_ypixel; /\* Vertical pixels. \*/

};

extern int ioctl(int \_\_fd, unsigned long int \_\_request, ...);

### **14.4.72 sys/ipc.h**

#define IPC\_PRIVATE ((key\_t)0)

#define IPC\_RMID 0

#define IPC\_CREAT 00001000

#define IPC\_EXCL 00002000

#define IPC\_NOWAIT 00004000

#define IPC\_SET 1

#define IPC\_STAT 2

extern key\_t ftok(const char \*\_\_pathname, int \_\_proj\_id);

### **14.4.73 sys/mman.h**

#define MAP\_FAILED ((void\*)-1)

#define POSIX\_MADV\_NORMAL 0

#define PROT\_NONE 0x0

#define MAP\_SHARED 0x01

#define MAP\_PRIVATE 0x02

#define PROT\_READ 0x1

#define MAP\_FIXED 0x10

#define PROT\_WRITE 0x2

#define MAP\_ANONYMOUS 0x20

#define PROT\_EXEC 0x4

#define MREMAP\_MAYMOVE 1

#define MS\_ASYNC 1

#define POSIX\_MADV\_RANDOM 1

#define MREMAP\_FIXED 2

#define MS\_INVALIDATE 2

#define POSIX\_MADV\_SEQUENTIAL 2

#define POSIX\_MADV\_WILLNEED 3

#define MS\_SYNC 4

#define POSIX\_MADV\_DONTNEED 4

#define MAP\_ANON MAP\_ANONYMOUS

extern int mlock(const void \*\_\_addr, size\_t \_\_len);

extern int mlockall(int \_\_flags);

extern void \*mmap(void \*\_\_addr, size\_t \_\_len, int \_\_prot, int \_\_flags,

int \_\_fd, off\_t \_\_offset);

extern void \*mmap64(void \*\_\_addr, size\_t \_\_len, int \_\_prot, int \_\_flags,

int \_\_fd, off64\_t \_\_offset);

extern int mprotect(void \*\_\_addr, size\_t \_\_len, int \_\_prot);

extern void \*mremap(void \*\_\_addr, size\_t \_\_old\_len, size\_t \_\_new\_len,

int \_\_flags, ...);

extern int msync(void \*\_\_addr, size\_t \_\_len, int \_\_flags);

extern int munlock(const void \*\_\_addr, size\_t \_\_len);

extern int munlockall(void);

extern int munmap(void \*\_\_addr, size\_t \_\_len);

extern int posix\_madvise(void \*\_\_addr, size\_t \_\_len, int \_\_advice);

extern int shm\_open(const char \*\_\_name, int \_\_oflag, mode\_t \_\_mode);

extern int shm\_unlink(const char \*\_\_name);

### **14.4.74 sys/msg.h**

#define MSG\_NOERROR 010000

extern int msgctl(int \_\_msqid, int \_\_cmd, struct msqid\_ds \*\_\_buf);

extern int msgget(key\_t \_\_key, int \_\_msgflg);

extern ssize\_t msgrcv(int \_\_msqid, void \*\_\_msgp, size\_t \_\_msgsz,

long int \_\_msgtyp, int \_\_msgflg);

extern int msgsnd(int \_\_msqid, const void \*\_\_msgp, size\_t \_\_msgsz,

int \_\_msgflg);

### **14.4.75 sys/param.h**

#define NOFILE 256

#define MAXPATHLEN 4096

### **14.4.76 sys/poll.h**

#define POLLIN 0x0001 /\* There is data to read \*/

#define POLLPRI 0x0002 /\* There is urgent data to read \*/

#define POLLOUT 0x0004 /\* Writing now will not block \*/

#define POLLERR 0x0008 /\* Error condition \*/

#define POLLHUP 0x0010 /\* Hung up \*/

#define POLLNVAL 0x0020 /\* Invalid request: fd not open \*/

#define POLLRDNORM 0x0040 /\* Normal data may be read \*/

#define POLLRDBAND 0x0080 /\* Priority data may be read \*/

#define POLLWRNORM 0x0100 /\* Writing now will not block \*/

#define POLLWRBAND 0x0200 /\* Priority data may be written \*/

struct pollfd {

int fd; /\* File descriptor to poll. \*/

short events; /\* Types of events poller cares about. \*/

short revents; /\* Types of events that actually occurred. \*/

};

typedef unsigned long int nfds\_t;

### **14.4.77 sys/ptrace.h**

enum \_\_ptrace\_setoptions {

PTRACE\_O\_TRACESYSGOOD = 0x00000001,

PTRACE\_O\_TRACEFORK = 0x00000002,

PTRACE\_O\_TRACEVFORK = 0x00000004,

PTRACE\_O\_TRACECLONE = 0x00000008,

PTRACE\_O\_TRACEEXEC = 0x00000010,

PTRACE\_O\_TRACEVFORKDONE = 0x00000020,

PTRACE\_O\_TRACEEXIT = 0x00000040,

PTRACE\_O\_MASK = 0x0000007f

};

enum \_\_ptrace\_eventcodes {

PTRACE\_EVENT\_FORK = 1,

PTRACE\_EVENT\_VFORK = 2,

PTRACE\_EVENT\_CLONE = 3,

PTRACE\_EVENT\_EXEC = 4,

PTRACE\_EVENT\_VFORK\_DONE = 5,

PTRACE\_EVENT\_EXIT = 6

};

extern long int ptrace(enum \_\_ptrace\_request, ...);

### **14.4.78 sys/resource.h**

#define RUSAGE\_CHILDREN (-1)

#define RLIM\_INFINITY (~0UL)

#define RLIM\_SAVED\_CUR -1

#define RLIM\_SAVED\_MAX -1

#define RLIMIT\_CPU 0

#define RUSAGE\_SELF 0

#define RLIMIT\_FSIZE 1

#define RLIMIT\_LOCKS 10

#define RLIMIT\_SIGPENDING 11

#define RLIMIT\_MSGQUEUE 12

#define RLIMIT\_NICE 13

#define RLIMIT\_RTPRIO 14

#define RLIMIT\_RTTIME 15

#define RLIM\_NLIMITS 16

#define RLIMIT\_DATA 2

#define RLIMIT\_STACK 3

#define RLIMIT\_CORE 4

#define RLIMIT\_RSS 5

#define RLIMIT\_NPROC 6

#define RLIMIT\_NOFILE 7

#define RLIMIT\_MEMLOCK 8

#define RLIMIT\_AS 9

typedef unsigned long int rlim\_t;

typedef unsigned long long int rlim64\_t;

typedef int \_\_rlimit\_resource\_t;

struct rlimit {

rlim\_t rlim\_cur; /\* The current (soft) limit. \*/

rlim\_t rlim\_max; /\* The hard limit. \*/

};

struct rlimit64 {

rlim64\_t rlim\_cur; /\* The current (soft) limit. \*/

rlim64\_t rlim\_max; /\* The hard limit. \*/

};

struct rusage {

struct timeval ru\_utime; /\* Total amount of user time used. \*/

struct timeval ru\_stime; /\* Total amount of system time used. \*/

long int ru\_maxrss; /\* Maximum resident set size (in kilobytes). \*/

long int ru\_ixrss; /\* Amount of sharing of text segment memory with other p \*/

long int ru\_idrss; /\* Amount of data segment memory used (kilobyte-seconds). \*/

long int ru\_isrss; /\* Amount of stack memory used (kilobyte-seconds). \*/

long int ru\_minflt; /\* Number of soft page faults (i.e. those serviced by reclaimin \*/

long int ru\_majflt; /\* Number of hard page faults (i.e. those that required I/O). \*/

long int ru\_nswap; /\* Number of times a process was swapped out of physical memory \*/

long int ru\_inblock; /\* Number of input operations via the file system. Note: This \*/

long int ru\_oublock; /\* Number of output operations via the file system. \*/

long int ru\_msgsnd; /\* Number of IPC messages sent. \*/

long int ru\_msgrcv; /\* Number of IPC messages received. \*/

long int ru\_nsignals; /\* Number of signals delivered. \*/

long int ru\_nvcsw; /\* Number of voluntary context switches, i.e. because the proce \*/

long int ru\_nivcsw; /\* Number of involuntary context switches, i.e. a higher priori \*/

};

enum \_\_priority\_which {

PRIO\_PROCESS = 0, /\* WHO is a process ID. \*/

PRIO\_PGRP = 1, /\* WHO is a process group ID. \*/

PRIO\_USER = 2 /\* WHO is a user ID. \*/

};

#define PRIO\_PGRP PRIO\_PGRP

#define PRIO\_PROCESS PRIO\_PROCESS

#define PRIO\_USER PRIO\_USER

typedef enum \_\_priority\_which \_\_priority\_which\_t;

extern int getpriority(\_\_priority\_which\_t \_\_which, id\_t \_\_who);

extern int getrlimit(\_\_rlimit\_resource\_t \_\_resource,

struct rlimit \*\_\_rlimits);

extern int getrlimit64(\_\_rlimit\_resource\_t \_\_resource,

struct rlimit64 \*\_\_rlimits);

extern int getrusage(int \_\_who, struct rusage \*\_\_usage);

extern int setpriority(\_\_priority\_which\_t \_\_which, id\_t \_\_who, int \_\_prio);

extern int setrlimit(\_\_rlimit\_resource\_t \_\_resource,

const struct rlimit \*\_\_rlimits);

extern int setrlimit64(\_\_rlimit\_resource\_t \_\_resource,

const struct rlimit64 \*\_\_rlimits);

### **14.4.79 sys/select.h**

#define FD\_ISSET(d,set) (((set)->fds\_bits[((d)/(8\*sizeof(long)))]&(1L<<((d)%(8\*sizeof(long)))))!=0)

#define FD\_CLR(d,set) ((set)->fds\_bits[((d)/(8\*sizeof(long)))]&=~(1L<<((d)%(8\*sizeof(long)))))

#define FD\_SET(d,set) ((set)->fds\_bits[((d)/(8\*sizeof(long)))]|=(1L<<((d)%(8\*sizeof(long)))))

#define NFDBITS (8 \* sizeof (long))

#define FD\_SETSIZE 1024

#define FD\_ZERO(fdsetp) bzero(fdsetp, sizeof(\*(fdsetp)))

typedef struct {

unsigned long int fds\_bits[FD\_SETSIZE / NFDBITS];

} fd\_set;

extern int pselect(int \_\_nfds, fd\_set \* \_\_readfds, fd\_set \* \_\_writefds,

fd\_set \* \_\_exceptfds, const struct timespec \*\_\_timeout,

const sigset\_t \* \_\_sigmask);

extern int select(int \_\_nfds, fd\_set \* \_\_readfds, fd\_set \* \_\_writefds,

fd\_set \* \_\_exceptfds, struct timeval \*\_\_timeout);

### **14.4.80 sys/sem.h**

#define SEM\_UNDO 0x1000

#define GETPID 11

#define GETVAL 12

#define GETALL 13

#define GETNCNT 14

#define GETZCNT 15

#define SETVAL 16

#define SETALL 17

struct sembuf {

short sem\_num;

short sem\_op;

short sem\_flg;

};

extern int semctl(int \_\_semid, int \_\_semnum, int \_\_cmd, ...);

extern int semget(key\_t \_\_key, int \_\_nsems, int \_\_semflg);

extern int semop(int \_\_semid, struct sembuf \*\_\_sops, size\_t \_\_nsops);

### **14.4.81 sys/sendfile.h**

extern ssize\_t sendfile(int \_\_out\_fd, int \_\_in\_fd, off\_t \* \_\_offset,

size\_t \_\_count);

extern ssize\_t sendfile64(int \_\_out\_fd, int \_\_in\_fd, off64\_t \* \_\_offset,

size\_t \_\_count);

### **14.4.82 sys/shm.h**

#define SHM\_RDONLY 010000

#define SHM\_W 0200

#define SHM\_RND 020000

#define SHM\_R 0400

#define SHM\_REMAP 040000

#define SHM\_LOCK 11

#define SHM\_UNLOCK 12

extern int \_\_getpagesize(void);

extern void \*shmat(int \_\_shmid, const void \*\_\_shmaddr, int \_\_shmflg);

extern int shmctl(int \_\_shmid, int \_\_cmd, struct shmid\_ds \*\_\_buf);

extern int shmdt(const void \*\_\_shmaddr);

extern int shmget(key\_t \_\_key, size\_t \_\_size, int \_\_shmflg);

### **14.4.83 sys/socket.h**

#define CMSG\_FIRSTHDR(msg) ((size\_t) (msg)->msg\_controllen >= sizeof(struct cmsghdr) ? (struct cmsghdr \*)(msg)->msg\_control : (struct cmsghdr \*) NULL)

#define CMSG\_LEN(len) (CMSG\_ALIGN(sizeof(struct cmsghdr))+(len))

#define SCM\_RIGHTS 0x01

#define SOL\_SOCKET 1

#define SOMAXCONN 128

#define SOL\_RAW 255

#define CMSG\_ALIGN(len) \

(((len)+sizeof(size\_t)-1)&(size\_t)~(sizeof(size\_t)-1))

#define CMSG\_DATA(cmsg) \

((unsigned char \*) (cmsg) + CMSG\_ALIGN(sizeof(struct cmsghdr)))

#define CMSG\_SPACE(len) \

(CMSG\_ALIGN(sizeof(struct cmsghdr))+CMSG\_ALIGN(len))

#define CMSG\_NXTHDR(mhdr,cmsg) \

(((cmsg) == NULL) ? CMSG\_FIRSTHDR(mhdr) : \

(((u\_char \*)(cmsg) + CMSG\_ALIGN((cmsg)->cmsg\_len) \

+ CMSG\_ALIGN(sizeof(struct cmsghdr)) > \

(u\_char \*)((mhdr)->msg\_control) + (mhdr)->msg\_controllen) ? \

(struct cmsghdr \*)NULL : \

(struct cmsghdr \*)((u\_char \*)(cmsg) + CMSG\_ALIGN((cmsg)->cmsg\_len))))

struct linger {

int l\_onoff;

int l\_linger;

};

struct cmsghdr {

size\_t cmsg\_len;

int cmsg\_level;

int cmsg\_type;

};

struct iovec {

void \*iov\_base;

size\_t iov\_len;

};

typedef unsigned short sa\_family\_t;

typedef unsigned int socklen\_t;

struct sockaddr {

sa\_family\_t sa\_family;

char sa\_data[14];

};

struct sockaddr\_storage {

sa\_family\_t ss\_family;

\_\_ss\_aligntype \_\_ss\_align;

char \_\_ss\_padding[(128 - (2 \* sizeof(\_\_ss\_aligntype)))];

};

struct msghdr {

void \*msg\_name;

int msg\_namelen;

struct iovec \*msg\_iov;

size\_t msg\_iovlen;

void \*msg\_control;

size\_t msg\_controllen;

unsigned int msg\_flags;

};

#define AF\_UNSPEC 0

#define AF\_UNIX 1

#define AF\_INET6 10

#define AF\_INET 2

#define PF\_INET AF\_INET

#define PF\_INET6 AF\_INET6

#define PF\_UNIX AF\_UNIX

#define PF\_UNSPEC AF\_UNSPEC

#define SOCK\_STREAM 1

#define SOCK\_PACKET 10

#define SOCK\_DGRAM 2

#define SOCK\_RAW 3

#define SOCK\_RDM 4

#define SOCK\_SEQPACKET 5

#define SO\_DEBUG 1

#define SO\_OOBINLINE 10

#define SO\_NO\_CHECK 11

#define SO\_PRIORITY 12

#define SO\_LINGER 13

#define SO\_BSDCOMPAT 14

#define SO\_REUSEADDR 2

#define SO\_TYPE 3

#define SO\_ACCEPTCONN 30

#define SO\_ERROR 4

#define SO\_DONTROUTE 5

#define SO\_BROADCAST 6

#define SO\_SNDBUF 7

#define SO\_RCVBUF 8

#define SO\_KEEPALIVE 9

#define SIOCGIFNAME 0x8910

#define SIOCGIFCONF 0x8912

#define SIOCGIFFLAGS 0x8913

#define SIOCGIFADDR 0x8915

#define SIOCGIFDSTADDR 0x8917

#define SIOCGIFBRDADDR 0x8919

#define SIOCGIFNETMASK 0x891b

#define SIOCGIFMTU 0x8921

#define SIOCGIFHWADDR 0x8927

#define SHUT\_RD 0

#define SHUT\_WR 1

#define SHUT\_RDWR 2

#define MSG\_WAITALL 0x100

#define MSG\_TRUNC 0x20

#define MSG\_NOSIGNAL 0x4000

#define MSG\_EOR 0x80

#define MSG\_OOB 1

#define MSG\_PEEK 2

#define MSG\_DONTROUTE 4

#define MSG\_CTRUNC 8

extern ssize\_t \_\_recv\_chk(int, void \*, size\_t, size\_t, int);

extern ssize\_t \_\_recvfrom\_chk(int, void \*, size\_t, size\_t, int,

struct sockaddr \*, socklen\_t \*);

extern int accept(int \_\_fd, struct sockaddr \*\_\_addr,

socklen\_t \* \_\_addr\_len);

extern int bind(int \_\_fd, const struct sockaddr \*\_\_addr, socklen\_t \_\_len);

extern int connect(int \_\_fd, const struct sockaddr \*\_\_addr,

socklen\_t \_\_len);

extern int getnameinfo(const struct sockaddr \*\_\_sa, socklen\_t \_\_salen,

char \*\_\_host, socklen\_t \_\_hostlen, char \*\_\_serv,

socklen\_t \_\_servlen, unsigned int \_\_flags);

extern int getpeername(int \_\_fd, struct sockaddr \*\_\_addr,

socklen\_t \* \_\_len);

extern int getsockname(int \_\_fd, struct sockaddr \*\_\_addr,

socklen\_t \* \_\_len);

extern int getsockopt(int \_\_fd, int \_\_level, int \_\_optname, void \*\_\_optval,

socklen\_t \* \_\_optlen);

extern int listen(int \_\_fd, int \_\_n);

extern ssize\_t recv(int \_\_fd, void \*\_\_buf, size\_t \_\_n, int \_\_flags);

extern ssize\_t recvfrom(int \_\_fd, void \*\_\_buf, size\_t \_\_n, int \_\_flags,

struct sockaddr \*\_\_addr, socklen\_t \* \_\_addr\_len);

extern ssize\_t recvmsg(int \_\_fd, struct msghdr \*\_\_message, int \_\_flags);

extern ssize\_t send(int \_\_fd, const void \*\_\_buf, size\_t \_\_n, int \_\_flags);

extern ssize\_t sendmsg(int \_\_fd, const struct msghdr \*\_\_message,

int \_\_flags);

extern ssize\_t sendto(int \_\_fd, const void \*\_\_buf, size\_t \_\_n, int \_\_flags,

const struct sockaddr \*\_\_addr, socklen\_t \_\_addr\_len);

extern int setsockopt(int \_\_fd, int \_\_level, int \_\_optname,

const void \*\_\_optval, socklen\_t \_\_optlen);

extern int shutdown(int \_\_fd, int \_\_how);

extern int sockatmark(int \_\_fd);

extern int socket(int \_\_domain, int \_\_type, int \_\_protocol);

extern int socketpair(int \_\_domain, int \_\_type, int \_\_protocol,

int \_\_fds[2]);

### **14.4.84 sys/stat.h**

#define S\_ISBLK(m) (((m)&S\_IFMT)==S\_IFBLK)

#define S\_ISCHR(m) (((m)&S\_IFMT)==S\_IFCHR)

#define S\_ISDIR(m) (((m)&S\_IFMT)==S\_IFDIR)

#define S\_ISFIFO(m) (((m)&S\_IFMT)==S\_IFIFO)

#define S\_ISLNK(m) (((m)&S\_IFMT)==S\_IFLNK)

#define S\_ISREG(m) (((m)&S\_IFMT)==S\_IFREG)

#define S\_ISSOCK(m) (((m)&S\_IFMT)==S\_IFSOCK)

#define UTIME\_NOW ((1l << 30) - 1l)

#define UTIME\_OMIT ((1l << 30) - 2l)

#define S\_TYPEISMQ(buf) ((buf)->st\_mode - (buf)->st\_mode)

#define S\_TYPEISSEM(buf) ((buf)->st\_mode - (buf)->st\_mode)

#define S\_TYPEISSHM(buf) ((buf)->st\_mode - (buf)->st\_mode)

#define S\_IRWXU (S\_IREAD|S\_IWRITE|S\_IEXEC)

#define S\_IROTH (S\_IRGRP>>3)

#define S\_IRGRP (S\_IRUSR>>3)

#define S\_IRWXO (S\_IRWXG>>3)

#define S\_IRWXG (S\_IRWXU>>3)

#define S\_IWOTH (S\_IWGRP>>3)

#define S\_IWGRP (S\_IWUSR>>3)

#define S\_IXOTH (S\_IXGRP>>3)

#define S\_IXGRP (S\_IXUSR>>3)

#define S\_ISVTX 01000

#define S\_IXUSR 0x0040

#define S\_IWUSR 0x0080

#define S\_IRUSR 0x0100

#define S\_ISGID 0x0400

#define S\_ISUID 0x0800

#define S\_IFIFO 0x1000

#define S\_IFCHR 0x2000

#define S\_IFDIR 0x4000

#define S\_IFBLK 0x6000

#define S\_IFREG 0x8000

#define S\_IFLNK 0xa000

#define S\_IFSOCK 0xc000

#define S\_IFMT 0xf000

#define st\_atime st\_atim.tv\_sec

#define st\_ctime st\_ctim.tv\_sec

#define st\_mtime st\_mtim.tv\_sec

#define S\_IREAD S\_IRUSR

#define S\_IWRITE S\_IWUSR

#define S\_IEXEC S\_IXUSR

extern int \_\_fxstat(int \_\_ver, int \_\_fildes, struct stat \*\_\_stat\_buf);

extern int \_\_fxstat64(int \_\_ver, int \_\_fildes, struct stat64 \*\_\_stat\_buf);

extern int \_\_fxstatat(int \_\_ver, int \_\_fildes, const char \*\_\_filename,

struct stat \*\_\_stat\_buf, int \_\_flag);

extern int \_\_fxstatat64(int \_\_ver, int \_\_fildes, const char \*\_\_filename,

struct stat64 \*\_\_stat\_buf, int \_\_flag);

extern int \_\_lxstat(int \_\_ver, const char \*\_\_filename,

struct stat \*\_\_stat\_buf);

extern int \_\_lxstat64(int \_\_ver, const char \*\_\_filename,

struct stat64 \*\_\_stat\_buf);

extern int \_\_xmknod(int \_\_ver, const char \*\_\_path, mode\_t \_\_mode,

dev\_t \* \_\_dev);

extern int \_\_xmknodat(int \_\_ver, int \_\_fd, const char \*\_\_path,

mode\_t \_\_mode, dev\_t \* \_\_dev);

extern int \_\_xstat(int \_\_ver, const char \*\_\_filename,

struct stat \*\_\_stat\_buf);

extern int \_\_xstat64(int \_\_ver, const char \*\_\_filename,

struct stat64 \*\_\_stat\_buf);

extern int chmod(const char \*\_\_file, mode\_t \_\_mode);

extern int fchmod(int \_\_fd, mode\_t \_\_mode);

extern int fchmodat(int \_\_fd, const char \*\_\_file, mode\_t mode, int \_\_flag);

extern int fstat(int \_\_fd, struct stat \*\_\_buf);

extern int fstat64(int \_\_fd, struct stat64 \*\_\_buf);

extern int fstatat(int \_\_fd, const char \*\_\_file, struct stat \*\_\_buf,

int \_\_flag);

extern int fstatat64(int \_\_fd, const char \*\_\_file, struct stat64 \*\_\_buf,

int \_\_flag);

extern int futimens(int fd, const struct timespec times[2]);

extern int lstat(const char \*\_\_file, struct stat \*\_\_buf);

extern int lstat64(const char \*\_\_file, struct stat64 \*\_\_buf);

extern int mkdir(const char \*\_\_path, mode\_t \_\_mode);

extern int mkdirat(int \_\_fd, const char \*\_\_path, mode\_t \_\_mode);

extern int mkfifo(const char \*\_\_path, mode\_t \_\_mode);

extern int mkfifoat(int \_\_fd, const char \*\_\_path, mode\_t \_\_mode);

extern int mknod(const char \*\_\_path, mode\_t \_\_mode, dev\_t \_\_dev);

extern int mknodat(int \_\_fd, const char \*\_\_path, mode\_t \_\_mode,

dev\_t \_\_dev);

extern int stat(const char \*\_\_file, struct stat \*\_\_buf);

extern int stat64(const char \*\_\_file, struct stat64 \*\_\_buf);

extern mode\_t umask(mode\_t \_\_mask);

extern int utimensat(int fd, const char \*path,

const struct timespec times[2], int flags);

### **14.4.85 sys/statfs.h**

#define NFS\_SUPER\_MAGIC 0x6969

extern int fstatfs(int \_\_fildes, struct statfs \*\_\_buf);

extern int fstatfs64(int \_\_fildes, struct statfs64 \*\_\_buf);

extern int statfs(const char \*\_\_file, struct statfs \*\_\_buf);

extern int statfs64(const char \*\_\_file, struct statfs64 \*\_\_buf);

### **14.4.86 sys/statvfs.h**

extern int fstatvfs(int \_\_fildes, struct statvfs \*\_\_buf);

extern int fstatvfs64(int \_\_fildes, struct statvfs64 \*\_\_buf);

extern int statvfs(const char \*\_\_file, struct statvfs \*\_\_buf);

extern int statvfs64(const char \*\_\_file, struct statvfs64 \*\_\_buf);

### **14.4.87 sys/sysinfo.h**

struct sysinfo {

long int uptime; /\* Seconds since boot \*/

unsigned long int loads[3]; /\* 1, 5, and 15 minute load averages \*/

unsigned long int totalram; /\* Total usable main memory size \*/

unsigned long int freeram; /\* Available memory size \*/

unsigned long int sharedram; /\* Amount of shared memory \*/

unsigned long int bufferram; /\* Memory used by buffers \*/

unsigned long int totalswap; /\* Total swap space size \*/

unsigned long int freeswap; /\* Swap space still available \*/

unsigned short procs; /\* Number of current processes \*/

unsigned short pad; /\* Padding for m68k \*/

unsigned long int totalhigh; /\* Total high memory size \*/

unsigned long int freehigh; /\* Available high memory size \*/

unsigned int mem\_unit; /\* Memory unit size in bytes \*/

char \_f[20 - 2 \* sizeof(long) - sizeof(int)]; /\* Padding for libc5 \*/

};

extern int sysinfo(struct sysinfo \*info);

### **14.4.88 sys/time.h**

#define ITIMER\_REAL 0

#define ITIMER\_VIRTUAL 1

#define ITIMER\_PROF 2

struct timezone {

int tz\_minuteswest;

int tz\_dsttime;

};

typedef int \_\_itimer\_which\_t;

struct itimerval {

struct timeval it\_interval;

struct timeval it\_value;

};

extern int adjtime(const struct timeval \*\_\_delta,

struct timeval \*\_\_olddelta);

extern int futimes(int fd, const struct timeval tv[2]);

extern int getitimer(\_\_itimer\_which\_t \_\_which, struct itimerval \*\_\_value);

extern int gettimeofday(struct timeval \*\_\_tv, struct timezone \*\_\_tz);

extern int lutimes(const char \*filename, const struct timeval tv[2]);

extern int setitimer(\_\_itimer\_which\_t \_\_which,

const struct itimerval \*\_\_new,

struct itimerval \*\_\_old);

extern int utimes(const char \*\_\_file, const struct timeval \*\_\_tvp);

### **14.4.89 sys/timeb.h**

struct timeb {

time\_t time; /\* Seconds since epoch, as from time. \*/

unsigned short millitm; /\* Additional milliseconds. \*/

short timezone; /\* Minutes west of GMT. \*/

short dstflag; /\* Nonzero if Daylight Savings Time used. \*/

};

extern int ftime(struct timeb \*\_\_timebuf);

### **14.4.90 sys/times.h**

struct tms {

clock\_t tms\_utime;

clock\_t tms\_stime;

clock\_t tms\_cutime;

clock\_t tms\_cstime;

};

extern clock\_t times(struct tms \*\_\_buffer);

### **14.4.91 sys/types.h**

#ifndef FALSE

#define FALSE 0

#endif

#ifndef TRUE

#define TRUE 1

#endif

typedef unsigned char u\_int8\_t;

typedef unsigned short u\_int16\_t;

typedef unsigned int u\_int32\_t;

typedef unsigned long long int u\_int64\_t;

typedef unsigned int uid\_t;

typedef int pid\_t;

typedef long int off\_t;

typedef long long int off64\_t;

typedef int key\_t;

typedef long int suseconds\_t;

typedef unsigned int u\_int;

typedef struct {

int \_\_val[2];

} fsid\_t;

typedef unsigned int useconds\_t;

typedef long int blksize\_t;

typedef long int fd\_mask;

typedef void \*timer\_t;

typedef int clockid\_t;

typedef unsigned int id\_t;

typedef unsigned long long int ino64\_t;

typedef long long int loff\_t;

typedef long int blkcnt\_t;

typedef unsigned long int fsblkcnt\_t;

typedef unsigned long int fsfilcnt\_t;

typedef long long int blkcnt64\_t;

typedef unsigned long long int fsblkcnt64\_t;

typedef unsigned long long int fsfilcnt64\_t;

typedef unsigned char u\_char;

typedef unsigned short u\_short;

typedef unsigned long int u\_long;

typedef unsigned long int ino\_t;

typedef unsigned int gid\_t;

typedef unsigned long long int dev\_t;

typedef unsigned int mode\_t;

typedef unsigned long int nlink\_t;

typedef char \*caddr\_t;

typedef long int clock\_t;

typedef long int time\_t;

### **14.4.92 sys/uio.h**

extern ssize\_t readv(int \_\_fd, const struct iovec \*\_\_iovec, int \_\_count);

extern ssize\_t writev(int \_\_fd, const struct iovec \*\_\_iovec, int \_\_count);

### **14.4.93 sys/un.h**

struct sockaddr\_un {

sa\_family\_t sun\_family; /\* AF\_UNIX \*/

char sun\_path[108];

};

### **14.4.94 sys/utsname.h**

#define SYS\_NMLN 65

struct utsname {

char sysname[65];

char nodename[65];

char release[65];

char version[65];

char machine[65];

char domainname[65];

};

extern int uname(struct utsname \*\_\_name);

### **14.4.95 sys/wait.h**

#define WIFSIGNALED(status) (!WIFSTOPPED(status) && !WIFEXITED(status))

#define WIFSTOPPED(status) (((status) & 0xff) == 0x7f)

#define WEXITSTATUS(status) (((status) & 0xff00) >> 8)

#define WTERMSIG(status) ((status) & 0x7f)

#define WCOREDUMP(status) ((status) & 0x80)

#define WIFEXITED(status) (WTERMSIG(status) == 0)

#define WNOHANG 0x00000001

#define WUNTRACED 0x00000002

#define WCOREFLAG 0x80

#define WSTOPSIG(status) WEXITSTATUS(status)

typedef enum {

P\_ALL,

P\_PID,

P\_PGID

} idtype\_t;

extern pid\_t wait(int \*\_\_stat\_loc);

extern pid\_t wait4(pid\_t \_\_pid, int \*\_\_stat\_loc, int \_\_options,

struct rusage \*\_\_usage);

extern int waitid(idtype\_t \_\_idtype, id\_t \_\_id, siginfo\_t \* \_\_infop,

int \_\_options);

extern pid\_t waitpid(pid\_t \_\_pid, int \*\_\_stat\_loc, int \_\_options);

### **14.4.96 sysexits.h**

#define EX\_OK 0 /\* successful termination \*/

#define EX\_USAGE 64 /\* command line usage error \*/

#define EX\_\_BASE 64 /\* base value for error messages \*/

#define EX\_DATAERR 65 /\* data format error \*/

#define EX\_NOINPUT 66 /\* cannot open input \*/

#define EX\_NOUSER 67 /\* addressee unknown \*/

#define EX\_NOHOST 68 /\* host name unknown \*/

#define EX\_UNAVAILABLE 69 /\* service unavailable \*/

#define EX\_SOFTWARE 70 /\* internal software error \*/

#define EX\_OSERR 71 /\* system error (e.g., cannot fork) \*/

#define EX\_OSFILE 72 /\* critical OS file missing \*/

#define EX\_CANTCREAT 73 /\* cannot create (user) output file \*/

#define EX\_IOERR 74 /\* input/output error \*/

#define EX\_TEMPFAIL 75 /\* temp failure; user is invited to retry \*/

#define EX\_PROTOCOL 76 /\* remote error in protocol \*/

#define EX\_NOPERM 77 /\* permission denied \*/

#define EX\_CONFIG 78 /\* configuration error \*/

### **14.4.97 syslog.h**

#define LOG\_MAKEPRI(fac, pri) (((fac) << 3) | (pri))

#define LOG\_PRI(p) ((p) & LOG\_PRIMASK) /\* extract priority \*/

#define LOG\_EMERG 0 /\* system is unusable \*/

#define LOG\_PRIMASK 0x07 /\* mask to extract priority part \*/

#define LOG\_ALERT 1 /\* action must be taken immediately \*/

#define LOG\_CRIT 2 /\* critical conditions \*/

#define LOG\_ERR 3 /\* error conditions \*/

#define LOG\_WARNING 4 /\* warning conditions \*/

#define LOG\_NOTICE 5 /\* normal but significant condition \*/

#define LOG\_INFO 6 /\* informational \*/

#define LOG\_DEBUG 7 /\* debug-level messages \*/

#define LOG\_FAC(p) (((p) & LOG\_FACMASK) >> 3) /\* facility of pri \*/

#define LOG\_KERN (0<<3) /\* kernel messages \*/

#define LOG\_AUTHPRIV (10<<3) /\* security/authorization messages (private) \*/

#define LOG\_FTP (11<<3) /\* ftp daemon \*/

#define LOG\_USER (1<<3) /\* random user-level messages \*/

#define LOG\_MAIL (2<<3) /\* mail system \*/

#define LOG\_DAEMON (3<<3) /\* system daemons \*/

#define LOG\_AUTH (4<<3) /\* security/authorization messages \*/

#define LOG\_SYSLOG (5<<3) /\* messages generated internally by syslogd \*/

#define LOG\_LPR (6<<3) /\* line printer subsystem \*/

#define LOG\_NEWS (7<<3) /\* network news subsystem \*/

#define LOG\_UUCP (8<<3) /\* UUCP subsystem \*/

#define LOG\_CRON (9<<3) /\* clock daemon \*/

#define LOG\_FACMASK 0x03f8 /\* mask to extract facility part \*/

#define LOG\_LOCAL0 (16<<3) /\* reserved for local use \*/

#define LOG\_LOCAL1 (17<<3) /\* reserved for local use \*/

#define LOG\_LOCAL2 (18<<3) /\* reserved for local use \*/

#define LOG\_LOCAL3 (19<<3) /\* reserved for local use \*/

#define LOG\_LOCAL4 (20<<3) /\* reserved for local use \*/

#define LOG\_LOCAL5 (21<<3) /\* reserved for local use \*/

#define LOG\_LOCAL6 (22<<3) /\* reserved for local use \*/

#define LOG\_LOCAL7 (23<<3) /\* reserved for local use \*/

#define LOG\_UPTO(pri) ((1 << ((pri)+1)) - 1) /\* all priorities through pri \*/

#define LOG\_MASK(pri) (1 << (pri)) /\* mask for one priority \*/

#define LOG\_PID 0x01 /\* log the pid with each message \*/

#define LOG\_CONS 0x02 /\* log on the console if errors in sending \*/

#define LOG\_ODELAY 0x04 /\* delay open until first syslog() (default) \*/

#define LOG\_NDELAY 0x08 /\* don't delay open \*/

#define LOG\_NOWAIT 0x10 /\* don't wait for console forks: DEPRECATED \*/

#define LOG\_PERROR 0x20 /\* log to stderr as well \*/

extern void \_\_syslog\_chk(int, int, const char \*, ...);

extern void \_\_vsyslog\_chk(int, int, const char \*, va\_list);

extern void closelog(void);

extern void openlog(const char \*\_\_ident, int \_\_option, int \_\_facility);

extern int setlogmask(int \_\_mask);

extern void syslog(int \_\_pri, const char \*\_\_fmt, ...);

extern void vsyslog(int \_\_pri, const char \*\_\_fmt, va\_list \_\_ap);

### **14.4.98 tar.h**

#define REGTYPE '0'

#define LNKTYPE '1'

#define SYMTYPE '2'

#define CHRTYPE '3'

#define BLKTYPE '4'

#define DIRTYPE '5'

#define FIFOTYPE '6'

#define CONTTYPE '7'

#define AREGTYPE '\0'

#define TVERSION "00"

#define TOEXEC 00001

#define TOWRITE 00002

#define TOREAD 00004

#define TGEXEC 00010

#define TGWRITE 00020

#define TGREAD 00040

#define TUEXEC 00100

#define TUWRITE 00200

#define TUREAD 00400

#define TSVTX 01000

#define TSGID 02000

#define TSUID 04000

#define TVERSLEN 2

#define TMAGLEN 6

#define TMAGIC "ustar"

### **14.4.99 termios.h**

#define TCIFLUSH 0

#define TCOOFF 0

#define TCSANOW 0

#define BS0 0000000

#define CR0 0000000

#define FF0 0000000

#define NL0 0000000

#define TAB0 0000000

#define VT0 0000000

#define OPOST 0000001

#define OCRNL 0000010

#define ONOCR 0000020

#define ONLRET 0000040

#define OFILL 0000100

#define OFDEL 0000200

#define NL1 0000400

#define TCOFLUSH 1

#define TCOON 1

#define TCSADRAIN 1

#define TCIOFF 2

#define TCIOFLUSH 2

#define TCSAFLUSH 2

#define TCION 3

typedef unsigned int speed\_t;

typedef unsigned char cc\_t;

typedef unsigned int tcflag\_t;

#define NCCS 32

struct termios {

tcflag\_t c\_iflag; /\* input mode flags \*/

tcflag\_t c\_oflag; /\* output mode flags \*/

tcflag\_t c\_cflag; /\* control mode flags \*/

tcflag\_t c\_lflag; /\* local mode flags \*/

cc\_t c\_line; /\* line discipline \*/

cc\_t c\_cc[NCCS]; /\* control characters \*/

speed\_t c\_ispeed; /\* input speed \*/

speed\_t c\_ospeed; /\* output speed \*/

};

#define VINTR 0

#define VQUIT 1

#define VLNEXT 15

#define VERASE 2

#define VKILL 3

#define VEOF 4

#define IGNBRK 0000001

#define BRKINT 0000002

#define IGNPAR 0000004

#define PARMRK 0000010

#define INPCK 0000020

#define ISTRIP 0000040

#define INLCR 0000100

#define IGNCR 0000200

#define ICRNL 0000400

#define IXANY 0004000

#define IMAXBEL 0020000

#define CS5 0000000

#define ECHO 0000010

#define B0 0000000

#define B50 0000001

#define B75 0000002

#define B110 0000003

#define B134 0000004

#define B150 0000005

#define B200 0000006

#define B300 0000007

#define B600 0000010

#define B1200 0000011

#define B1800 0000012

#define B2400 0000013

#define B4800 0000014

#define B9600 0000015

#define B19200 0000016

#define B38400 0000017

extern speed\_t cfgetispeed(const struct termios \*\_\_termios\_p);

extern speed\_t cfgetospeed(const struct termios \*\_\_termios\_p);

extern void cfmakeraw(struct termios \*\_\_termios\_p);

extern int cfsetispeed(struct termios \*\_\_termios\_p, speed\_t \_\_speed);

extern int cfsetospeed(struct termios \*\_\_termios\_p, speed\_t \_\_speed);

extern int cfsetspeed(struct termios \*\_\_termios\_p, speed\_t \_\_speed);

extern int tcdrain(int \_\_fd);

extern int tcflow(int \_\_fd, int \_\_action);

extern int tcflush(int \_\_fd, int \_\_queue\_selector);

extern int tcgetattr(int \_\_fd, struct termios \*\_\_termios\_p);

extern pid\_t tcgetsid(int \_\_fd);

extern int tcsendbreak(int \_\_fd, int \_\_duration);

extern int tcsetattr(int \_\_fd, int \_\_optional\_actions,

const struct termios \*\_\_termios\_p);

### **14.4.100 time.h**

#define CLK\_TCK ((clock\_t)sysconf(2))

#define timerclear(tvp) ((tvp)->tv\_sec = (tvp)->tv\_usec = 0)

#define timerisset(tvp) ((tvp)->tv\_sec || (tvp)->tv\_usec)

#define CLOCK\_REALTIME 0

#define CLOCK\_MONOTONIC 1

#define TIMER\_ABSTIME 1

#define CLOCKS\_PER\_SEC 1000000l

#define CLOCK\_PROCESS\_CPUTIME\_ID 2

#define CLOCK\_THREAD\_CPUTIME\_ID 3

#define timeradd(a,b,result) \

do { \

(result)->tv\_sec = (a)->tv\_sec + (b)->tv\_sec; \

(result)->tv\_usec = (a)->tv\_usec + (b)->tv\_usec; \

if ((result)->tv\_usec >= 1000000) \

{ \

++(result)->tv\_sec; \

(result)->tv\_usec -= 1000000; \

} \

} while (0)

#define timersub(a,b,result) \

do { \

(result)->tv\_sec = (a)->tv\_sec - (b)->tv\_sec; \

(result)->tv\_usec = (a)->tv\_usec - (b)->tv\_usec; \

if ((result)->tv\_usec < 0) { \

--(result)->tv\_sec; \

(result)->tv\_usec += 1000000; \

} \

} while (0)

#define timercmp(a,b,CMP) \

(((a)->tv\_sec == (b)->tv\_sec) ? \

((a)->tv\_usec CMP (b)->tv\_usec) : \

((a)->tv\_sec CMP (b)->tv\_sec))

struct tm {

int tm\_sec;

int tm\_min;

int tm\_hour;

int tm\_mday;

int tm\_mon;

int tm\_year;

int tm\_wday;

int tm\_yday;

int tm\_isdst;

long int tm\_gmtoff;

char \*tm\_zone;

};

struct timespec {

time\_t tv\_sec;

long int tv\_nsec;

};

struct itimerspec {

struct timespec it\_interval;

struct timespec it\_value;

};

extern int \_\_daylight;

extern long int \_\_timezone;

extern char \*\_\_tzname[];

extern char \*asctime(const struct tm \*\_\_tp);

extern char \*asctime\_r(const struct tm \*\_\_tp, char \*\_\_buf);

extern clock\_t clock(void);

extern int clock\_getcpuclockid(pid\_t \_\_pid, clockid\_t \* \_\_clock\_id);

extern int clock\_getres(clockid\_t \_\_clock\_id, struct timespec \*\_\_res);

extern int clock\_gettime(clockid\_t \_\_clock\_id, struct timespec \*\_\_tp);

extern int clock\_nanosleep(clockid\_t \_\_clock\_id, int \_\_flags,

const struct timespec \*\_\_req,

struct timespec \*\_\_rem);

extern int clock\_settime(clockid\_t \_\_clock\_id,

const struct timespec \*\_\_tp);

extern char \*ctime(const time\_t \* \_\_timer);

extern char \*ctime\_r(const time\_t \* \_\_timer, char \*\_\_buf);

extern int daylight;

extern double difftime(time\_t \_\_time1, time\_t \_\_time0);

extern struct tm \*getdate(const char \*\_\_string);

extern int getdate\_err;

extern struct tm \*gmtime(const time\_t \* \_\_timer);

extern struct tm \*gmtime\_r(const time\_t \* \_\_timer, struct tm \*\_\_tp);

extern struct tm \*localtime(const time\_t \* \_\_timer);

extern struct tm \*localtime\_r(const time\_t \* \_\_timer, struct tm \*\_\_tp);

extern time\_t mktime(struct tm \*\_\_tp);

extern int nanosleep(const struct timespec \*\_\_requested\_time,

struct timespec \*\_\_remaining);

extern int stime(const time\_t \* \_\_when);

extern size\_t strftime(char \*\_\_s, size\_t \_\_maxsize, const char \*\_\_format,

const struct tm \*\_\_tp);

extern size\_t strftime\_l(char \*s, size\_t maxsize, const char \*format,

const struct tm \*timeptr, locale\_t locale);

extern char \*strptime(const char \*\_\_s, const char \*\_\_fmt, struct tm \*\_\_tp);

extern time\_t time(time\_t \* \_\_timer);

extern int timer\_create(clockid\_t \_\_clock\_id, struct sigevent \*\_\_evp,

timer\_t \* \_\_timerid);

extern int timer\_delete(timer\_t \_\_timerid);

extern int timer\_getoverrun(timer\_t \_\_timerid);

extern int timer\_gettime(timer\_t \_\_timerid, struct itimerspec \*\_\_value);

extern int timer\_settime(timer\_t \_\_timerid, int \_\_flags,

const struct itimerspec \*\_\_value,

struct itimerspec \*\_\_ovalue);

extern long int timezone;

extern char \*tzname[];

extern void tzset(void);

### **14.4.101 ucontext.h**

extern int getcontext(ucontext\_t \* \_\_ucp);

extern void makecontext(ucontext\_t \* \_\_ucp, void (\*\_\_func) (void),

int \_\_argc, ...);

extern int setcontext(const struct ucontext \*\_\_ucp);

extern int swapcontext(ucontext\_t \* \_\_oucp, const struct ucontext \*\_\_ucp);

### **14.4.102 ulimit.h**

#define UL\_GETFSIZE 1

#define UL\_SETFSIZE 2

extern long int ulimit(int \_\_cmd, ...);

### **14.4.103 unistd.h**

#ifndef SEEK\_SET

#define SEEK\_SET 0

#endif

#define STDIN\_FILENO 0

#ifndef SEEK\_CUR

#define SEEK\_CUR 1

#endif

#define STDOUT\_FILENO 1

#ifndef SEEK\_END

#define SEEK\_END 2

#endif

#define STDERR\_FILENO 2

#define F\_OK 0

#define X\_OK 1

#define W\_OK 2

#define R\_OK 4

#define \_POSIX\_VDISABLE '\0'

#define \_POSIX\_ASYNC\_IO 0

#define \_POSIX\_CHOWN\_RESTRICTED 1

#define \_POSIX\_JOB\_CONTROL 1

#define \_POSIX\_NO\_TRUNC 1

#define \_POSIX\_SHELL 1

#define \_POSIX2\_CHAR\_TERM 200809L

#define \_POSIX2\_C\_BIND 200809L

#define \_POSIX2\_LOCALEDEF 200809L

#define \_POSIX2\_VERSION 200809L

#define \_POSIX\_ADVISORY\_INFO 200809L

#define \_POSIX\_BARRIERS 200809L

#define \_POSIX\_CLOCK\_SELECTION 200809L

#define \_POSIX\_FSYNC 200809L

#define \_POSIX\_IPV6 200809L

#define \_POSIX\_MAPPED\_FILES 200809L

#define \_POSIX\_MEMLOCK 200809L

#define \_POSIX\_MEMLOCK\_RANGE 200809L

#define \_POSIX\_MEMORY\_PROTECTION 200809L

#define \_POSIX\_MESSAGE\_PASSING 200809L

#define \_POSIX\_PRIORITIZED\_IO 200809L

#define \_POSIX\_PRIORITY\_SCHEDULING 200809L

#define \_POSIX\_RAW\_SOCKETS 200809L

#define \_POSIX\_READER\_WRITER\_LOCKS 200809L

#define \_POSIX\_REALTIME\_SIGNALS 200809L

#define \_POSIX\_SEMAPHORES 200809L

#define \_POSIX\_SHARED\_MEMORY\_OBJECTS 200809L

#define \_POSIX\_SPAWN 200809L

#define \_POSIX\_SPIN\_LOCKS 200809L

#define \_POSIX\_SYNCHRONIZED\_IO 200809L

#define \_POSIX\_THREADS 200809L

#define \_POSIX\_THREAD\_ATTR\_STACKADDR 200809L

#define \_POSIX\_THREAD\_ATTR\_STACKSIZE 200809L

#define \_POSIX\_THREAD\_PRIORITY\_SCHEDULING 200809L

#define \_POSIX\_THREAD\_PRIO\_INHERIT 200809L

#define \_POSIX\_THREAD\_PRIO\_PROTECT 200809L

#define \_POSIX\_THREAD\_PROCESS\_SHARED 200809L

#define \_POSIX\_THREAD\_ROBUST\_PRIO\_INHERIT 200809L

#define \_POSIX\_THREAD\_SAFE\_FUNCTIONS 200809L

#define \_POSIX\_TIMEOUTS 200809L

#define \_POSIX\_TIMERS 200809L

#define \_POSIX\_VERSION 200809L

#define \_PC\_LINK\_MAX 0

#define \_PC\_MAX\_CANON 1

#define \_PC\_ASYNC\_IO 10

#define \_PC\_PRIO\_IO 11

#define \_PC\_FILESIZEBITS 13

#define \_PC\_REC\_INCR\_XFER\_SIZE 14

#define \_PC\_REC\_MIN\_XFER\_SIZE 16

#define \_PC\_REC\_XFER\_ALIGN 17

#define \_PC\_ALLOC\_SIZE\_MIN 18

#define \_PC\_MAX\_INPUT 2

#define \_PC\_2\_SYMLINKS 20

#define \_PC\_NAME\_MAX 3

#define \_PC\_PATH\_MAX 4

#define \_PC\_PIPE\_BUF 5

#define \_PC\_CHOWN\_RESTRICTED 6

#define \_PC\_NO\_TRUNC 7

#define \_PC\_VDISABLE 8

#define \_PC\_SYNC\_IO 9

#define \_SC\_ARG\_MAX 0

#define \_SC\_CHILD\_MAX 1

#define \_SC\_PRIORITY\_SCHEDULING 10

#define \_SC\_XOPEN\_XPG4 100

#define \_SC\_CHAR\_BIT 101

#define \_SC\_CHAR\_MAX 102

#define \_SC\_CHAR\_MIN 103

#define \_SC\_INT\_MAX 104

#define \_SC\_INT\_MIN 105

#define \_SC\_LONG\_BIT 106

#define \_SC\_WORD\_BIT 107

#define \_SC\_MB\_LEN\_MAX 108

#define \_SC\_NZERO 109

#define \_SC\_TIMERS 11

#define \_SC\_SSIZE\_MAX 110

#define \_SC\_SCHAR\_MAX 111

#define \_SC\_SCHAR\_MIN 112

#define \_SC\_SHRT\_MAX 113

#define \_SC\_SHRT\_MIN 114

#define \_SC\_UCHAR\_MAX 115

#define \_SC\_UINT\_MAX 116

#define \_SC\_ULONG\_MAX 117

#define \_SC\_USHRT\_MAX 118

#define \_SC\_NL\_ARGMAX 119

#define \_SC\_ASYNCHRONOUS\_IO 12

#define \_SC\_NL\_LANGMAX 120

#define \_SC\_NL\_MSGMAX 121

#define \_SC\_NL\_NMAX 122

#define \_SC\_NL\_SETMAX 123

#define \_SC\_NL\_TEXTMAX 124

#define \_SC\_XBS5\_ILP32\_OFF32 125

#define \_SC\_XBS5\_ILP32\_OFFBIG 126

#define \_SC\_XBS5\_LP64\_OFF64 127

#define \_SC\_XBS5\_LPBIG\_OFFBIG 128

#define \_SC\_XOPEN\_LEGACY 129

#define \_SC\_PRIORITIZED\_IO 13

#define \_SC\_XOPEN\_REALTIME 130

#define \_SC\_XOPEN\_REALTIME\_THREADS 131

#define \_SC\_ADVISORY\_INFO 132

#define \_SC\_BARRIERS 133

#define \_SC\_BASE 134

#define \_SC\_C\_LANG\_SUPPORT 135

#define \_SC\_C\_LANG\_SUPPORT\_R 136

#define \_SC\_CLOCK\_SELECTION 137

#define \_SC\_CPUTIME 138

#define \_SC\_THREAD\_CPUTIME 139

#define \_SC\_SYNCHRONIZED\_IO 14

#define \_SC\_DEVICE\_IO 140

#define \_SC\_DEVICE\_SPECIFIC 141

#define \_SC\_DEVICE\_SPECIFIC\_R 142

#define \_SC\_FD\_MGMT 143

#define \_SC\_FIFO 144

#define \_SC\_PIPE 145

#define \_SC\_FILE\_ATTRIBUTES 146

#define \_SC\_FILE\_LOCKING 147

#define \_SC\_FILE\_SYSTEM 148

#define \_SC\_MONOTONIC\_CLOCK 149

#define \_SC\_FSYNC 15

#define \_SC\_MULTI\_PROCESS 150

#define \_SC\_SINGLE\_PROCESS 151

#define \_SC\_NETWORKING 152

#define \_SC\_READER\_WRITER\_LOCKS 153

#define \_SC\_SPIN\_LOCKS 154

#define \_SC\_REGEXP 155

#define \_SC\_REGEX\_VERSION 156

#define \_SC\_SHELL 157

#define \_SC\_SIGNALS 158

#define \_SC\_SPAWN 159

#define \_SC\_MAPPED\_FILES 16

#define \_SC\_SPORADIC\_SERVER 160

#define \_SC\_THREAD\_SPORADIC\_SERVER 161

#define \_SC\_SYSTEM\_DATABASE 162

#define \_SC\_SYSTEM\_DATABASE\_R 163

#define \_SC\_TIMEOUTS 164

#define \_SC\_TYPED\_MEMORY\_OBJECTS 165

#define \_SC\_USER\_GROUPS 166

#define \_SC\_USER\_GROUPS\_R 167

#define \_SC\_2\_PBS 168

#define \_SC\_2\_PBS\_ACCOUNTING 169

#define \_SC\_MEMLOCK 17

#define \_SC\_2\_PBS\_LOCATE 170

#define \_SC\_2\_PBS\_MESSAGE 171

#define \_SC\_2\_PBS\_TRACK 172

#define \_SC\_SYMLOOP\_MAX 173

#define \_SC\_STREAMS 174

#define \_SC\_2\_PBS\_CHECKPOINT 175

#define \_SC\_V6\_ILP32\_OFF32 176

#define \_SC\_V6\_ILP32\_OFFBIG 177

#define \_SC\_V6\_LP64\_OFF64 178

#define \_SC\_V6\_LPBIG\_OFFBIG 179

#define \_SC\_MEMLOCK\_RANGE 18

#define \_SC\_HOST\_NAME\_MAX 180

#define \_SC\_TRACE 181

#define \_SC\_TRACE\_EVENT\_FILTER 182

#define \_SC\_TRACE\_INHERIT 183

#define \_SC\_TRACE\_LOG 184

#define \_SC\_LEVEL1\_ICACHE\_SIZE 185

#define \_SC\_LEVEL1\_ICACHE\_ASSOC 186

#define \_SC\_LEVEL1\_ICACHE\_LINESIZE 187

#define \_SC\_LEVEL1\_DCACHE\_SIZE 188

#define \_SC\_LEVEL1\_DCACHE\_ASSOC 189

#define \_SC\_MEMORY\_PROTECTION 19

#define \_SC\_LEVEL1\_DCACHE\_LINESIZE 190

#define \_SC\_LEVEL2\_CACHE\_SIZE 191

#define \_SC\_LEVEL2\_CACHE\_ASSOC 192

#define \_SC\_LEVEL2\_CACHE\_LINESIZE 193

#define \_SC\_LEVEL3\_CACHE\_SIZE 194

#define \_SC\_LEVEL3\_CACHE\_ASSOC 195

#define \_SC\_LEVEL3\_CACHE\_LINESIZE 196

#define \_SC\_LEVEL4\_CACHE\_SIZE 197

#define \_SC\_LEVEL4\_CACHE\_ASSOC 198

#define \_SC\_LEVEL4\_CACHE\_LINESIZE 199

#define \_SC\_CLK\_TCK 2

#define \_SC\_MESSAGE\_PASSING 20

#define \_SC\_SEMAPHORES 21

#define \_SC\_SHARED\_MEMORY\_OBJECTS 22

#define \_SC\_AIO\_LISTIO\_MAX 23

#define \_SC\_IPV6 235

#define \_SC\_RAW\_SOCKETS 236

#define \_SC\_V7\_ILP32\_OFF32 237

#define \_SC\_V7\_ILP32\_OFFBIG 238

#define \_SC\_V7\_LP64\_OFF64 239

#define \_SC\_AIO\_MAX 24

#define \_SC\_V7\_LPBIG\_OFFBIG 240

#define \_SC\_SS\_REPL\_MAX 241

#define \_SC\_TRACE\_EVENT\_NAME\_MAX 242

#define \_SC\_TRACE\_NAME\_MAX 243

#define \_SC\_TRACE\_SYS\_MAX 244

#define \_SC\_TRACE\_USER\_EVENT\_MAX 245

#define \_SC\_XOPEN\_STREAMS 246

#define \_SC\_THREAD\_ROBUST\_PRIO\_INHERIT 247

#define \_SC\_THREAD\_ROBUST\_PRIO\_PROTECT 248

#define \_SC\_AIO\_PRIO\_DELTA\_MAX 25

#define \_SC\_DELAYTIMER\_MAX 26

#define \_SC\_MQ\_OPEN\_MAX 27

#define \_SC\_MQ\_PRIO\_MAX 28

#define \_SC\_VERSION 29

#define \_SC\_NGROUPS\_MAX 3

#define \_SC\_PAGESIZE 30

#define \_SC\_PAGE\_SIZE 30

#define \_SC\_RTSIG\_MAX 31

#define \_SC\_SEM\_NSEMS\_MAX 32

#define \_SC\_SEM\_VALUE\_MAX 33

#define \_SC\_SIGQUEUE\_MAX 34

#define \_SC\_TIMER\_MAX 35

#define \_SC\_BC\_BASE\_MAX 36

#define \_SC\_BC\_DIM\_MAX 37

#define \_SC\_BC\_SCALE\_MAX 38

#define \_SC\_BC\_STRING\_MAX 39

#define \_SC\_OPEN\_MAX 4

#define \_SC\_COLL\_WEIGHTS\_MAX 40

#define \_SC\_EQUIV\_CLASS\_MAX 41

#define \_SC\_EXPR\_NEST\_MAX 42

#define \_SC\_LINE\_MAX 43

#define \_SC\_RE\_DUP\_MAX 44

#define \_SC\_CHARCLASS\_NAME\_MAX 45

#define \_SC\_2\_VERSION 46

#define \_SC\_2\_C\_BIND 47

#define \_SC\_2\_C\_DEV 48

#define \_SC\_2\_FORT\_DEV 49

#define \_SC\_STREAM\_MAX 5

#define \_SC\_2\_FORT\_RUN 50

#define \_SC\_2\_SW\_DEV 51

#define \_SC\_2\_LOCALEDEF 52

#define \_SC\_PII 53

#define \_SC\_PII\_XTI 54

#define \_SC\_PII\_SOCKET 55

#define \_SC\_PII\_INTERNET 56

#define \_SC\_PII\_OSI 57

#define \_SC\_POLL 58

#define \_SC\_SELECT 59

#define \_SC\_TZNAME\_MAX 6

#define \_SC\_IOV\_MAX 60

#define \_SC\_UIO\_MAXIOV 60

#define \_SC\_PII\_INTERNET\_STREAM 61

#define \_SC\_PII\_INTERNET\_DGRAM 62

#define \_SC\_PII\_OSI\_COTS 63

#define \_SC\_PII\_OSI\_CLTS 64

#define \_SC\_PII\_OSI\_M 65

#define \_SC\_T\_IOV\_MAX 66

#define \_SC\_THREADS 67

#define \_SC\_THREAD\_SAFE\_FUNCTIONS 68

#define \_SC\_GETGR\_R\_SIZE\_MAX 69

#define \_SC\_JOB\_CONTROL 7

#define \_SC\_GETPW\_R\_SIZE\_MAX 70

#define \_SC\_LOGIN\_NAME\_MAX 71

#define \_SC\_TTY\_NAME\_MAX 72

#define \_SC\_THREAD\_DESTRUCTOR\_ITERATIONS 73

#define \_SC\_THREAD\_KEYS\_MAX 74

#define \_SC\_THREAD\_STACK\_MIN 75

#define \_SC\_THREAD\_THREADS\_MAX 76

#define \_SC\_THREAD\_ATTR\_STACKADDR 77

#define \_SC\_THREAD\_ATTR\_STACKSIZE 78

#define \_SC\_THREAD\_PRIORITY\_SCHEDULING 79

#define \_SC\_SAVED\_IDS 8

#define \_SC\_THREAD\_PRIO\_INHERIT 80

#define \_SC\_THREAD\_PRIO\_PROTECT 81

#define \_SC\_THREAD\_PROCESS\_SHARED 82

#define \_SC\_NPROCESSORS\_CONF 83

#define \_SC\_NPROCESSORS\_ONLN 84

#define \_SC\_PHYS\_PAGES 85

#define \_SC\_AVPHYS\_PAGES 86

#define \_SC\_ATEXIT\_MAX 87

#define \_SC\_PASS\_MAX 88

#define \_SC\_XOPEN\_VERSION 89

#define \_SC\_REALTIME\_SIGNALS 9

#define \_SC\_XOPEN\_XCU\_VERSION 90

#define \_SC\_XOPEN\_UNIX 91

#define \_SC\_XOPEN\_CRYPT 92

#define \_SC\_XOPEN\_ENH\_I18N 93

#define \_SC\_XOPEN\_SHM 94

#define \_SC\_2\_CHAR\_TERM 95

#define \_SC\_2\_C\_VERSION 96

#define \_SC\_2\_UPE 97

#define \_SC\_XOPEN\_XPG2 98

#define \_SC\_XOPEN\_XPG3 99

#define \_CS\_PATH 0

#define \_POSIX\_REGEXP 1

#define \_CS\_XBS5\_ILP32\_OFF32\_CFLAGS 1100

#define \_CS\_XBS5\_ILP32\_OFF32\_LDFLAGS 1101

#define \_CS\_XBS5\_ILP32\_OFF32\_LIBS 1102

#define \_CS\_XBS5\_ILP32\_OFF32\_LINTFLAGS 1103

#define \_CS\_XBS5\_ILP32\_OFFBIG\_CFLAGS 1104

#define \_CS\_XBS5\_ILP32\_OFFBIG\_LDFLAGS 1105

#define \_CS\_XBS5\_ILP32\_OFFBIG\_LIBS 1106

#define \_CS\_XBS5\_ILP32\_OFFBIG\_LINTFLAGS 1107

#define \_CS\_XBS5\_LP64\_OFF64\_CFLAGS 1108

#define \_CS\_XBS5\_LP64\_OFF64\_LDFLAGS 1109

#define \_CS\_XBS5\_LP64\_OFF64\_LIBS 1110

#define \_CS\_XBS5\_LP64\_OFF64\_LINTFLAGS 1111

#define \_CS\_XBS5\_LPBIG\_OFFBIG\_CFLAGS 1112

#define \_CS\_XBS5\_LPBIG\_OFFBIG\_LDFLAGS 1113

#define \_CS\_XBS5\_LPBIG\_OFFBIG\_LIBS 1114

#define \_CS\_XBS5\_LPBIG\_OFFBIG\_LINTFLAGS 1115

#define \_CS\_POSIX\_V6\_ILP32\_OFF32\_CFLAGS 1116

#define \_CS\_POSIX\_V6\_ILP32\_OFF32\_LDFLAGS 1117

#define \_CS\_POSIX\_V6\_ILP32\_OFF32\_LIBS 1118

#define \_CS\_POSIX\_V6\_ILP32\_OFF32\_LINTFLAGS 1119

#define \_CS\_POSIX\_V6\_ILP32\_OFFBIG\_CFLAGS 1120

#define \_CS\_POSIX\_V6\_ILP32\_OFFBIG\_LDFLAGS 1121

#define \_CS\_POSIX\_V6\_ILP32\_OFFBIG\_LIBS 1122

#define \_CS\_POSIX\_V6\_ILP32\_OFFBIG\_LINTFLAGS 1123

#define \_CS\_POSIX\_V6\_LP64\_OFF64\_CFLAGS 1124

#define \_CS\_POSIX\_V6\_LP64\_OFF64\_LDFLAGS 1125

#define \_CS\_POSIX\_V6\_LP64\_OFF64\_LIBS 1126

#define \_CS\_POSIX\_V6\_LP64\_OFF64\_LINTFLAGS 1127

#define \_CS\_POSIX\_V6\_LPBIG\_OFFBIG\_CFLAGS 1128

#define \_CS\_POSIX\_V6\_LPBIG\_OFFBIG\_LDFLAGS 1129

#define \_CS\_POSIX\_V6\_LPBIG\_OFFBIG\_LIBS 1130

#define \_CS\_POSIX\_V6\_LPBIG\_OFFBIG\_LINTFLAGS 1131

#define \_CS\_POSIX\_V7\_ILP32\_OFF32\_CFLAGS 1132

#define \_CS\_POSIX\_V7\_ILP32\_OFF32\_LDFLAGS 1133

#define \_CS\_POSIX\_V7\_ILP32\_OFF32\_LIBS 1134

#define \_CS\_POSIX\_V7\_ILP32\_OFF32\_LINTFLAGS 1135

#define \_CS\_POSIX\_V7\_ILP32\_OFFBIG\_CFLAGS 1136

#define \_CS\_POSIX\_V7\_ILP32\_OFFBIG\_LDFLAGS 1137

#define \_CS\_POSIX\_V7\_ILP32\_OFFBIG\_LIBS 1138

#define \_CS\_POSIX\_V7\_ILP32\_OFFBIG\_LINTFLAGS 1139

#define \_CS\_POSIX\_V7\_LP64\_OFF64\_CFLAGS 1140

#define \_CS\_POSIX\_V7\_LP64\_OFF64\_LDFLAGS 1141

#define \_CS\_POSIX\_V7\_LP64\_OFF64\_LIBS 1142

#define \_CS\_POSIX\_V7\_LP64\_OFF64\_LINTFLAGS 1143

#define \_CS\_POSIX\_V7\_LPBIG\_OFFBIG\_CFLAGS 1144

#define \_CS\_POSIX\_V7\_LPBIG\_OFFBIG\_LDFLAGS 1145

#define \_CS\_POSIX\_V7\_LPBIG\_OFFBIG\_LIBS 1146

#define \_CS\_POSIX\_V7\_LPBIG\_OFFBIG\_LINTFLAGS 1147

#define \_CS\_V6\_ENV 1148

#define \_CS\_V7\_ENV 1149

#define \_XOPEN\_XPG4 1

#define \_XOPEN\_VERSION 700

#define F\_ULOCK 0

#define F\_LOCK 1

#define F\_TLOCK 2

#define F\_TEST 3

extern size\_t \_\_confstr\_chk(int, char \*, size\_t, size\_t);

extern char \*\*\_\_environ;

extern char \*\_\_getcwd\_chk(char \*, size\_t, size\_t);

extern int \_\_getgroups\_chk(int, gid\_t \*, size\_t);

extern int \_\_gethostname\_chk(char \*, size\_t, size\_t);

extern int \_\_getlogin\_r\_chk(char \*, size\_t, size\_t);

extern pid\_t \_\_getpgid(pid\_t \_\_pid);

extern ssize\_t \_\_pread64\_chk(int, void \*, size\_t, off64\_t, size\_t);

extern ssize\_t \_\_pread\_chk(int, void \*, size\_t, off\_t, size\_t);

extern ssize\_t \_\_read\_chk(int, void \*, size\_t, size\_t);

extern ssize\_t \_\_readlink\_chk(const char \*, char \*, size\_t, size\_t);

extern int \_\_ttyname\_r\_chk(int, char \*, size\_t, size\_t);

extern char \*\*\_environ;

extern void \_exit(int \_\_status);

extern int access(const char \*\_\_name, int \_\_type);

extern int acct(const char \*\_\_name);

extern unsigned int alarm(unsigned int \_\_seconds);

extern int brk(void \*\_\_addr);

extern int chdir(const char \*\_\_path);

extern int chown(const char \*\_\_file, uid\_t \_\_owner, gid\_t \_\_group);

extern int chroot(const char \*\_\_path);

extern int close(int \_\_fd);

extern size\_t confstr(int \_\_name, char \*\_\_buf, size\_t \_\_len);

extern char \*crypt(const char \*\_\_key, const char \*\_\_salt);

extern char \*ctermid(char \*\_\_s);

extern char \*cuserid(char \*\_\_s);

extern int daemon(int \_\_nochdir, int \_\_noclose);

extern int dup(int \_\_fd);

extern int dup2(int \_\_fd, int \_\_fd2);

extern void encrypt(char \*\_\_block, int \_\_edflag);

extern int execl(const char \*\_\_path, const char \*\_\_arg, ...);

extern int execle(const char \*\_\_path, const char \*\_\_arg, ...);

extern int execlp(const char \*\_\_file, const char \*\_\_arg, ...);

extern int execv(const char \*\_\_path, char \*const \_\_argv[]);

extern int execve(const char \*\_\_path, char \*const \_\_argv[],

char \*const \_\_envp[]);

extern int execvp(const char \*\_\_file, char \*const \_\_argv[]);

extern int faccessat(int \_\_fd, const char \*\_\_file, int \_\_type, int \_\_flag);

extern int fchdir(int \_\_fd);

extern int fchown(int \_\_fd, uid\_t \_\_owner, gid\_t \_\_group);

extern int fchownat(int \_\_fd, const char \*\_\_file, uid\_t \_\_owner,

gid\_t \_\_group, int \_\_flag);

extern int fdatasync(int \_\_fildes);

extern int fexecve(int \_\_fd, char \*const \_\_argv[], char \*const \_\_envp[]);

extern pid\_t fork(void);

extern long int fpathconf(int \_\_fd, int \_\_name);

extern int fsync(int \_\_fd);

extern int ftruncate(int \_\_fd, off\_t \_\_length);

extern int ftruncate64(int \_\_fd, off64\_t \_\_length);

extern char \*getcwd(char \*\_\_buf, size\_t \_\_size);

extern int getdomainname(char \*\_\_name, size\_t \_\_len);

extern int getdtablesize(void);

extern gid\_t getegid(void);

extern uid\_t geteuid(void);

extern gid\_t getgid(void);

extern int getgroups(int \_\_size, gid\_t \_\_list[]);

extern long int gethostid(void);

extern int gethostname(char \*\_\_name, size\_t \_\_len);

extern char \*getlogin(void);

extern int getlogin\_r(char \*\_\_name, size\_t \_\_name\_len);

extern int getopt(int \_\_\_argc, char \*const \_\_\_argv[],

const char \*\_\_shortopts);

extern int getpagesize(void);

extern pid\_t getpgid(pid\_t \_\_pid);

extern pid\_t getpgrp(void);

extern pid\_t getpid(void);

extern pid\_t getppid(void);

extern pid\_t getsid(pid\_t \_\_pid);

extern uid\_t getuid(void);

extern char \*getwd(char \*\_\_buf);

extern int isatty(int \_\_fd);

extern int lchown(const char \*\_\_file, uid\_t \_\_owner, gid\_t \_\_group);

extern int link(const char \*\_\_from, const char \*\_\_to);

extern int linkat(int \_\_fromfd, const char \*\_\_from, int \_\_tofd,

const char \*\_\_to, int \_\_flags);

extern int lockf(int \_\_fd, int \_\_cmd, off\_t \_\_len);

extern int lockf64(int \_\_fd, int \_\_cmd, off64\_t \_\_len);

extern off\_t lseek(int \_\_fd, off\_t \_\_offset, int \_\_whence);

extern loff\_t lseek64(int \_\_fd, loff\_t \_\_offset, int \_\_whence);

extern int nice(int \_\_inc);

extern char \*optarg;

extern int opterr;

extern int optind;

extern int optopt;

extern long int pathconf(const char \*\_\_path, int \_\_name);

extern int pause(void);

extern int pipe(int \_\_pipedes[2]);

extern ssize\_t pread(int \_\_fd, void \*\_\_buf, size\_t \_\_nbytes,

off\_t \_\_offset);

extern ssize\_t pread64(int \_\_fd, void \*\_\_buf, size\_t \_\_nbytes,

off64\_t \_\_offset);

extern ssize\_t pwrite(int \_\_fd, const void \*\_\_buf, size\_t \_\_n,

off\_t \_\_offset);

extern ssize\_t pwrite64(int \_\_fd, const void \*\_\_buf, size\_t \_\_n,

off64\_t \_\_offset);

extern ssize\_t read(int \_\_fd, void \*\_\_buf, size\_t \_\_nbytes);

extern ssize\_t readlink(const char \*\_\_path, char \*\_\_buf, size\_t \_\_len);

extern ssize\_t readlinkat(int \_\_fd, const char \*\_\_path, char \*\_\_buf,

size\_t \_\_len);

extern int rmdir(const char \*\_\_path);

extern void \*sbrk(intptr\_t \_\_delta);

extern int setegid(gid\_t \_\_gid);

extern int seteuid(uid\_t \_\_uid);

extern int setgid(gid\_t \_\_gid);

extern int sethostname(const char \*\_\_name, size\_t \_\_len);

extern void setkey(const char \*\_\_key);

extern int setpgid(pid\_t \_\_pid, pid\_t \_\_pgid);

extern int setpgrp(void);

extern int setregid(gid\_t \_\_rgid, gid\_t \_\_egid);

extern int setreuid(uid\_t \_\_ruid, uid\_t \_\_euid);

extern pid\_t setsid(void);

extern int setuid(uid\_t \_\_uid);

extern unsigned int sleep(unsigned int \_\_seconds);

extern void swab(const void \*\_\_from, void \*\_\_to, ssize\_t \_\_n);

extern int symlink(const char \*\_\_from, const char \*\_\_to);

extern int symlinkat(const char \*\_\_from, int \_\_tofd, const char \*\_\_to);

extern void sync(void);

extern long int sysconf(int \_\_name);

extern pid\_t tcgetpgrp(int \_\_fd);

extern int tcsetpgrp(int \_\_fd, pid\_t \_\_pgrp\_id);

extern int truncate(const char \*\_\_file, off\_t \_\_length);

extern int truncate64(const char \*\_\_file, off64\_t \_\_length);

extern char \*ttyname(int \_\_fd);

extern int ttyname\_r(int \_\_fd, char \*\_\_buf, size\_t \_\_buflen);

extern unsigned int ualarm(useconds\_t \_\_value, useconds\_t \_\_interval);

extern int unlink(const char \*\_\_name);

extern int unlinkat(int \_\_fd, const char \*\_\_name, int \_\_flag);

extern int usleep(useconds\_t \_\_useconds);

extern pid\_t vfork(void);

extern ssize\_t write(int \_\_fd, const void \*\_\_buf, size\_t \_\_n);

### **14.4.104 utime.h**

struct utimbuf {

time\_t actime;

time\_t modtime;

};

extern int utime(const char \*\_\_file, const struct utimbuf \*\_\_file\_times);

### **14.4.105 utmp.h**

#define UT\_HOSTSIZE 256

#define UT\_LINESIZE 32

#define UT\_NAMESIZE 32

#define ut\_addr ut\_addr\_v6[0]

#define ut\_time ut\_tv.tv\_sec

#define ut\_name ut\_user /\* Backwards compatability \*/

struct exit\_status {

short e\_termination; /\* Process termination status. \*/

short e\_exit; /\* Process exit status. \*/

};

#define EMPTY 0 /\* No valid user accounting information. \*/

#define RUN\_LVL 1 /\* The system's runlevel. \*/

#define BOOT\_TIME 2 /\* Time of system boot. \*/

#define NEW\_TIME 3 /\* Time after system clock changed. \*/

#define OLD\_TIME 4 /\* Time when system clock changed. \*/

#define INIT\_PROCESS 5 /\* Process spawned by the init process. \*/

#define LOGIN\_PROCESS 6 /\* Session leader of a logged in user. \*/

#define USER\_PROCESS 7 /\* Normal process. \*/

#define DEAD\_PROCESS 8 /\* Terminated process. \*/

#define ACCOUNTING 9

extern void endutent(void);

extern struct utmp \*getutent(void);

extern int getutent\_r(struct utmp \*\_\_buffer, struct utmp \*\*\_\_result);

extern void login(const struct utmp \*\_\_entry);

extern int login\_tty(int \_\_fd);

extern int logout(const char \*\_\_ut\_line);

extern void logwtmp(const char \*\_\_ut\_line, const char \*\_\_ut\_name,

const char \*\_\_ut\_host);

extern void setutent(void);

extern int utmpname(const char \*\_\_file);

### **14.4.106 utmpx.h**

extern void endutxent(void);

extern struct utmpx \*getutxent(void);

extern struct utmpx \*getutxid(const struct utmpx \*\_\_id);

extern struct utmpx \*getutxline(const struct utmpx \*\_\_line);

extern struct utmpx \*pututxline(const struct utmpx \*\_\_utmpx);

extern void setutxent(void);

### **14.4.107 wchar.h**

#define WEOF (0xffffffffu)

#define WCHAR\_MAX 0x7FFFFFFF

#define WCHAR\_MIN 0x80000000

typedef unsigned long int wctype\_t;

typedef const int32\_t \*wctrans\_t;

extern wchar\_t \*\_\_fgetws\_chk(wchar\_t \*, size\_t, int, FILE \*);

extern wchar\_t \*\_\_fgetws\_unlocked\_chk(wchar\_t \*, size\_t, int, FILE \*);

extern int \_\_fwprintf\_chk(FILE \*, int, const wchar\_t \*, ...);

extern size\_t \_\_mbsnrtowcs\_chk(wchar\_t \*, const char \*\*, size\_t, size\_t,

mbstate\_t \*, size\_t);

extern size\_t \_\_mbsrtowcs\_chk(wchar\_t \*, const char \*\*, size\_t,

mbstate\_t \*, size\_t);

extern int \_\_swprintf\_chk(wchar\_t \*, size\_t, int, size\_t, const wchar\_t \*,

...);

extern int \_\_vfwprintf\_chk(FILE \*, int, const wchar\_t \*, va\_list);

extern int \_\_vswprintf\_chk(wchar\_t \*, size\_t, int, size\_t, const wchar\_t \*,

va\_list);

extern int \_\_vwprintf\_chk(int, const wchar\_t \*, va\_list);

extern wchar\_t \*\_\_wcpcpy\_chk(wchar\_t \*, const wchar\_t \*, size\_t);

extern wchar\_t \*\_\_wcpncpy\_chk(wchar\_t \*, const wchar\_t \*, size\_t, size\_t);

extern size\_t \_\_wcrtomb\_chk(char \*, wchar\_t, mbstate\_t \*, size\_t);

extern wchar\_t \*\_\_wcscat\_chk(wchar\_t \*, const wchar\_t \*, size\_t);

extern wchar\_t \*\_\_wcscpy\_chk(wchar\_t \*, const wchar\_t \*, size\_t);

extern wchar\_t \*\_\_wcsncat\_chk(wchar\_t \*, const wchar\_t \*, size\_t, size\_t);

extern wchar\_t \*\_\_wcsncpy\_chk(wchar\_t \*, const wchar\_t \*, size\_t, size\_t);

extern size\_t \_\_wcsnrtombs\_chk(char \*, const wchar\_t \* \*, size\_t, size\_t,

mbstate\_t \*, size\_t);

extern size\_t \_\_wcsrtombs\_chk(char \*, const wchar\_t \* \*, size\_t,

mbstate\_t \*, size\_t);

extern double \_\_wcstod\_internal(const wchar\_t \*, wchar\_t \* \*, int);

extern float \_\_wcstof\_internal(const wchar\_t \*, wchar\_t \* \*, int);

extern long int \_\_wcstol\_internal(const wchar\_t \*, wchar\_t \* \*, int, int);

extern long double \_\_wcstold\_internal(const wchar\_t \*, wchar\_t \* \*, int);

extern unsigned long int \_\_wcstoul\_internal(const wchar\_t \*, wchar\_t \* \*,

int, int);

extern wchar\_t \*\_\_wmemcpy\_chk(wchar\_t \*, const wchar\_t \*, size\_t, size\_t);

extern wchar\_t \*\_\_wmemmove\_chk(wchar\_t \*, const wchar\_t \*, size\_t, size\_t);

extern wchar\_t \*\_\_wmempcpy\_chk(wchar\_t \*, const wchar\_t \*, size\_t, size\_t);

extern wchar\_t \*\_\_wmemset\_chk(wchar\_t \*, wchar\_t, size\_t, size\_t);

extern int \_\_wprintf\_chk(int, const wchar\_t \*, ...);

extern wint\_t btowc(int \_\_c);

extern wint\_t fgetwc(FILE \* \_\_stream);

extern wint\_t fgetwc\_unlocked(FILE \* \_\_stream);

extern wchar\_t \*fgetws(wchar\_t \* \_\_ws, int \_\_n, FILE \* \_\_stream);

extern wchar\_t \*fgetws\_unlocked(wchar\_t \* \_\_ws, int \_\_n, FILE \* \_\_stream);

extern wint\_t fputwc(wchar\_t \_\_wc, FILE \* \_\_stream);

extern wint\_t fputwc\_unlocked(wchar\_t \_\_wc, FILE \* \_\_stream);

extern int fputws(const wchar\_t \* \_\_ws, FILE \* \_\_stream);

extern int fputws\_unlocked(const wchar\_t \* \_\_ws, FILE \* \_\_stream);

extern int fwide(FILE \* \_\_fp, int \_\_mode);

extern int fwprintf(FILE \* \_\_stream, const wchar\_t \* \_\_format, ...);

extern int fwscanf(FILE \* \_\_stream, const wchar\_t \* \_\_format, ...);

extern wint\_t getwc(FILE \* \_\_stream);

extern wint\_t getwc\_unlocked(FILE \* \_\_stream);

extern wint\_t getwchar(void);

extern wint\_t getwchar\_unlocked(void);

extern size\_t mbrlen(const char \*\_\_s, size\_t \_\_n, mbstate\_t \* \_\_ps);

extern size\_t mbrtowc(wchar\_t \* \_\_pwc, const char \*\_\_s, size\_t \_\_n,

mbstate\_t \* \_\_p);

extern int mbsinit(const mbstate\_t \* \_\_ps);

extern size\_t mbsnrtowcs(wchar\_t \* \_\_dst, const char \*\*\_\_src, size\_t \_\_nmc,

size\_t \_\_len, mbstate\_t \* \_\_ps);

extern size\_t mbsrtowcs(wchar\_t \* \_\_dst, const char \*\*\_\_src, size\_t \_\_len,

mbstate\_t \* \_\_ps);

extern FILE \*open\_wmemstream(wchar\_t \* \*\_\_bufloc, size\_t \* \_\_sizeloc);

extern wint\_t putwc(wchar\_t \_\_wc, FILE \* \_\_stream);

extern wint\_t putwc\_unlocked(wchar\_t \_\_wc, FILE \* \_\_stream);

extern wint\_t putwchar(wchar\_t \_\_wc);

extern wint\_t putwchar\_unlocked(wchar\_t \_\_wc);

extern int swprintf(wchar\_t \* \_\_s, size\_t \_\_n, const wchar\_t \* \_\_format,

...);

extern int swscanf(const wchar\_t \* \_\_s, const wchar\_t \* \_\_format, ...);

extern wint\_t ungetwc(wint\_t \_\_wc, FILE \* \_\_stream);

extern int vfwprintf(FILE \* \_\_s, const wchar\_t \* \_\_format, va\_list \_\_arg);

extern int vfwscanf(FILE \* \_\_s, const wchar\_t \* \_\_format, va\_list \_\_arg);

extern int vswprintf(wchar\_t \* \_\_s, size\_t \_\_n, const wchar\_t \* \_\_format,

va\_list \_\_arg);

extern int vswscanf(const wchar\_t \* \_\_s, const wchar\_t \* \_\_format,

va\_list \_\_arg);

extern int vwprintf(const wchar\_t \* \_\_format, va\_list \_\_arg);

extern int vwscanf(const wchar\_t \* \_\_format, va\_list \_\_arg);

extern wchar\_t \*wcpcpy(wchar\_t \* \_\_dest, const wchar\_t \* \_\_src);

extern wchar\_t \*wcpncpy(wchar\_t \* \_\_dest, const wchar\_t \* \_\_src,

size\_t \_\_n);

extern size\_t wcrtomb(char \*\_\_s, wchar\_t \_\_wc, mbstate\_t \* \_\_ps);

extern int wcscasecmp(const wchar\_t \* \_\_s1, const wchar\_t \* \_\_s2);

extern int wcscasecmp\_l(const wchar\_t \* ws1, const wchar\_t \* ws2,

locale\_t locale);

extern wchar\_t \*wcscat(wchar\_t \* \_\_dest, const wchar\_t \* \_\_src);

extern wchar\_t \*wcschr(const wchar\_t \* \_\_wcs, wchar\_t \_\_wc);

extern int wcscmp(const wchar\_t \* \_\_s1, const wchar\_t \* \_\_s2);

extern int wcscoll(const wchar\_t \* \_\_s1, const wchar\_t \* \_\_s2);

extern int wcscoll\_l(const wchar\_t \* ws1, const wchar\_t \* ws2,

locale\_t locale);

extern wchar\_t \*wcscpy(wchar\_t \* \_\_dest, const wchar\_t \* \_\_src);

extern size\_t wcscspn(const wchar\_t \* \_\_wcs, const wchar\_t \* \_\_reject);

extern wchar\_t \*wcsdup(const wchar\_t \* \_\_s);

extern size\_t wcsftime(wchar\_t \* \_\_s, size\_t \_\_maxsize,

const wchar\_t \* \_\_format, const struct tm \*\_\_tp);

extern size\_t wcslen(const wchar\_t \* \_\_s);

extern int wcsncasecmp(const wchar\_t \* \_\_s1, const wchar\_t \* \_\_s2,

size\_t \_\_n);

extern int wcsncasecmp\_l(const wchar\_t \* ws1, const wchar\_t \* ws2,

size\_t n, locale\_t locale);

extern wchar\_t \*wcsncat(wchar\_t \* \_\_dest, const wchar\_t \* \_\_src,

size\_t \_\_n);

extern int wcsncmp(const wchar\_t \* \_\_s1, const wchar\_t \* \_\_s2, size\_t \_\_n);

extern wchar\_t \*wcsncpy(wchar\_t \* \_\_dest, const wchar\_t \* \_\_src,

size\_t \_\_n);

extern size\_t wcsnlen(const wchar\_t \* \_\_s, size\_t \_\_maxlen);

extern size\_t wcsnrtombs(char \*\_\_dst, const wchar\_t \* \*\_\_src, size\_t \_\_nwc,

size\_t \_\_len, mbstate\_t \* \_\_ps);

extern wchar\_t \*wcspbrk(const wchar\_t \* \_\_wcs, const wchar\_t \* \_\_accept);

extern wchar\_t \*wcsrchr(const wchar\_t \* \_\_wcs, wchar\_t \_\_wc);

extern size\_t wcsrtombs(char \*\_\_dst, const wchar\_t \* \*\_\_src, size\_t \_\_len,

mbstate\_t \* \_\_ps);

extern size\_t wcsspn(const wchar\_t \* \_\_wcs, const wchar\_t \* \_\_accept);

extern wchar\_t \*wcsstr(const wchar\_t \* \_\_haystack,

const wchar\_t \* \_\_needle);

extern double wcstod(const wchar\_t \* \_\_nptr, wchar\_t \* \*\_\_endptr);

extern float wcstof(const wchar\_t \* \_\_nptr, wchar\_t \* \*\_\_endptr);

extern wchar\_t \*wcstok(wchar\_t \* \_\_s, const wchar\_t \* \_\_delim,

wchar\_t \* \*\_\_ptr);

extern long int wcstol(const wchar\_t \* \_\_nptr, wchar\_t \* \*\_\_endptr,

int \_\_base);

extern long double wcstold(const wchar\_t \* \_\_nptr, wchar\_t \* \*\_\_endptr);

extern long long int wcstoll(const wchar\_t \* \_\_nptr, wchar\_t \* \*\_\_endptr,

int \_\_base);

extern long long int wcstoq(const wchar\_t \* \_\_nptr, wchar\_t \* \*\_\_endptr,

int \_\_base);

extern unsigned long int wcstoul(const wchar\_t \* \_\_nptr,

wchar\_t \* \*\_\_endptr, int \_\_base);

extern unsigned long long int wcstoull(const wchar\_t \* \_\_nptr,

wchar\_t \* \*\_\_endptr, int \_\_base);

extern unsigned long long int wcstouq(const wchar\_t \* \_\_nptr,

wchar\_t \* \*\_\_endptr, int \_\_base);

extern wchar\_t \*wcswcs(const wchar\_t \* \_\_haystack,

const wchar\_t \* \_\_needle);

extern int wcswidth(const wchar\_t \* \_\_s, size\_t \_\_n);

extern size\_t wcsxfrm(wchar\_t \* \_\_s1, const wchar\_t \* \_\_s2, size\_t \_\_n);

extern size\_t wcsxfrm\_l(const wchar\_t \* ws1, const wchar\_t \* ws2, size\_t n,

locale\_t locale);

extern int wctob(wint\_t \_\_c);

extern int wcwidth(wchar\_t \_\_c);

extern wchar\_t \*wmemchr(const wchar\_t \* \_\_s, wchar\_t \_\_c, size\_t \_\_n);

extern int wmemcmp(const wchar\_t \* \_\_s1, const wchar\_t \* \_\_s2, size\_t \_\_n);

extern wchar\_t \*wmemcpy(wchar\_t \* \_\_s1, const wchar\_t \* \_\_s2, size\_t \_\_n);

extern wchar\_t \*wmemmove(wchar\_t \* \_\_s1, const wchar\_t \* \_\_s2, size\_t \_\_n);

extern wchar\_t \*wmemset(wchar\_t \* \_\_s, wchar\_t \_\_c, size\_t \_\_n);

extern int wprintf(const wchar\_t \* \_\_format, ...);

extern int wscanf(const wchar\_t \* \_\_format, ...);

### **14.4.108 wctype.h**

extern int iswalnum(wint\_t \_\_wc);

extern int iswalnum\_l(wint\_t wc, locale\_t locale);

extern int iswalpha(wint\_t \_\_wc);

extern int iswalpha\_l(wint\_t wc, locale\_t locale);

extern int iswblank(wint\_t \_\_wc);

extern int iswblank\_l(wint\_t wc, locale\_t locale);

extern int iswcntrl(wint\_t \_\_wc);

extern int iswcntrl\_l(wint\_t wc, locale\_t locale);

extern int iswctype(wint\_t \_\_wc, wctype\_t \_\_desc);

extern int iswctype\_l(wint\_t wc, locale\_t locale);

extern int iswdigit(wint\_t \_\_wc);

extern int iswdigit\_l(wint\_t wc, locale\_t locale);

extern int iswgraph(wint\_t \_\_wc);

extern int iswgraph\_l(wint\_t wc, locale\_t locale);

extern int iswlower(wint\_t \_\_wc);

extern int iswlower\_l(wint\_t wc, locale\_t locale);

extern int iswprint(wint\_t \_\_wc);

extern int iswprint\_l(wint\_t wc, locale\_t locale);

extern int iswpunct(wint\_t \_\_wc);

extern int iswpunct\_l(wint\_t wc, locale\_t locale);

extern int iswspace(wint\_t \_\_wc);

extern int iswspace\_l(wint\_t wc, locale\_t locale);

extern int iswupper(wint\_t \_\_wc);

extern int iswupper\_l(wint\_t wc, locale\_t locale);

extern int iswxdigit(wint\_t \_\_wc);

extern int iswxdigit\_l(wint\_t wc, locale\_t locale);

extern wint\_t towctrans(wint\_t \_\_wc, wctrans\_t \_\_desc);

extern wint\_t towctrans\_l(wint\_t wc, wctrans\_t desc, locale\_t locale);

extern wint\_t towlower(wint\_t \_\_wc);

extern wint\_t towlower\_l(wint\_t wc, locale\_t locale);

extern wint\_t towupper(wint\_t \_\_wc);

extern wint\_t towupper\_l(wint\_t wc, locale\_t locale);

extern wctrans\_t wctrans(const char \*\_\_property);

extern size\_t wctrans\_l(const char \*charclass, locale\_t locale);

extern wctype\_t wctype(const char \*\_\_property);

extern size\_t wctype\_l(const char \*property, locale\_t locale);

### **14.4.109 wordexp.h**

enum {

WRDE\_DOOFFS = 1,

WRDE\_APPEND = 2,

WRDE\_NOCMD = 4,

WRDE\_REUSE = 8,

WRDE\_SHOWERR = 16,

WRDE\_UNDEF = 32

};

typedef struct {

size\_t we\_wordc;

char \*\*we\_wordv;

size\_t we\_offs;

} wordexp\_t;

enum {

WRDE\_NOSYS = -1,

WRDE\_NOSPACE = 1,

WRDE\_BADCHAR = 2,

WRDE\_BADVAL = 3,

WRDE\_CMDSUB = 4,

WRDE\_SYNTAX = 5

};

extern int wordexp(const char \*\_\_words, wordexp\_t \* \_\_pwordexp,

int \_\_flags);

extern void wordfree(wordexp\_t \* \_\_wordexp);

## **14.5 Interface Definitions for libc**

The interfaces defined on the following pages are included in libc and are defined by this specification. Unless otherwise noted, these interfaces shall be included in the source standard.

Other interfaces listed in [Section 14.3](#ID_LIBC) shall behave as described in the referenced base document.

#### \_IO\_feof

##### Name

\_IO\_feof — alias for feof

##### Synopsis

int \_IO\_feof(\_IO\_FILE \* *\_\_fp*);

##### Description

\_IO\_feof() tests the end-of-file indicator for the stream pointed to by *\_\_fp*, returning a non-zero value if it is set.

\_IO\_feof() is not in the source standard; it is only in the binary standard.

#### \_IO\_getc

##### Name

\_IO\_getc — alias for getc

##### Synopsis

int \_IO\_getc(\_IO\_FILE \* *\_\_fp*);

##### Description

\_IO\_getc() reads the next character from *\_\_fp* and returns it as an unsigned char cast to an int, or EOF on end-of-file or error.

\_IO\_getc() is not in the source standard; it is only in the binary standard.

#### \_IO\_putc

##### Name

\_IO\_putc — alias for putc

##### Synopsis

int \_IO\_putc(int *\_\_c*, \_IO\_FILE \* *\_\_fp*);

##### Description

\_IO\_putc() writes the character *\_\_c*, cast to an unsigned char, to *\_\_fp*.

\_IO\_putc() is not in the source standard; it is only in the binary standard.

#### \_IO\_puts

##### Name

\_IO\_puts — alias for puts

##### Synopsis

int \_IO\_puts(const char \* *\_\_c*);

##### Description

\_IO\_puts() writes the string *\_\_s* and a trailing newline to stdout.

\_IO\_puts() is not in the source standard; it is only in the binary standard.

#### \_\_assert\_fail

##### Name

\_\_assert\_fail — abort the program after false assertion

##### Synopsis

void \_\_assert\_fail(const char \* *assertion*, const char \* *file*, unsigned int *line*, const char \* *function*);

##### Description

The \_\_assert\_fail() function is used to implement the assert() interface of [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4). The \_\_assert\_fail() function shall print the given *file* filename, *line* line number, *function* function name and a message on the standard error stream in an unspecified format, and abort program execution via the abort() function. For example:

a.c:10: foobar: Assertion a == b failed.

If *function* is NULL, \_\_assert\_fail() shall omit information about the function.

*assertion*, *file*, and *line* shall be non-NULL.

The \_\_assert\_fail() function is not in the source standard; it is only in the binary standard. The assert() interface is not in the binary standard; it is only in the source standard. The assert() may be implemented as a macro.

#### \_\_chk\_fail

##### Name

\_\_chk\_fail — terminate a function in case of buffer overflow

##### Synopsis

void \_\_chk\_fail(void);

##### Description

The interface \_\_chk\_fail() shall abort the function that called it with a message that a buffer overflow has been detected. The program that called the function shall then exit.

The \_\_chk\_fail() function is not in the source standard; it is only in the binary standard.

##### Application Usage (informative)

The interface \_\_chk\_fail() does not check for a buffer overflow itself. It merely reports one when invoked.

#### \_\_confstr\_chk

##### Name

\_\_confstr\_chk — get configuration dependent string variables, with buffer overflow checking

##### Synopsis

#include <unistd.h>

size\_t \_\_confstr\_chk(int *name*, char \* *buf*, size\_t *len*, size\_t *buflen*);

##### Description

The interface \_\_confstr\_chk() shall function in the same way as the interface confstr(), except that \_\_confstr\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *buflen* specifies the size of the buffer *buf*. If *len* exceeds *buflen*, the function shall abort, and the program calling it shall exit.

The \_\_confstr\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_ctype\_b\_loc

##### Name

\_\_ctype\_b\_loc — accessor function for \_\_ctype\_b array for ctype functions

##### Synopsis

#include <ctype.h>

const unsigned short \* \* \_\_ctype\_b\_loc (void);

##### Description

The \_\_ctype\_b\_loc() function shall return a pointer into an array of characters in the current locale that contains characteristics for each character in the current character set. The array shall contain a total of 384 characters, and can be indexed with any signed or unsigned char (i.e. with an index value between -128 and 255). If the application is multithreaded, the array shall be local to the current thread.

This interface is not in the source standard; it is only in the binary standard.

##### Return Value

The \_\_ctype\_b\_loc() function shall return a pointer to the array of characters to be used for the ctype() family of functions (see <ctype.h>).

#### \_\_ctype\_get\_mb\_cur\_max

##### Name

\_\_ctype\_get\_mb\_cur\_max — maximum length of a multibyte character in the current locale

##### Synopsis

size\_t \_\_ctype\_get\_mb\_cur\_max(void);

##### Description

\_\_ctype\_get\_mb\_cur\_max() returns the maximum length of a multibyte character in the current locale.

\_\_ctype\_get\_mb\_cur\_max() is not in the source standard; it is only in the binary standard.

#### \_\_ctype\_tolower\_loc

##### Name

\_\_ctype\_tolower\_loc — accessor function for \_\_ctype\_b\_tolower array for ctype tolower() function

##### Synopsis

#include <ctype.h>

int32\_t \* \* \_\_ctype\_tolower\_loc(void);

##### Description

The \_\_ctype\_tolower\_loc() function shall return a pointer into an array of characters in the current locale that contains lower case equivalents for each character in the current character set. The array shall contain a total of 384 characters, and can be indexed with any signed or unsigned char (i.e. with an index value between -128 and 255). If the application is multithreaded, the array shall be local to the current thread.

This interface is not in the source standard; it is only in the binary standard.

##### Return Value

The \_\_ctype\_tolower\_loc() function shall return a pointer to the array of characters to be used for the ctype() family of functions (see <ctype.h>).

#### \_\_ctype\_toupper\_loc

##### Name

\_\_ctype\_toupper\_loc — accessor function for \_\_ctype\_b\_toupper() array for ctype toupper() function

##### Synopsis

#include <ctype.h>

int32\_t \* \* \_\_ctype\_toupper\_loc(void);

##### Description

The \_\_ctype\_toupper\_loc() function shall return a pointer into an array of characters in the current locale that contains upper case equivalents for each character in the current character set. The array shall contain a total of 384 characters, and can be indexed with any signed or unsigned char (i.e. with an index value between -128 and 255). If the application is multithreaded, the array shall be local to the current thread.

This interface is not in the source standard; it is only in the binary standard.

##### Return Value

The \_\_ctype\_toupper\_loc() function shall return a pointer to the array of characters to be used for the ctype() family of functions (see <ctype.h>).

#### \_\_cxa\_atexit

##### Name

\_\_cxa\_atexit — register a function to be called by exit or when a shared library is unloaded

##### Synopsis

int \_\_cxa\_atexit(void (\**func*) (void \*), void \* *arg*, void \* *dso\_handle*);

##### Description

As described in the [Itanium™ C++ ABI](#ID_STD_46_CXXABI), \_\_cxa\_atexit() registers a destructor function to be called by exit() or when a shared library is unloaded. When a shared library is unloaded, any destructor function associated with that shared library, identified by *dso\_handle*, shall be called with the single argument *arg*, and then that function shall be removed, or marked as complete, from the list of functions to run at exit(). On a call to exit(), any remaining functions registered shall be called with the single argument *arg*. Destructor functions shall always be called in the reverse order to their registration (i.e. the most recently registered function shall be called first),

The \_\_cxa\_atexit() function is used to implement atexit(), as described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4). Calling atexit(func) from the statically linked part of an application shall be equivalent to \_\_cxa\_atexit(func, NULL, NULL).

\_\_cxa\_atexit() is not in the source standard; it is only in the binary standard.

**Note:** atexit() is not in the binary standard; it is only in the source standard.

#### \_\_cxa\_finalize

##### Name

\_\_cxa\_finalize — call destructors of global (or local static) C++ objects and exit functions registered with atexit

##### Synopsis

void \_\_cxa\_finalize(void \* *d*);

##### Description

As described in the Itanium® C++ ABI, the C runtime library shall maintain a list of termination function entries containing the following information:

• A pointer to a termination function.

• An operand to be passed to the function.

• A handle identifying the home shared library of the entry.

The list is populated by entries of two kinds:

• Destructors of global (or local static) C++ objects that require destruction on exit.

• Functions registered by the user with atexit().

In the former case an entry consists of a pointer to the destructor, a pointer to the corresponding object and a handle for the home shared library of the object. In the latter case the pointer to the function is the pointer passed to atexit(), while the other pointers are NULL.

When \_\_cxa\_finalize(d) is called, it shall walk the termination function list, calling each in turn if d matches the handle of the termination function entry. If d is NULL, it shall call all the termination funtions. Multiple calls to \_\_cxa\_finalize shall not result in calling termination function entries multiple times; the implementation may either remove entries or mark them finished. The termination functions shall always be called in the reverse order of their registration (i.e. the most recently registered function shall be called first).

An application shall not call \_\_cxa\_finalize() directly. The implementation shall arrange for\_\_cxa\_finalize() to be called during early shared library unload (e.g. dlclose()) with a handle to the shared library. When the main program calls exit, the implementation shall cause any remaining \_\_cxa\_atexit-registered functions to be called, either by calling \_\_cxa\_finalize(NULL), or by walking the registration list itself.

\_\_cxa\_finalize() is not in the source standard; it is only in the binary standard.

#### \_\_daylight

##### Name

\_\_daylight — external daylight savings time flag

##### Synopsis

int \_\_daylight;

##### Description

The external variable \_\_daylight shall implement the daylight savings time flag daylight as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4). \_\_daylight has the same specification as daylight.

#### \_\_environ

##### Name

\_\_environ — alias for environ - user environment

##### Synopsis

extern char \*\*\_\_environ;

##### Description

The external variable \_\_environ shall implement the environment variable environ as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4). \_\_environ has the same specification as environ.

#### \_\_errno\_location

##### Name

\_\_errno\_location — address of errno variable

##### Synopsis

int \* \_\_errno\_location(void);

##### Description

The \_\_errno\_location() function shall return the address of the errno variable for the current thread.

\_\_errno\_location() is not in the source standard; it is only in the binary standard.

#### \_\_fgets\_chk

##### Name

\_\_fgets\_chk — string input, with buffer overflow checking

##### Synopsis

#include <stdio.h>

char \* \_\_fgets\_chk(char \* *s*, size\_t *size*, int *strsize*, FILE \* *stream*);

##### Description

The interface \_\_fgets\_chk() shall function in the same way as the interface fgets(), except that \_\_fgets\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *strsize* specifies the size of the object pointed to by *stream*.

The \_\_fgets\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_fgets\_unlocked\_chk

##### Name

\_\_fgets\_unlocked\_chk — non-locking string input, with buffer overflow checking

##### Synopsis

#include <stdio.h>

char \* \_\_fgets\_unlocked\_chk(char \* *s*, size\_t *size*, int *strsize*, FILE \* *stream*);

##### Description

The interface \_\_fgets\_unlocked\_chk() shall function in the same way as the interface fgets\_unlocked(), except that \_\_fgets\_unlocked\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *strsize* specifies the size of the object pointed to by *stream*.

The \_\_fgets\_unlocked\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_fgetws\_chk

##### Name

\_\_fgetws\_chk — read a wide-character string from a FILE stream, with buffer overflow checking

##### Synopsis

#include <wchar.h>

wchar\_t \* \_\_fgetws\_chk(wchar\_t \* *ws*, size\_t *size*, int *strsize*, FILE \* *stream*);

##### Description

The interface \_\_fgetws\_chk() shall function in the same way as the interface fgetws(), except that \_\_fgetws\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *strsize* specifies the size of the object pointed to by *stream*.

The \_\_fgetws\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_fgetws\_unlocked\_chk

##### Name

\_\_fgetws\_unlocked\_chk — read a wide-character string from a FILE stream in a non-locking manner, with stack checking

##### Synopsis

#include <wchar.h>

wchar\_t \* \_\_fgetws\_unlocked\_chk(wchar\_t \* *ws*, size\_t *strsize*, int *n*, FILE \* *stream*);

##### Description

The interface \_\_fgetws\_unlocked\_chk() shall function in the same way as the interface fgetws\_unlocked(), except that \_\_fgetws\_unlocked\_chk() shall check for stack overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *strsize* specifies the size of the object pointed to by *stream*.

The \_\_fgetws\_unlocked\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_fpending

##### Name

\_\_fpending — returns in bytes the amount of output pending on a stream

##### Synopsis

size\_t \_\_fpending(FILE \* *stream*);

##### Description

\_\_fpending() returns the amount of output in bytes pending on a stream.

\_\_fpending() is not in the source standard; it is only in the binary standard.

#### \_\_fprintf\_chk

##### Name

\_\_fprintf\_chk — convert formatted output, with stack checking

##### Synopsis

#include <stdio.h>

int \_\_fprintf\_chk(FILE \* *stream*, int *flag*, const char \* *format*);

##### Description

The interface \_\_fprintf\_chk() shall function in the same way as the interface fprintf(), except that \_\_fprintf\_chk() shall check for stack overflow before computing a result, depending on the value of the *flag* parameter. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

In general, the higher the value of *flag*, the more security measures this interface shall take in the form of checking the stack, parameter values, and so on.

The \_\_fprintf\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_fwprintf\_chk

##### Name

\_\_fwprintf\_chk — convert formatted wide-character output, with stack checking

##### Synopsis

#include <wchar.h>

int \_\_fwprintf\_chk(FILE \* *stream*, int *flag*, const wchar\_t \* *format*);

##### Description

The interface \_\_fwprintf\_chk() shall function in the same way as the interface fwprintf(), except that \_\_fwprintf\_chk() shall check for stack overflow before computing a result, depending on the value of the *flag* parameter. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

In general, the higher the value of *flag*, the more security measures this interface shall take in the form of checking the stack, parameter values, and so on.

The \_\_fwprintf\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_fxstatat

##### Name

\_\_fxstatat — get file status relative to directory file descriptor

##### Synopsis

#include <fcntl.h>

#include <sys/stat.h>

int \_\_fxstatat(int *ver*, int *dirfd*, const char \* *path*, struct stat \* *stat\_buf*, int *flags*);

##### Description

The \_\_fxstatat() function shall implement the fstatat() function. The behavior of \_\_fxstatat() for values of *ver* other than \_STAT\_VER is undefined. See Data Definitions in the architecture specific part of this specification for the correct value of \_STAT\_VER.

\_\_fxstatat(\_STAT\_VER, *dirfd*, *stat\_buf*, *flags*) shall behave as fstatat(*dirfd*, *stat\_buf*, *flags*) as specified by [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

\_\_fxstatat() is not in the source standard; it is only in the binary standard.

**Note:** The fstatat() function is not in the binary standard; it is only in the source standard.

#### \_\_fxstatat64, fstatat64

##### Name

\_\_fxstatat64, fstatat64 — get file status relative to a directory file descriptor (Large File Support)

##### Synopsis

#include <fcntl.h>

#include <sys/stat.h>

int \_\_fxstatat64(int *ver*, int *dirfd*, const char \* *path*, struct stat64 \* *stat\_buf*, int *flags*);

int fstatat64(int *dirfd*, const char \* *file*, struct stat64 \* *buf*, int *flag*);

##### Description

fstatat64() is a large-file version of the fstatat() function as defined in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4). It differs from fstatat() only in that the *buf* parameter refers to a large-file version of the stat structure.

The \_\_fxstatat64() function shall implement the fstatat64() function. The behavior of \_\_fxstatat64() for values of *ver* other than \_STAT\_VER is undefined. See Data Definitions in the architecture specific part of this specification for the correct value of \_STAT\_VER.

\_\_fxstatat64(\_STAT\_VER, *dirfd*, *stat\_buf*, *flags*) shall behave as fstatat64(*dirfd*, *stat\_buf*, *flags*)

\_\_fxstatat64() is not in the source standard; it is only in the binary standard.

**Note:** The fstatat64() function is not in the binary standard; it is only in the source standard.

#### \_\_getcwd\_chk

##### Name

\_\_getcwd\_chk — get current working directory, with buffer overflow checking

##### Synopsis

#include <unistd.h>

char \* \_\_getcwd\_chk(char \* *buf*, size\_t *len*, size\_t *buflen*);

##### Description

The interface \_\_getcwd\_chk() shall function in the same way as the interface getcwd(), except that \_\_getcwd\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *buflen* specifies the size of the buffer *buf*. If *len* exceeds *buflen*, the function shall abort, and the program calling it shall exit.

The \_\_getcwd\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_getgroups\_chk

##### Name

\_\_getgroups\_chk — get list of supplementary group IDs, with buffer overflow checking

##### Synopsis

#include <unistd.h>

int \_\_getgroups\_chk(int *size*, gid\_t \* *list*, size\_t *listlen*);

##### Description

The interface \_\_getgroups\_chk() shall function in the same way as the interface getgroups(), except that \_\_getgroups\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *listlen* specifies the size in bytes of the object *list*.

The \_\_getgroups\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_gethostname\_chk

##### Name

\_\_gethostname\_chk — get host name, with buffer overflow checking

##### Synopsis

#include <unistd.h>

int \_\_gethostname\_chk(char \* *buf*, size\_t *buflen*, size\_t *maxlen*);

##### Description

The interface \_\_gethostname\_chk() shall function in the same way as the interface gethostname(), except that \_\_gethostname\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *buflen* specifies the size of the buffer *buf*. If *buflen* exceeds *maxlen*, the function shall abort, and the program calling it shall exit.

The \_\_gethostname\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_getlogin\_r\_chk

##### Name

\_\_getlogin\_r\_chk — get user name, with buffer overflow checking (reentrant)

##### Synopsis

#include <unistd.h>

int \_\_getlogin\_r\_chk(char \* *buf*, size\_t *buflen*, size\_t *maxlen*);

##### Description

The interface \_\_getlogin\_r\_chk() shall function in the same way as the interface getlogin\_r(), except that \_\_getlogin\_r\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *buflen* specifies the size of the buffer *buf*. If *buflen* exceeds *maxlen*, the function shall abort, and the program calling it shall exit.

The \_\_getlogin\_r\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_getpagesize

##### Name

\_\_getpagesize — alias for getpagesize - get current page size

##### Synopsis

int \_\_getpagesize(void);

##### Description

\_\_getpagesize() is an alias for getpagesize() - get current page size.

\_\_getpagesize() has the same specification as getpagesize().

\_\_getpagesize() is not in the source standard; it is only in the binary standard.

#### \_\_getpgid

##### Name

\_\_getpgid — get the process group id

##### Synopsis

pid\_t \_\_getpgid(pid\_t *pid*);

##### Description

\_\_getpgid() has the same specification as getpgid().

\_\_getpgid() is not in the source standard; it is only in the binary standard.

#### \_\_h\_errno\_location

##### Name

\_\_h\_errno\_location — address of h\_errno variable

##### Synopsis

int \* \_\_h\_errno\_location(void);

##### Description

\_\_h\_errno\_location() returns the address of the h\_errno variable, where h\_errno is as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

\_\_h\_errno\_location() is not in the source standard; it is only in the binary standard. Note that h\_errno itself is only in the source standard; it is not in the binary standard.

#### \_\_isinf

##### Name

\_\_isinf — test for infinity

##### Synopsis

int \_\_isinf(double *arg*);

##### Description

\_\_isinf() has the same specification as isinf() in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except that the argument type for \_\_isinf() is known to be double.

\_\_isinf() is not in the source standard; it is only in the binary standard.

#### \_\_isinff

##### Name

\_\_isinff — test for infinity

##### Synopsis

int \_\_isinff(float *arg*);

##### Description

\_\_isinff() has the same specification as isinf() in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) except that the argument type for \_\_isinff() is known to be float.

\_\_isinff() is not in the source standard; it is only in the binary standard.

#### \_\_isinfl

##### Name

\_\_isinfl — test for infinity

##### Synopsis

int \_\_isinfl(long double *arg*);

##### Description

\_\_isinfl() has the same specification as isinf() in the [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except that the argument type for \_\_isinfl() is known to be long double.

\_\_isinfl() is not in the source standard; it is only in the binary standard.

#### \_\_isnan

##### Name

\_\_isnan — test for infinity

##### Synopsis

int \_\_isnan(double *arg*);

##### Description

\_\_isnan() has the same specification as isnan() in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except that the argument type for \_\_isnan() is known to be double.

\_\_isnan() is not in the source standard; it is only in the binary standard.

#### \_\_isnanf

##### Name

\_\_isnanf — test for infinity

##### Synopsis

int \_\_isnanf(float *arg*);

##### Description

\_\_isnanf() has the same specification as isnan() in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except that the argument type for \_\_isnanf() is known to be float.

\_\_isnanf() is not in the source standard; it is only in the binary standard.

#### \_\_isnanl

##### Name

\_\_isnanl — test for infinity

##### Synopsis

int \_\_isnanl(long double *arg*);

##### Description

\_\_isnanl() has the same specification as isnan() in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except that the argument type for \_\_isnanl() is known to be long double.

\_\_isnanl() is not in the source standard; it is only in the binary standard.

#### \_\_libc\_current\_sigrtmax

##### Name

\_\_libc\_current\_sigrtmax — return number of available real-time signal with lowest priority

##### Synopsis

int \_\_libc\_current\_sigrtmax(void);

##### Description

\_\_libc\_current\_sigrtmax() returns the number of an available real-time signal with the lowest priority.

\_\_libc\_current\_sigrtmax() is not in the source standard; it is only in the binary standard.

#### \_\_libc\_current\_sigrtmin

##### Name

\_\_libc\_current\_sigrtmin — return number of available real-time signal with highest priority

##### Synopsis

int \_\_libc\_current\_sigrtmin(void);

##### Description

\_\_libc\_current\_sigrtmin() returns the number of an available real-time signal with the highest priority.

\_\_libc\_current\_sigrtmin() is not in the source standard; it is only in the binary standard.

#### \_\_libc\_start\_main

##### Name

\_\_libc\_start\_main — initialization routine

##### Synopsis

int \_\_libc\_start\_main(int (\**main*) (int, char \*\*, char \*\*), int *argc*, char \*\* *ubp\_av*, void (\**init*) (void), void (\**fini*) (void), void (\**rtld\_fini*) (void), void (\**stack\_end*));

##### Description

The \_\_libc\_start\_main() function shall perform any necessary initialization of the execution environment, call the *main* function with appropriate arguments, and handle the return from main(). If the main() function returns, the return value shall be passed to the exit() function.

**Note:** While this specification is intended to be implementation independent, process and library initialization may include:

• performing any necessary security checks if the effective user ID is not the same as the real user ID.

• initialize the threading subsystem.

• registering the *rtld\_fini* to release resources when this dynamic shared object exits (or is unloaded).

• registering the *fini* handler to run at program exit.

• calling the initializer function (\**init*)().

• calling main() with appropriate arguments.

• calling exit() with the return value from main().

This list is an example only.

\_\_libc\_start\_main() is not in the source standard; it is only in the binary standard.

##### See Also

The section on Process Initialization in each of the architecture specific parts of the LSB Core Specification.

#### \_\_mbsnrtowcs\_chk

##### Name

\_\_mbsnrtowcs\_chk — convert a multibyte string to a wide-character string, with buffer overflow checking

##### Synopsis

#include <wchar.h>

size\_t \_\_mbsnrtowcs\_chk(wchar\_t \* *dest*, const char \* \* *src*, size\_t *nmc*, size\_t *len*, mbstate\_t \* *ps*, size\_t *destlen*);

##### Description

The interface \_\_mbsnrtowcs\_chk() shall function in the same way as the interface mbsnrtowcs(), except that \_\_mbsnrtowcs\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *destlen* specifies the size of the object *dest*. If *len* exceeds *destlen*, the function shall abort, and the program calling it shall exit.

The \_\_mbsnrtowcs\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_mbsrtowcs\_chk

##### Name

\_\_mbsrtowcs\_chk — convert a multibyte string to a wide-character string, with buffer overflow checking

##### Synopsis

#include <wchar.h>

size\_t \_\_mbsrtowcs\_chk(wchar\_t \* *dest*, const char \* \* *src*, size\_t *len*, mbstate\_t \* *ps*, size\_t *destlen*);

##### Description

The interface \_\_mbsrtowcs\_chk() shall function in the same way as the interface mbsrtowcs(), except that \_\_mbsrtowcs\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *destlen* specifies the size of the object *dest*. If *len* exceeds *destlen*, the function shall abort, and the program calling it shall exit.

The \_\_mbsrtowcs\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_mbstowcs\_chk

##### Name

\_\_mbstowcs\_chk — convert a multibyte string to a wide-character string, with buffer overflow checking

##### Synopsis

#include <stdlib.h>

size\_t \_\_mbstowcs\_chk(wchar\_t \* *dest*, const char \* *src*, size\_t *len*, size\_t *destlen*);

##### Description

The interface \_\_mbstowcs\_chk() shall function in the same way as the interface mbstowcs(), except that \_\_mbstowcs\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *destlen* specifies the size of the object *dest*. If *len* exceeds *destlen*, the function shall abort, and the program calling it shall exit.

The \_\_mbstowcs\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_memcpy\_chk

##### Name

\_\_memcpy\_chk — copy memory area, with buffer overflow checking

##### Synopsis

#include <string.h>

void \* \_\_memcpy\_chk(void \* *dest*, const void \* *src*, size\_t *len*, size\_t *destlen*);

##### Description

The interface \_\_memcpy\_chk() shall function in the same way as the interface memcpy(), except that \_\_memcpy\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *destlen* specifies the size of the object *dest*. If *len* exceeds *destlen*, the function shall abort, and the program calling it shall exit.

The \_\_memcpy\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_memmove\_chk

##### Name

\_\_memmove\_chk — copy memory area, with buffer overflow checking

##### Synopsis

#include <string.h>

void \* \_\_memmove\_chk(void \* *dest*, const void \* *src*, size\_t *len*, size\_t *destlen*);

##### Description

The interface \_\_memmove\_chk() shall function in the same way as the interface memmove(), except that \_\_memmove\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *destlen* specifies the size of the object *dest*. If *len* exceeds *destlen*, the function shall abort, and the program calling it shall exit.

The \_\_memmove\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_mempcpy

##### Name

\_\_mempcpy — copy given number of bytes of source to destination

##### Synopsis

#include <string.h>

void \* \_\_mempcpy(void \* restrict *dest*, const void \* restrict *src*, size\_t *n*);

##### Description

\_\_mempcpy() copies *n* bytes of *src* to *dest*, returning a pointer to the byte after the last written byte.

If copying takes place between objects that overlap, the behavior is undefined.

If either *dest* or *src* is a null pointer, the behavior is undefined.

If *n* is 0 and the other parameters are valid, the return value is *dest*.

\_\_mempcpy() is not in the source standard; it is only in the binary standard.

#### \_\_mempcpy\_chk

##### Name

\_\_mempcpy\_chk — copy memory area, with buffer overflow checking

##### Synopsis

#include <string.h>

void \* \_\_mempcpy\_chk(void \* *dest*, const void \* *src*, size\_t *len*, size\_t *destlen*);

##### Description

The interface \_\_mempcpy\_chk() shall function in the same way as the interface mempcpy(), except that \_\_mempcpy\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *destlen* specifies the size of the object *dest*. If *len* exceeds *destlen*, the function shall abort, and the program calling it shall exit.

The \_\_mempcpy\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_memset\_chk

##### Name

\_\_memset\_chk — fill memory with a constant byte, using buffer overflow checking

##### Synopsis

#include <string.h>

void \* \_\_memset\_chk(void \* *dest*, int *c*, size\_t *len*, size\_t *destlen*);

##### Description

The interface \_\_memset\_chk() shall function in the same way as the interface memset(), except that \_\_memset\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *destlen* specifies the size of the object *dest*. If *len* exceeds *destlen*, the function shall abort, and the program calling it shall exit.

The \_\_memset\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_pread64\_chk

##### Name

\_\_pread64\_chk — read from a file descriptor at a given offset, with buffer overflow checking

##### Synopsis

#include <unistd.h>

ssize\_t \_\_pread64\_chk(int *fd*, void \* *buf*, size\_t *nbytes*, off64\_t *offset*, size\_t *buflen*);

##### Description

The interface \_\_pread64\_chk() shall function in the same way as the interface pread64(), except that \_\_pread64\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *buflen* specifies the size of the buffer *buf*. If *nbytes* exceeds *buflen*, the function shall abort, and the program calling it shall exit.

The \_\_pread64\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_pread\_chk

##### Name

\_\_pread\_chk — read from a file descriptor at a given offset, with buffer overflow checking

##### Synopsis

#include <unistd.h>

ssize\_t \_\_pread\_chk(int *fd*, void \* *buf*, size\_t *nbytes*, off\_t *offset*, size\_t *buflen*);

##### Description

The interface \_\_pread\_chk() shall function in the same way as the interface pread(), except that \_\_pread\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *buflen* specifies the size of the buffer *buf*. If *nbytes* exceeds *buflen*, the function shall abort, and the program calling it shall exit.

The \_\_pread\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_printf\_chk

##### Name

\_\_printf\_chk — format and print data, with stack checking

##### Synopsis

#include <stdio.h>

int \_\_printf\_chk(int *flag*, const char \* *format*);

##### Description

The interface \_\_printf\_chk() shall function in the same way as the interface printf(), except that \_\_printf\_chk() shall check for stack overflow before computing a result, depending on the value of the *flag* parameter. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

In general, the higher the value of *flag*, the more security measures this interface shall take in the form of checking the stack, parameter values, and so on.

The \_\_printf\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_rawmemchr

##### Name

\_\_rawmemchr — scan memory

##### Synopsis

#include <string.h>

void \* \_\_rawmemchr(const void \* *s*, int *c*);

##### Description

The \_\_rawmemchr() function shall locate the first occurrence of *c* (converted to an unsigned char) in the object pointed to by *s*. If the byte does not occur in the object, then the behavior is undefined.

\_\_rawmemchr() is a weak alias for rawmemchr(). It is similar to memchr(), but it has no length limit.

\_\_rawmemchr() is not in the source standard; it is only in the binary standard.

##### Return Value

The \_\_rawmemchr() function shall return a pointer to the located byte.

#### \_\_read\_chk

##### Name

\_\_read\_chk — read from a file descriptor, with buffer overflow checking

##### Synopsis

#include <unistd.h>

ssize\_t \_\_read\_chk(int *fd*, void \* *buf*, size\_t *nbytes*, size\_t *buflen*);

##### Description

The interface \_\_read\_chk() shall function in the same way as the interface read(), except that \_\_read\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *buflen* specifies the size of the buffer *buf*. If *nbytes* exceeds *buflen*, the function shall abort, and the program calling it shall exit.

The \_\_read\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_readlink\_chk

##### Name

\_\_readlink\_chk — display value of a symbolic link, with buffer overflow checking

##### Synopsis

#include <unistd.h>

ssize\_t \_\_readlink\_chk(const char \* *path*, char \* *buf*, size\_t *len*, size\_t *buflen*);

##### Description

The interface \_\_readlink\_chk() shall function in the same way as the interface readlink(), except that \_\_readlink\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *buflen* specifies the size of the buffer *buf*. If *len* exceeds *buflen*, the function shall abort, and the program calling it shall exit.

The \_\_readlink\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_realpath\_chk

##### Name

\_\_realpath\_chk — return the canonicalized absolute pathname, with buffer overflow checking

##### Synopsis

#include <stdlib.h>

char \* \_\_realpath\_chk(const char \* *path*, char \* *resolved\_path*, size\_t *resolved\_len*);

##### Description

The interface \_\_realpath\_chk() shall function in the same way as the interface realpath(), except that \_\_realpath\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *resolved\_len* specifies the size of the string *resolved\_path*. If *resolved\_len* is less than PATH\_MAX, then the function shall abort, and the program calling it shall exit.

The \_\_realpath\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_recv\_chk

##### Name

\_\_recv\_chk — receive a message from a socket, with buffer overflow checking

##### Synopsis

#include <sys/socket.h>

ssize\_t \_\_recv\_chk(int *fd*, void \* *buf*, size\_t *len*, size\_t *buflen*, int *flag*);

##### Description

The interface \_\_recv\_chk() shall function in the same way as the interface recv(), except that \_\_recv\_chk() shall check for buffer overflow before computing a result, depending on the value of the *flag* parameter. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

In general, the higher the value of *flag*, the more security measures this interface shall take in the form of checking the buffer, parameter values, and so on.

The parameter *buflen* specifies the size of the buffer *buf*. If *len* exceeds *buflen*, the function shall abort, and the program calling it shall exit.

The \_\_recv\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_recvfrom\_chk

##### Name

\_\_recvfrom\_chk — receive a message from a socket, with buffer overflow checking

##### Synopsis

#include <sys/socket.h>

ssize\_t \_\_recvfrom\_chk(int *fd*, void \* *buf*, size\_t *len*, size\_t *buflen*, int *flag*, struct sockaddr \* *from*, socklen\_t \* *fromlen*);

##### Description

The interface \_\_recvfrom\_chk() shall function in the same way as the interface recvfrom(), except that \_\_recvfrom\_chk() shall check for buffer overflow before computing a result, depending on the value of the *flag* parameter. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

In general, the higher the value of *flag*, the more security measures this interface shall take in the form of checking the buffer, parameter values, and so on.

The parameter *buflen* specifies the size of the buffer *buf*. If *len* exceeds *buflen*, the function shall abort, and the program calling it shall exit.

The \_\_recvfrom\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_register\_atfork

##### Name

\_\_register\_atfork — alias for register\_atfork

##### Synopsis

int \_\_register\_atfork(void (\**prepare*) (void), void (\**parent*) (void), void (\**child*) (void), void \**\_\_dso\_handle*);

##### Description

\_\_register\_atfork() implements pthread\_atfork() as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4). The additional parameter *\_\_dso\_handle* allows a shared object to pass in it's handle so that functions registered by \_\_register\_atfork() can be unregistered by the runtime when the shared object is unloaded.

#### \_\_sigsetjmp

##### Name

\_\_sigsetjmp — save stack context for non-local goto

##### Synopsis

int \_\_sigsetjmp(jmp\_buf *env*, int *savemask*);

##### Description

\_\_sigsetjmp() has the same behavior as sigsetjmp() as specified by [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

\_\_sigsetjmp() is not in the source standard; it is only in the binary standard.

#### \_\_snprintf\_chk

##### Name

\_\_snprintf\_chk — convert formatted output, with buffer overflow checking

##### Synopsis

#include <stdio.h>

int \_\_snprintf\_chk(char \* *str*, size\_t *maxlen*, int *flag*, size\_t *strlen*, const char \* *format*);

##### Description

The interface \_\_snprintf\_chk() shall function in the same way as the interface snprintf(), except that \_\_snprintf\_chk() shall check for buffer overflow before computing a result, depending on the value of the *flag* parameter. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

In general, the higher the value of *flag*, the more security measures this interface shall take in the form of checking the buffer, parameter values, and so on.

The parameter *strlen* specifies the size of the buffer *str*. If *strlen* is less than *maxlen*, the function shall abort, and the program calling it shall exit.

The \_\_snprintf\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_sprintf\_chk

##### Name

\_\_sprintf\_chk — convert formatted output, with stack checking

##### Synopsis

#include <stdio.h>

int \_\_sprintf\_chk(char \* *str*, int *flag*, size\_t *strlen*, const char \* *format*);

##### Description

The interface \_\_sprintf\_chk() shall function in the same way as the interface sprintf(), except that \_\_sprintf\_chk() shall check for stack overflow before computing a result, depending on the value of the *flag* parameter. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

In general, the higher the value of *flag*, the more security measures this interface shall take in the form of checking the stack, parameter values, and so on.

The parameter *strlen* specifies the size of the string *str*. If *strlen* is zero, the function shall abort, and the program calling it shall exit.

The \_\_sprintf\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_stack\_chk\_fail

##### Name

\_\_stack\_chk\_fail — terminate a function in case of stack overflow

##### Synopsis

void \_\_stack\_chk\_fail(void);

##### Description

The interface \_\_stack\_chk\_fail() shall abort the function that called it with a message that a stack overflow has been detected. The program that called the function shall then exit.

The \_\_stack\_chk\_fail() function is not in the source standard; it is only in the binary standard.

##### Application Usage (informative)

The interface \_\_stack\_chk\_fail() does not check for a stack overflow itself. It merely reports one when invoked.

#### \_\_stpcpy

##### Name

\_\_stpcpy — alias for stpcpy

##### Synopsis

#include <string.h>

char \* \_\_stpcpy(char \* *dest*, const char \* *src*);

##### Description

The \_\_stpcpy() function has the same specification as the stpcpy().

\_\_stpcpy() is not in the source standard; it is only in the binary standard.

#### \_\_stpcpy\_chk

##### Name

\_\_stpcpy\_chk — copy a string returning a pointer to its end, with buffer overflow checking

##### Synopsis

#include <string.h>

char \* \_\_stpcpy\_chk(char \* *dest*, const char \* *src*, size\_t *destlen*);

##### Description

The interface \_\_stpcpy\_chk() shall function in the same way as the interface stpcpy(), except that \_\_stpcpy\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *destlen* specifies the size of the object pointed to by *dest*.

The \_\_stpcpy\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_stpncpy\_chk

##### Name

\_\_stpncpy\_chk — copy a fixed-size string, returning a pointer to its end, with buffer overflow checking

##### Synopsis

#include <string.h>

char \* \_\_stpncpy\_chk(char \* *dest*, const char \* *src*, size\_t *n*, size\_t *destlen*);

##### Description

The interface \_\_stpncpy\_chk() shall function in the same way as the interface stpncpy(), except that \_\_stpncpy\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *destlen* specifies the size of the object pointed to by *dest*. If *n* exceeds *destlen*, the function shall abort, and the program calling it shall exit.

The \_\_stpncpy\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_strcat\_chk

##### Name

\_\_strcat\_chk — concatenate two strings, with buffer overflow checking

##### Synopsis

#include <string.h>

char \* \_\_strcat\_chk(char \* *dest*, const char \* *src*, size\_t *destlen*);

##### Description

The interface \_\_strcat\_chk() shall function in the same way as the interface strcat(), except that \_\_strcat\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *destlen* specifies the size of the object pointed to by *dest*.

The \_\_strcat\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_strcpy\_chk

##### Name

\_\_strcpy\_chk — copy a string, with buffer overflow checking

##### Synopsis

#include <string.h>

char \* \_\_strcpy\_chk(char \* *dest*, const char \* *src*, size\_t *destlen*);

##### Description

The interface \_\_strcpy\_chk() shall function in the same way as the interface strcpy(), except that \_\_strcpy\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *destlen* specifies the size of the object pointed to by *dest*.

The \_\_strcpy\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_strdup

##### Name

\_\_strdup — alias for strdup

##### Synopsis

char \* \_\_strdup(const char \* *string*);

##### Description

\_\_strdup() has the same specification as strdup().

\_\_strdup() is not in the source standard; it is only in the binary standard.

#### \_\_strncat\_chk

##### Name

\_\_strncat\_chk — concatenate two strings, with buffer overflow checking

##### Synopsis

#include <string.h>

char \* \_\_strncat\_chk(char \* *s1*, const char \* *s2*, size\_t *n*, size\_t *s1len*);

##### Description

The interface \_\_strncat\_chk() shall function in the same way as the interface strncat(), except that \_\_strncat\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *s1len* specifies the size of the object pointed to by *s1*.

The \_\_strncat\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_strncpy\_chk

##### Name

\_\_strncpy\_chk — copy a string, with buffer overflow checking

##### Synopsis

#include <string.h>

char \* \_\_strncpy\_chk(char \* *s1*, const char \* *s2*, size\_t *n*, size\_t *s1len*);

##### Description

The interface \_\_strncpy\_chk() shall function in the same way as the interface strncpy(), except that \_\_strncpy\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *s1len* specifies the size of the object pointed to by *s1*.

The \_\_strncpy\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_strtod\_internal

##### Name

\_\_strtod\_internal — underlying function for strtod

##### Synopsis

double \_\_strtod\_internal(const char \* *\_\_nptr*, char \* \* *\_\_endptr*, int *\_\_group*);

##### Description

*\_\_group* shall be 0 or the behavior of \_\_strtod\_internal() is undefined.

\_\_strtod\_internal(*\_\_nptr*, *\_\_endptr*, 0)() has the same specification as strtod(*\_\_nptr*, *\_\_endptr*)().

\_\_strtod\_internal() is not in the source standard; it is only in the binary standard.

#### \_\_strtof\_internal

##### Name

\_\_strtof\_internal — underlying function for strtof

##### Synopsis

float \_\_strtof\_internal(const char \* *\_\_nptr*, char \* \* *\_\_endptr*, int *\_\_group*);

##### Description

*\_\_group* shall be 0 or the behavior of \_\_strtof\_internal() is undefined.

\_\_strtof\_internal(*\_\_nptr*, *\_\_endptr*, 0)() has the same specification as strtof(*\_\_nptr*, *\_\_endptr*)().

\_\_strtof\_internal() is not in the source standard; it is only in the binary standard.

#### \_\_strtok\_r

##### Name

\_\_strtok\_r — alias for strtok\_r

##### Synopsis

char \* \_\_strtok\_r(char \* restrict *s*, const char \* restrict *delim*, char \* \* restrict *save\_ptr*);

##### Description

\_\_strtok\_r() has the same specification as strtok\_r().

\_\_strtok\_r() is not in the source standard; it is only in the binary standard.

#### \_\_strtol\_internal

##### Name

\_\_strtol\_internal — alias for strtol

##### Synopsis

long int \_\_strtol\_internal(const char \* *\_\_nptr*, char \* \* *\_\_endptr*, int *\_\_base*, int *\_\_group*);

##### Description

*\_\_group* shall be 0 or the behavior of \_\_strtol\_internal() is undefined.

\_\_strtol\_internal(*\_\_nptr*, *\_\_endptr*, *\_\_base*, 0) has the same specification as strtol(*\_\_nptr*, *\_\_endptr*, *\_\_base*).

\_\_strtol\_internal() is not in the source standard; it is only in the binary standard.

#### \_\_strtold\_internal

##### Name

\_\_strtold\_internal — underlying function for strtold

##### Synopsis

long double \_\_strtold\_internal(const char \* *\_\_nptr*, char \* \* *\_\_endptr*, int *\_\_group*);

##### Description

*\_\_group* shall be 0 or the behavior of \_\_strtold\_internal() is undefined.

\_\_strtold\_internal(*\_\_nptr*, *\_\_endptr*, 0) has the same specification as strtold(*\_\_nptr*, *\_\_endptr*).

\_\_strtold\_internal() is not in the source standard; it is only in the binary standard.

#### \_\_strtoll\_internal

##### Name

\_\_strtoll\_internal — underlying function for strtoll

##### Synopsis

long long \_\_strtoll\_internal(const char \* *\_\_nptr*, char \* \* *\_\_endptr*, int *\_\_base*, int *\_\_group*);

##### Description

*\_\_group* shall be 0 or the behavior of \_\_strtoll\_internal() is undefined.

\_\_strtoll\_internal(*\_\_nptr*, *\_\_endptr*, *\_\_base*, 0) has the same specification as strtoll(*\_\_nptr*, *\_\_endptr*, *\_\_base*).

\_\_strtoll\_internal() is not in the source standard; it is only in the binary standard.

#### \_\_strtoul\_internal

##### Name

\_\_strtoul\_internal — underlying function for strtoul

##### Synopsis

unsigned long int \_\_strtoul\_internal(const char \* *\_\_nptr*, char \* \* *\_\_endptr*, int *\_\_base*, int *\_\_group*);

##### Description

*\_\_group* shall be 0 or the behavior of \_\_strtoul\_internal() is undefined.

\_\_strtoul\_internal(*\_\_nptr*, *\_\_endptr*, *\_\_base*, 0) has the same specification as strtoul(*\_\_nptr*, *\_\_endptr*, *\_\_base*).

\_\_strtoul\_internal() is not in the source standard; it is only in the binary standard.

#### \_\_strtoull\_internal

##### Name

\_\_strtoull\_internal — underlying function for strtoull

##### Synopsis

unsigned long long \_\_strtoull\_internal(const char \* *\_\_nptr*, char \* \* *\_\_endptr*, int *\_\_base*, int *\_\_group*);

##### Description

*\_\_group* shall be 0 or the behavior of \_\_strtoull\_internal() is undefined.

\_\_strtoull\_internal(*\_\_nptr*, *\_\_endptr*, *\_\_base*, 0) has the same specification as strtoull(*\_\_nptr*, *\_\_endptr*, *\_\_base*).

\_\_strtoull\_internal() is not in the source standard; it is only in the binary standard.

#### \_\_swprintf\_chk

##### Name

\_\_swprintf\_chk — convert formatted wide-character output, with stack checking

##### Synopsis

#include <wchar.h>

int \_\_swprintf\_chk(wchar\_t \* *s*, size\_t *n*, int *flag*, size\_t *slen*, const wchar\_t \* *format*);

##### Description

The interface \_\_swprintf\_chk() shall function in the same way as the interface swprintf(), except that \_\_swprintf\_chk() shall check for stack overflow before computing a result, depending on the value of the *flag* parameter. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

In general, the higher the value of *flag*, the more security measures this interface shall take in the form of checking the stack, parameter values, and so on.

The parameter *slen* specifies the size of the object pointed to by *s*. If *slen* is less than *maxlen*, the function shall abort and the program calling it shall exit.

In general, the higher the value of *flag*, the more security measures this interface shall take in the form of checking the stack, parameter values, and so on.

The \_\_swprintf\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_sysconf

##### Name

\_\_sysconf — get configuration information at runtime

##### Synopsis

#include <unistd.h>

long \_\_sysconf(int *name*);

##### Description

\_\_sysconf() gets configuration information at runtime.

\_\_sysconf() is weak alias to sysconf().

\_\_sysconf() has the same specification as sysconf().

\_\_sysconf() is not in the source standard; it is only in the binary standard.

#### \_\_syslog\_chk

##### Name

\_\_syslog\_chk — send messages to the system logger, with stack checking

##### Synopsis

#include <syslog.h>

void \_\_syslog\_chk(int *priority*, int *flag*, const char \* *format*);

##### Description

The interface \_\_syslog\_chk() shall function in the same way as the interface syslog(), except that \_\_syslog\_chk() shall check for stack overflow before computing a result, depending on the value of the *flag* parameter. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

In general, the higher the value of *flag*, the more security measures this interface shall take in the form of checking the stack, parameter values, and so on.

The \_\_syslog\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_sysv\_signal

##### Name

\_\_sysv\_signal — signal handling

##### Synopsis

\_\_sighandler\_t \_\_sysv\_signal(int *sig*, \_\_sighandler\_t *handler*);

##### Description

\_\_sysv\_signal() has the same behavior as signal() as specified by [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

\_\_sysv\_signal() is not in the source standard; it is only in the binary standard.

#### \_\_timezone

##### Name

\_\_timezone — external variable containing timezone

##### Synopsis

long int \_\_timezone;

##### Description

The external variable \_\_timezone shall implement the timezone variable timezone as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4). \_\_timezone has the same specification as timezone.

#### \_\_ttyname\_r\_chk

##### Name

\_\_ttyname\_r\_chk — return name of a terminal, with buffer overflow checking (reentrant)

##### Synopsis

#include <unistd.h>

int \_\_ttyname\_r\_chk(int *fd*, char \* *buf*, size\_t *buflen*, size\_t *nreal*);

##### Description

The interface \_\_ttyname\_r\_chk() shall function in the same way as the interface ttyname\_r(), except that \_\_ttyname\_r\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *buflen* specifies the size of the object pointed to by *buf*. If *buflen* exceeds *nreal*, the function shall abort and the program calling it shall exit.

The \_\_ttyname\_r\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_tzname

##### Name

\_\_tzname — external variable containing the timezone names

##### Synopsis

char \* \_\_tzname[2];

##### Description

The external variable \_\_tzname shall implement the timezone name variable tzname as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) function tzset(). \_\_tzname has the same specification as tzname.

#### \_\_vfprintf\_chk

##### Name

\_\_vfprintf\_chk — convert formatted output, with stack checking

##### Synopsis

#include <stdio.h>

int \_\_vfprintf\_chk(FILE \* *fp*, int *flag*, const char \* *format*, va\_list *ap*);

##### Description

The interface \_\_vfprintf\_chk() shall function in the same way as the interface vfprintf(), except that \_\_vfprintf\_chk() shall check for stack overflow before computing a result, depending on the value of the *flag* parameter. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

In general, the higher the value of *flag*, the more security measures this interface shall take in the form of checking the stack, parameter values, and so on.

The \_\_vfprintf\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_vfwprintf\_chk

##### Name

\_\_vfwprintf\_chk — convert formatted wide-character output, with stack checking

##### Synopsis

#include <wchar.h>

int \_\_vfwprintf\_chk(FILE \* *fp*, int *flag*, const wchar\_t \* *format*, va\_list *ap*);

##### Description

The interface \_\_vfwprintf\_chk() shall function in the same way as the interface vfwprintf(), except that \_\_vfwprintf\_chk() shall check for stack overflow before computing a result, depending on the value of the *flag* parameter. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

In general, the higher the value of *flag*, the more security measures this interface shall take in the form of checking the stack, parameter values, and so on.

The \_\_vfwprintf\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_vprintf\_chk

##### Name

\_\_vprintf\_chk — convert formatted output, with stack checking

##### Synopsis

#include <stdio.h>

int \_\_vprintf\_chk(int *flag*, const char \* *format*, va\_list *ap*);

##### Description

The interface \_\_vprintf\_chk() shall function in the same way as the interface vprintf(), except that \_\_vprintf\_chk() shall check for stack overflow before computing a result, depending on the value of the *flag* parameter. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

In general, the higher the value of *flag*, the more security measures this interface shall take in the form of checking the stack, parameter values, and so on.

The \_\_vprintf\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_vsnprintf\_chk

##### Name

\_\_vsnprintf\_chk — convert formatted output, with stack checking

##### Synopsis

#include <stdio.h>

int \_\_vsnprintf\_chk(char \* *s*, size\_t *maxlen*, int *flag*, size\_t *slen*, const char \* *format*, va\_list *args*);

##### Description

The interface \_\_vsnprintf\_chk() shall function in the same way as the interface vsnprintf(), except that \_\_vsnprintf\_chk() shall check for stack overflow before computing a result, depending on the value of the *flag* parameter. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

In general, the higher the value of *flag*, the more security measures this interface shall take in the form of checking the stack, parameter values, and so on.

The parameter *slen* specifies the size of the object pointed to by *s*. If *slen* is less than *maxlen*, the function shall abort and the program calling it shall exit.

In general, the higher the value of *flag*, the more security measures this interface shall take in the form of checking the stack, parameter values, and so on.

The \_\_vsnprintf\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_vsprintf\_chk

##### Name

\_\_vsprintf\_chk — convert formatted output, with stack checking

##### Synopsis

#include <stdio.h>

int \_\_vsprintf\_chk(char \* *s*, int *flag*, size\_t *slen*, const char \* *format*, va\_list *args*);

##### Description

The interface \_\_vsprintf\_chk() shall function in the same way as the interface vsprintf(), except that \_\_vsprintf\_chk() shall check for stack overflow before computing a result, depending on the value of the *flag* parameter. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

In general, the higher the value of *flag*, the more security measures this interface shall take in the form of checking the stack, parameter values, and so on.

The parameter *slen* specifies the size of the object pointed to by *s*. If its value is zero, the function shall abort and the program calling it shall exit.

The \_\_vsprintf\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_vswprintf\_chk

##### Name

\_\_vswprintf\_chk — convert formatted wide-character output, with stack checking

##### Synopsis

#include <wchar.h>

int \_\_vswprintf\_chk(wchar\_t \* *s*, size\_t *maxlen*, int *flag*, size\_t *slen*, const wchar\_t \* *format*, va\_list *args*);

##### Description

The interface \_\_vswprintf\_chk() shall function in the same way as the interface vswprintf(), except that \_\_vswprintf\_chk() shall check for stack overflow before computing a result, depending on the value of the *flag* parameter. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

In general, the higher the value of *flag*, the more security measures this interface shall take in the form of checking the stack, parameter values, and so on.

The parameter *slen* specifies the size of the object pointed to by *s*. If *slen* is less than *maxlen*, the function shall abort and the program calling it shall exit.

The \_\_vswprintf\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_vsyslog\_chk

##### Name

\_\_vsyslog\_chk — send messages to the system logger, with stack checking

##### Synopsis

#include <syslog.h>

void \_\_vsyslog\_chk(int *priority*, int *flag*, const char \* *format*, va\_list *ap*);

##### Description

The interface \_\_vsyslog\_chk() shall function in the same way as the interface vsyslog(), except that \_\_vsyslog\_chk() shall check for stack overflow before computing a result, depending on the value of the *flag* parameter. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

In general, the higher the value of *flag*, the more security measures this interface shall take in the form of checking the stack, parameter values, and so on.

The \_\_vsyslog\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_vwprintf\_chk

##### Name

\_\_vwprintf\_chk — convert formatted wide-character output, with stack checking

##### Synopsis

#include <wchar.h>

int \_\_vwprintf\_chk(int *flag*, const wchar\_t \* *format*, va\_list *ap*);

##### Description

The interface \_\_vwprintf\_chk() shall function in the same way as the interface vwprintf(), except that \_\_vwprintf\_chk() shall check for stack overflow before computing a result, depending on the value of the *flag* parameter. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

In general, the higher the value of *flag*, the more security measures this interface shall take in the form of checking the stack, parameter values, and so on.

The \_\_vwprintf\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_wcpcpy\_chk

##### Name

\_\_wcpcpy\_chk — copy a wide-character string, returning a pointer to its end, with buffer overflow checking

##### Synopsis

#include <wchar.h>

wchar\_t \* \_\_wcpcpy\_chk(wchar\_t \* *dest*, const wchar\_t \* *src*, size\_t *destlen*);

##### Description

The interface \_\_wcpcpy\_chk() shall function in the same way as the interface wcpcpy(), except that \_\_wcpcpy\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *destlen* specifies the size of the object pointed to by *dest*.

The \_\_wcpcpy\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_wcpncpy\_chk

##### Name

\_\_wcpncpy\_chk — copy a fixed-size string of wide characters, returning a pointer to its end, with buffer overflow checking

##### Synopsis

#include <wchar.h>

wchar\_t \* \_\_wcpncpy\_chk(wchar\_t \* *dest*, const wchar\_t \* *src*, size\_t *n*, size\_t *destlen*);

##### Description

The interface \_\_wcpncpy\_chk() shall function in the same way as the interface wcpncpy(), except that \_\_wcpncpy\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *destlen* specifies the size of the object pointed to by *dest*. If *n* exceeds *destlen*, the function shall abort and the program calling it shall exit.

The \_\_wcpncpy\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_wcrtomb\_chk

##### Name

\_\_wcrtomb\_chk — convert a wide character to a multibyte sequence, with buffer overflow checking

##### Synopsis

#include <wchar.h>

size\_t \_\_wcrtomb\_chk(char \* *s*, wchar\_t *wchar*, mbstate\_t \* *ps*, size\_t *buflen*);

##### Description

The interface \_\_wcrtomb\_chk() shall function in the same way as the interface wcrtomb(), except that \_\_wcrtomb\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *buflen* specifies the size of the object pointed to by *s*. If it is less than MB\_CUR\_MAX, then the function shall abort and the program calling it shall exit.

The \_\_wcrtomb\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_wcscat\_chk

##### Name

\_\_wcscat\_chk — concatenate two wide-character strings, with buffer overflow checking

##### Synopsis

#include <wchar.h>

wchar\_t \* \_\_wcscat\_chk(wchar\_t \* *dest*, const wchar\_t \* *src*, size\_t *destlen*);

##### Description

The interface \_\_wcscat\_chk() shall function in the same way as the interface wcscat(), except that \_\_wcscat\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *destlen* specifies the size of the object pointed to by *dest*.

The \_\_wcscat\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_wcscpy\_chk

##### Name

\_\_wcscpy\_chk — copy a wide-character string, with buffer overflow checking

##### Synopsis

#include <wchar.h>

wchar\_t \* \_\_wcscpy\_chk(wchar\_t \* *dest*, const wchar\_t \* *src*, size\_t *n*);

##### Description

The interface \_\_wcscpy\_chk() shall function in the same way as the interface wcscpy(), except that \_\_wcscpy\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The \_\_wcscpy\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_wcsncat\_chk

##### Name

\_\_wcsncat\_chk — concatenate two wide-character strings, with buffer overflow checking

##### Synopsis

#include <wchar.h>

wchar\_t \* \_\_wcsncat\_chk(wchar\_t \* *dest*, const wchar\_t \* *src*, size\_t *n*, size\_t *destlen*);

##### Description

The interface \_\_wcsncat\_chk() shall function in the same way as the interface wcsncat(), except that \_\_wcsncat\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *destlen* specifies the size of the object pointed to by *dest*.

The \_\_wcsncat\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_wcsncpy\_chk

##### Name

\_\_wcsncpy\_chk — copy a fixed-size string of wide characters, with buffer overflow checking

##### Synopsis

#include <wchar.h>

wchar\_t \* \_\_wcsncpy\_chk(wchar\_t \* *dest*, const wchar\_t \* *src*, size\_t *n*, size\_t *destlen*);

##### Description

The interface \_\_wcsncpy\_chk() shall function in the same way as the interface wcsncpy(), except that \_\_wcsncpy\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *destlen* specifies the size of the object pointed to by *dest*. If *len* exceeds *destlen*, the function shall abort and the program calling it shall exit.

The \_\_wcsncpy\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_wcsnrtombs\_chk

##### Name

\_\_wcsnrtombs\_chk — convert a wide-character string to a multibyte string, with buffer overflow checking

##### Synopsis

#include <wchar.h>

size\_t \_\_wcsnrtombs\_chk(char \* *dest*, const wchar\_t \* \* *src*, size\_t *nwc*, size\_t *len*, mbstate\_t \* *ps*, size\_t *destlen*);

##### Description

The interface \_\_wcsnrtombs\_chk() shall function in the same way as the interface wcsnrtombs(), except that \_\_wcsnrtombs\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *destlen* specifies the size of the object pointed to by *dest*. If *len* exceeds *destlen*, the function shall abort and the program calling it shall exit.

The \_\_wcsnrtombs\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_wcsrtombs\_chk

##### Name

\_\_wcsrtombs\_chk — convert a wide-character string to a multibyte string, with buffer overflow checking

##### Synopsis

#include <wchar.h>

size\_t \_\_wcsrtombs\_chk(char \* *dest*, const wchar\_t \* \* *src*, size\_t *len*, mbstate\_t \* *ps*, size\_t *destlen*);

##### Description

The interface \_\_wcsrtombs\_chk() shall function in the same way as the interface wcsrtombs(), except that \_\_wcsrtombs\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *destlen* specifies the size of the object pointed to by *dest*. If *len* exceeds *destlen*, the function shall abort and the program calling it shall exit.

The \_\_wcsrtombs\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_wcstod\_internal

##### Name

\_\_wcstod\_internal — underlying function for wcstod

##### Synopsis

double \_\_wcstod\_internal(const wchar\_t \* *nptr*, wchar\_t \* \* *endptr*, int *group*);

##### Description

*group* shall be 0 or the behavior of \_\_wcstod\_internal() is undefined.

\_\_wcstod\_internal(*nptr*, *endptr*, 0) shall behave as wcstod(*nptr*, *endptr*) as specified by [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

\_\_wcstod\_internal() is not in the source standard; it is only in the binary standard.

#### \_\_wcstof\_internal

##### Name

\_\_wcstof\_internal — underlying function for wcstof

##### Synopsis

float \_\_wcstof\_internal(const wchar\_t \* *nptr*, wchar\_t \* \* *endptr*, int *group*);

##### Description

*group* shall be 0 or the behavior of \_\_wcstof\_internal() is undefined.

\_\_wcstof\_internal(*nptr*, *endptr*, 0) shall behave as wcstof(*nptr*, *endptr*) as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

\_\_wcstof\_internal() is not in the source standard; it is only in the binary standard.

#### \_\_wcstol\_internal

##### Name

\_\_wcstol\_internal — underlying function for wcstol

##### Synopsis

long \_\_wcstol\_internal(const wchar\_t \* *nptr*, wchar\_t \* \* *endptr*, int *base*, int *group*);

##### Description

*group* shall be 0 or the behavior of \_\_wcstol\_internal() is undefined.

\_\_wcstol\_internal(*nptr*, *endptr*, *base*, 0) shall behave as wcstol(*nptr*, *endptr*, *base*) as specified by [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

\_\_wcstol\_internal() is not in the source standard; it is only in the binary standard.

#### \_\_wcstold\_internal

##### Name

\_\_wcstold\_internal — underlying function for wcstold

##### Synopsis

long double \_\_wcstold\_internal(const wchar\_t \* *nptr*, wchar\_t \* \* *endptr*, int *group*);

##### Description

*group* shall be 0 or the behavior of \_\_wcstold\_internal() is undefined.

\_\_wcstold\_internal(*nptr*, *endptr*, 0) shall behave as wcstold(*nptr*, *endptr*) as specified by [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

\_\_wcstold\_internal() is not in the source standard; it is only in the binary standard.

#### \_\_wcstombs\_chk

##### Name

\_\_wcstombs\_chk — convert a wide-character string to a multibyte string, with buffer overflow checking

##### Synopsis

#include <stdlib.h>

size\_t \_\_wcstombs\_chk(char \* *dest*, const wchar\_t \* *src*, size\_t *len*, size\_t *destlen*);

##### Description

The interface \_\_wcstombs\_chk() shall function in the same way as the interface wcstombs(), except that \_\_wcstombs\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *destlen* specifies the size of the object pointed to by *dest*. If *len* exceeds *destlen*, the function shall abort and the program calling it shall exit.

The \_\_wcstombs\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_wcstoul\_internal

##### Name

\_\_wcstoul\_internal — underlying function for wcstoul

##### Synopsis

unsigned long \_\_wcstoul\_internal(const wchar\_t \* restrict *nptr*, wchar\_t \* \* restrict *endptr*, int *base*, int *group*);

##### Description

*group* shall be 0 or the behavior of \_\_wcstoul\_internal() is undefined.

\_\_wcstoul\_internal(*nptr*, *endptr*, *base*, 0)() shall behave as wcstoul(*nptr*, *endptr*, *base*)() as specified by [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

\_\_wcstoul\_internal() is not in the source standard; it is only in the binary standard.

#### \_\_wctomb\_chk

##### Name

\_\_wctomb\_chk — convert a wide character to a multibyte sequence, with buffer overflow checking

##### Synopsis

#include <stdlib.h>

int \_\_wctomb\_chk(char \* *s*, wchar\_t *wchar*, size\_t *buflen*);

##### Description

The interface \_\_wctomb\_chk() shall function in the same way as the interface wctomb(), except that \_\_wctomb\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *buflen* specifies the size of the object pointed to by *s*. If it is less than MB\_CUR\_MAX, then the function shall abort and the program calling it shall exit.

The \_\_wctomb\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_wmemcpy\_chk

##### Name

\_\_wmemcpy\_chk — copy an array of wide-characters, with buffer overflow checking

##### Synopsis

#include <wchar.h>

wchar\_t \* \_\_wmemcpy\_chk(wchar\_t \* *s1*, const wchar\_t \* *s2*, size\_t *n*, size\_t *ns1*);

##### Description

The interface \_\_wmemcpy\_chk() shall function in the same way as the interface wmemcpy(), except that \_\_wmemcpy\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *ns1* specifies the size of the object pointed to by *s1*. If *n* exceeds *ns1*, the function shall abort and the program calling it shall exit.

The \_\_wmemcpy\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_wmemmove\_chk

##### Name

\_\_wmemmove\_chk — copy an array of wide-characters, with buffer overflow checking

##### Synopsis

#include <wchar.h>

wchar\_t \* \_\_wmemmove\_chk(wchar\_t \* *s1*, const wchar\_t \* *s2*, size\_t *n*, size\_t *ns1*);

##### Description

The interface \_\_wmemmove\_chk() shall function in the same way as the interface wmemmove(), except that \_\_wmemmove\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *ns1* specifies the size of the object pointed to by *s1*. If *n* exceeds *ns1*, the function shall abort and the program calling it shall exit.

The \_\_wmemmove\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_wm**e**mpcpy\_chk

##### Name

\_\_wmempcpy\_chk — copy memory area, with buffer overflow checking

##### Synopsis

#include <wchar.h>

wchar\_t \* \_\_wmempcpy\_chk(wchar\_t \* *s1*, const wchar\_t \* *s2*, size\_t *n*, size\_t *ns1*);

##### Description

The interface \_\_wmempcpy\_chk() shall function in the same way as the interface wmempcpy(), except that \_\_wmempcpy\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *ns1* specifies the size of the object pointed to by *s1*. If *n* exceeds *ns1*, the function shall abort and the program calling it shall exit.

The \_\_wmempcpy\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_wmemset\_chk

##### Name

\_\_wmemset\_chk — fill an array of wide-characters with a constant wide character, with buffer overflow checking

##### Synopsis

#include <wchar.h>

wchar\_t \* \_\_wmemset\_chk(wchar\_t \* *s*, wchar\_t *c*, size\_t *n*, size\_t *destlen*);

##### Description

The interface \_\_wmemset\_chk() shall function in the same way as the interface wmemset(), except that \_\_wmemset\_chk() shall check for buffer overflow before computing a result. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

The parameter *destlen* specifies the size of the object pointed to by *s*. If *n* exceeds *destlen*, the function shall abort and the program calling it shall exit.

The \_\_wmemset\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_wprintf\_chk

##### Name

\_\_wprintf\_chk — convert formatted wide-character output, with stack checking

##### Synopsis

#include <wchar.h>

int \_\_wprintf\_chk(int *flag*, const wchar\_t \* *format*);

##### Description

The interface \_\_wprintf\_chk() shall function in the same way as the interface wprintf(), except that \_\_wprintf\_chk() shall check for stack overflow before computing a result, depending on the value of the *flag* parameter. If an overflow is anticipated, the function shall abort and the program calling it shall exit.

In general, the higher the value of *flag*, the more security measures this interface shall take in the form of checking the stack, parameter values, and so on.

The \_\_wprintf\_chk() function is not in the source standard; it is only in the binary standard.

#### \_\_xmknod

##### Name

\_\_xmknod — make a special file

##### Synopsis

#include <sys/stat.h>

int \_\_xmknod(int *ver*, const char \* *path*, mode\_t *mode*, dev\_t \* *dev*);

##### Description

The \_\_xmknod() function shall implement the mknod() interface. The behavior of \_\_xmknod() for values of *ver* other than \_MKNOD\_VER is undefined. See Data Definitions in the architecture specific part of this specification for the correct value of \_MKNOD\_VER.

\_\_xmknod(\_MKNOD\_VER, *path*, *mode*, *dev*) shall behave as mknod(*path*, *mode*, *dev*) as specified by [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

The \_\_xmknod() function is not in the source standard; it is only in the binary standard.

**Note:** The mknod() function is not in the binary standard; it is only in the source standard.

#### \_\_xmknodat

##### Name

\_\_xmknodat — make a special file relative to a directory file descriptor

##### Synopsis

#include <sys/stat.h>

int \_\_xmknodat(int *ver*, int *dirfd*, const char \* *path*, mode\_t *path*, dev\_t \* *dev*);

##### Description

The \_\_xmknodat() function shall implement the mknodat() function. The behavior of \_\_xmknodat() for values of *ver* other than \_MKNOD\_VER is undefined. See Data Definitions in the architecture specific part of this specification for the correct value of \_MKNOD\_VER.

\_\_xmknodat(\_MKNOD\_VER, *dirfd*, *path*, *mode*, *dev*) shall behave as mknodat(*dirfd*, *path*, *mode*, *dev*) as specified by [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

The \_\_xmknodat() function is not in the source standard; it is only in the binary standard.

**Note:** The mknodat() function is not in the binary standard; it is only in the source standard.

#### \_\_xpg\_basename

##### Name

\_\_xpg\_basename — return the last component of a file name

##### Synopsis

#include <libgen.h>

char \* \_\_xpg\_basename(const char \* *path*);

##### Description

The \_\_xpg\_basename() function shall return a pointer to the final component of the pathname named by *path*, as described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) basename().

This function is not in the source standard, it is only in the binary standard.

##### Return Value

See [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

#### \_\_xpg\_sigpause

##### Name

\_\_xpg\_sigpause — remove a signal from the signal mask and suspend the thread

##### Synopsis

#include <signal.h>

int \_\_xpg\_sigpause(int *sig*);

##### Description

The \_\_xpg\_sigpause() function shall implement the sigpause() described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

This function is not in the source standard, it is only in the binary standard.

##### Return Value

See [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

#### \_\_xpg\_strerror\_r

##### Name

\_\_xpg\_strerror\_r — return string describing error number

##### Synopsis

#include <string.h>

int \_\_xpg\_strerror\_r(int *errnum*, char \* *buf*, size\_t *buflen*);

##### Description

The \_\_xpg\_strerror\_r() function shall map the error number in *errnum* to a locale-dependent error message string and shall return the string in the buffer pointed to by *strerrbuf*, with length *buflen*, as described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) strerror\_r().

This function is not in the source standard, it is only in the binary standard.

##### Return Value

See [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

#### \_\_xstat

##### Name

\_\_xstat — get File Status

##### Synopsis

#include <sys/stat.h>

#include <unistd.h>

int \_\_xstat(int *ver*, const char \* *path*, struct stat \* *stat\_buf*);

int \_\_lxstat(int *ver*, const char \* *path*, struct stat \* *stat\_buf*);

int \_\_fxstat(int *ver*, int *fildes*, struct stat \* *stat\_buf*);

##### Description

The functions \_\_xstat(), \_\_lxstat(), and \_\_fxstat() shall implement the functions stat(), lstat(), and fstat() respectively.

The behavior of these functions for values of *ver* other than \_STAT\_VER is undefined. See Data Definitions in the architecture specific part of this specification for the correct value of \_STAT\_VER.

\_\_xstat(\_STAT\_VER, *path*, *stat\_buf*) shall implement stat(*path*, *stat\_buf*) as specified by [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

\_\_lxstat(\_STAT\_VER, *path*, *stat\_buf*) shall implement lstat(*path*, *stat\_buf*) as specified by [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

\_\_fxstat(\_STAT\_VER, *fildes*, *stat\_buf*) shall implement fstat(*fildes*, *stat\_buf*) as specified by [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

\_\_xstat(), \_\_lxstat(), and \_\_fxstat() are not in the source standard; they are only in the binary standard.

stat(), lstat(), and fstat() are not in the binary standard; they are only in the source standard.

#### \_\_xstat64

##### Name

\_\_xstat64 — get File Status

##### Synopsis

#define \_LARGEFILE\_SOURCE 1

#include <sys/stat.h>

#include <unistd.h>

int \_\_xstat64(int *ver*, const char \* *path*, struct stat64 \* *stat\_buf*);

int \_\_lxstat64(int *ver*, const char \* *path*, struct stat64 \* *stat\_buf*);

int \_\_fxstat64(int *ver*, int *fildes*, struct stat64 \* *stat\_buf*);

##### Description

The functions \_\_xstat64(), \_\_lxstat64(), and \_\_fxstat64() shall implement the functions stat64(), lstat64(), and fstat64() respectively.

The behavior of these functions for values of *ver* other than \_STAT\_VER is undefined. See Data Definitions in the architecture specific part of this specification for the correct value of \_STAT\_VER.

\_\_xstat64(\_STAT\_VER, *path*, *stat\_buf*) shall behave as stat64(*path*, *stat\_buf*) as specified by [Large File Support](#ID_STD_46_LFS).

\_\_lxstat64(\_STAT\_VER, *path*, *stat\_buf*) shall behave as lstat64(*path*, *stat\_buf*) as specified by [Large File Support](#ID_STD_46_LFS).

\_\_fxstat64(\_STAT\_VER, *fildes*, *stat\_buf*) shall behave as fstat64(*fildes*, *stat\_buf*) as specified by [Large File Support](#ID_STD_46_LFS).

\_\_xstat64(), \_\_lxstat64(), and \_\_fxstat64() are not in the source standard; they are only in the binary standard.

stat64(), lstat64(), and fstat64() are not in the binary standard; they are only in the source standard.

#### \_environ

##### Name

\_environ — alias for environ - user environment

##### Synopsis

extern char \* \*\_environ;

##### Description

\_environ is an alias for environ - user environment.

#### \_nl\_msg\_cat\_cntr

##### Name

\_nl\_msg\_cat\_cntr — new catalog load counter

##### Synopsis

#include <libintl.h>

extern int \_nl\_msg\_cat\_cntr;

##### Description

The global variable \_nl\_msg\_cat\_cntr is incremented each time a new catalog is loaded. This variable is only in the binary standard; it is not in the source standard.

#### \_sys\_errlist

##### Name

\_sys\_errlist — array containing the "C" locale strings used by strerror()

##### Synopsis

#include <stdio.h>

extern const char \*const \_sys\_errlist[];

##### Description

\_sys\_errlist is an array containing the "C" locale strings used by strerror(). This normally should not be used directly. strerror() provides all of the needed functionality.

#### \_sys\_siglist

##### Name

\_sys\_siglist — array containing the names of the signal names

##### Synopsis

#include <signal.h>

extern const char \*const \_sys\_siglist[NSIG];

##### Description

\_sys\_siglist is an array containing signal description strings ordered by signal number.

The \_sys\_siglist array is only in the binary standard; it is not in the source standard. Applications wishing to access signal descriptions should use the strsignal() function.

#### acct

##### Name

acct — switch process accounting on or off

##### Synopsis

#include <dirent.h>

int acct(const char \* *filename*);

##### Description

When *filename* is the name of an existing file, acct() turns accounting on and appends a record to *filename* for each terminating process. When *filename* is NULL, acct() turns accounting off.

##### Return Value

On success, 0 is returned. On error, -1 is returned and the global variable errno is set appropriately.

##### Errors

ENOSYS

  BSD process accounting has not been enabled when the operating system kernel was compiled. The kernel configuration parameter controlling this feature is *CONFIG\_BSD\_PROCESS\_ACCT*.

ENOMEM

  Out of memory.

EPERM

  The calling process has no permission to enable process accounting.

EACCES

*filename* is not a regular file.

EIO

  Error writing to the *filename*.

EUSERS

  There are no more free file structures or we run out of memory.

#### adjtime

##### Name

adjtime — correct the time to allow synchronization of the system clock

##### Synopsis

#include <time.h>

int adjtime(const struct timeval \* *delta*, struct timeval \* *olddelta*);

##### Description

adjtime() makes small adjustments to the system time as returned by gettimeofday()(2), advancing or retarding it by the time specified by the timeval *delta*. If *delta* is negative, the clock is slowed down by incrementing it more slowly than normal until the correction is complete. If *delta* is positive, a larger increment than normal is used. The skew used to perform the correction is generally a fraction of one percent. Thus, the time is always a monotonically increasing function. A time correction from an earlier call to adjtime() may not be finished when adjtime() is called again. If *olddelta* is non-NULL, the structure pointed to will contain, upon return, the number of microseconds still to be corrected from the earlier call.

adjtime() may be used by time servers that synchronize the clocks of computers in a local area network. Such time servers would slow down the clocks of some machines and speed up the clocks of others to bring them to the average network time.

Appropriate privilege is required to adjust the system time.

##### Return Value

On success, 0 is returned. On error, -1 is returned and the global variable errno is set appropriately.

##### Errors

EFAULT

  An argument points outside the process's allocated address space.

EPERM

  The process does not have appropriate privilege.

#### alphasort64

##### Name

alphasort64 — Comparison function for directory scanning (Large File Support)

##### Synopsis

#include <dirent.h>

int alphasort64(const struct dirent64 \*\* *d1*, const struct dirent64 \*\* *d2*);

##### Description

alpahsort64() is a large-file version of the alphasort() function as defined in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4). If differs only in that the *d1* and *d2* parameters are of type dirent64 instead of type dirent.

#### argz\_add, argz\_add\_sep, argz\_append, argz\_count, argz\_create, argz\_create\_sep, argz\_delete, argz\_extract, argz\_insert, argz\_next, argz\_replace, argz\_stringify

##### Name

argz\_add, argz\_add\_sep, argz\_append, argz\_count, argz\_create, argz\_create\_sep, argz\_delete, argz\_extract, argz\_insert, argz\_next, argz\_replace, argz\_stringify — Operate on argz vectors

##### Synopsis

#include <argz.h>

error\_t argz\_add(char \*\* *argz*, size\_t \* *argz\_len*, const char \* *str*);

error\_t argz\_add\_sep(char \*\* *argz*, size\_t \* *argz\_len*, const char \* *str*, int *sep*);

error\_t argz\_append(char \*\* *argz*, size\_t \* *argz\_len*, const char \* *buf*, size\_t *buf\_len*);

size\_t argz\_count(const char \* *argz*, size\_t \* *argz\_len*);

error\_t argz\_create(char \* const *argv*, char \*\* *argz*, size\_t \* *argz\_len*);

error\_t argz\_create\_sep(const char \* *str*, int *sep*, char \*\* *argz*, size\_t \* *argz\_len*);

void argz\_delete(char \*\* *argz*, size\_t \* *argz\_len*, char \* *entry*);

void argz\_extract(const char \* *argz*, size\_t *argz\_len*, char \*\* *argv*);

error\_t argz\_insert(char \*\* *argz\_insert*, size\_t \* *argz\_len*, char \* *before*, const char \* *entry*);

char argz\_next(const char \* *argz*, size\_t *argz\_len*, const char \* *entry*);

error\_t argz\_replace(char \*\* *argz*, size\_t \* *argz\_len*, const char \* *str*, const char \* *with*, unsigned int \* *replace\_count*);

void argz\_stringify(char \* *argz*, size\_t *argz\_len*, int *sep*);

##### Description

The argz functions operate on argz vectors, which are typically used to more easily manipulate program arguments, of the form described in [ISO C (1999)](#ID_STD_46_ISOC99) in section 5.1.2.2.1, Program Startup. While an argv is an array of character pointers to strings, an argz vector is a set of strings, separated by null characters, in contiguous memory; the vector is described by a pointer to the first element and a size. There is no limitation that the argz must be made up of program arguments.

The argz functions which change argz vectors expect them to use memory allocated using malloc(), and will themselves use malloc() or realloc().

The argz\_create() function converts an argv vector identified by argv to an argz vector with the same elements, identified by argz and argz\_len.

The argz\_create\_sep() function converts the string identified by str, spliting into a separate string at each occurence of sep, to an argz vector identified by argz and argz\_len.

The argz\_add() function adds the string identified by str to the vector identified by argz and argz\_len, updating argz and argz\_len.

The argz\_add\_sep() function adds the string identified by str, spliting into a separate string at each occurence of sep, to the vector identified by argz, updating argz and argz\_len.

The argz\_append() function appends the argz vector identified by buf and buf\_len to the argz vector identified by argz and argz\_len, thus updating argz and argz\_len.

The argz\_count() function returns the number of strings in the argz vector identified by argz and argz\_len.

The argz\_delete() function removes the string identified by entry from the the argz vector identified by argz, argz\_len, updating argz and argz\_len.

The argz\_extract() function performs the inverse of argz\_create(). It converts an argz vector identified by argz and argz\_len to an argv vector identified by argv with the same elements.

The argz\_insert() function inserts the string identified by entry at position before to the the argz vector identified by argz and argz\_len, updating argz and argz\_len.

The argz\_next() function returns the entry following the entry identfied by entry in the argz vector identified by argz and argz\_len. If entry is NULL the first entry is returned. This function can be used to step through an argz vector by obtaining the first entry by passing NULL, then passing the just obtained value to the next call, and so on. NULL is returned if there is no following entry.

The argz\_replace() function replaces each occurrence of str in the argz vector identified by argz and argz\_len with with, updating argz and argz\_len. The counter pointed to by replace\_count will be incremented by the number of replacements unless NULL is passed for replace\_count.

The argz\_stringify() function performs the inverse of argz\_create\_sep(). It converts the argz vector identified by argz and argz\_len into a regular string, with the strings in the original vector separated by sep in the converted string. The conversion is done in place, so in effect each null byte in argz but the last one is replaced by sep.

##### Return Value

All of the argz functions that perform memory allocation return an error\_t type. These functions return 0 on success; if memory allocation fails, they return ENOMEM.

argz\_count() returns a count of substrings in the argz vector as a size\_t type.

argz\_next() returns a pointer to a substring in an argz vector, or NULL.

##### See Also

[envz\_add, envz\_entry, envz\_get, envz\_merge, envz\_remove, envz\_strip](#ID_BASELIB_45_ENVZ_45_ADD)

#### asprintf

##### Name

asprintf — write formatted output to a dynamically allocated string

##### Synopsis

#include <stdio.h>

int asprintf(char \*\* restrict *ptr*, const char \* restrict *format*, ...);

##### Description

The asprintf() function shall behave as sprintf(), except that the output string shall be dynamically allocated space of sufficient length to hold the resulting string. The address of this dynamically allocated string shall be stored in the location referenced by *ptr*.

##### Return Value

Refer to fprintf().

##### Errors

Refer to fprintf().

#### backtrace, backtrace\_symbols, backtrace\_symbols\_fd

##### Name

backtrace, backtrace\_symbols, backtrace\_symbols\_fd — runtime stack back tracing

##### Synopsis

#include <execinfo.h>

int backtrace(void \*\**array*, int *size*);

char \*\*backtrace\_symbols(void \*const \**array*, int *size*);

void backtrace\_symbols\_fd(void \*const \**array*, int *size*, int *fd*);

##### Description

backtrace() obtains a backtrace for the current thread as a list of pointers filled in to *array*. The *size* parameter describes the number of elements that will fit into *array*, backtrace() will truncate the list if necessary. A backtrace is a list of currently active function calls in a thread; each function call allocates a new stack frame and backtrace() obtains the return address from each stack frame.

backtrace\_symbols() translates the information obtained from backtrace() into an array of strings. *array* is a pointer to an array of addresses as obtained from backtrace(). *size* is the number of entries in *array*, and should be the return value of the call to backtrace(). The strings contain the function name if it can be determined, a hedxadecimal offset into the function, and the actual return address in hexadecimal. Note that the pointer returned by backtrace\_symbols() is obtained by an internal call to malloc() and should be freed when no longer needed.

backtrace\_symbols\_fd() performs the same transformation as backtrace\_symbols() given the same argument pair *array*, *size*, but writes the strings to the file descriptor contained in *fd*. This avoids the allocation of string space.

##### Return Value

backtrace() returns the number of entries placed into *array*, no more than *size*. If the value is less than *size*, the full backtrace was returned; else it may have been truncated.

On success, backtrace\_symbols() returns a pointer to an array of strings, which will have *size* entries. On error, NULL is returned.

##### Errors

No errors are defined for these functions. If backtrace\_symbols\_fd() fails, it will be due to a failure in the call to malloc(), and errno will be set accordingly.

##### Notes

The ability to obtain useful backtrace information, in particular function names, is dependent on a number of factors at the time of program construction, such as compiler optimization options. Even if the program itself is constructed so as to make symbols visible, the call trace may descend into system libraries which have not been so constructed.

Inlined functions do not have stack frames, and functions declared as static are not exposed and so will not be available in the backtrace.

##### See Also

malloc()

#### basename

##### Name

basename — return the last component of a file name

##### Synopsis

#include <libgen.h>

char \* basename(const char \* *path*);

##### Description

In the source standard, basename() is implemented as a macro causing it to behave as described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), and is equivalent to the function \_\_xpg\_basename(). If the macro is undefined, basename() from the binary standard is used, with differences as described here:

The string identified by *path* shall not be modified.

If *path* is "/", or ends with a trailing '/' character, the basename() function shall return a pointer to an empty string.

##### Return Value

On success, the basename() function shall return a pointer to the final component of *path*. Otherwise, it shall return a null pointer.

##### See Also

\_\_xpg\_basename()

#### bind\_textdomain\_codeset

##### Name

bind\_textdomain\_codeset — specify encoding for message retrieval

##### Synopsis

#include <libintl.h>

char \* bind\_textdomain\_codeset (const char \* *domainname* , const char \* *codeset* );

##### Description

The bind\_textdomain\_codeset() function can be used to specify the output codeset for message catalogs for domain *domainname*. The *codeset* argument shall be a valid codeset name which can be used tor the *iconv\_open* function, or a null pointer. If the *codeset* argument is the null pointer, then function returns the currently selected codeset for the domain with the name *domainname*. It shall return a null pointer if no codeset has yet been selected.

Each successive call to bind\_textdomain\_codeset() function overrrides the settings made by the preceding call with the same *domainname*.

The bind\_textdomain\_codeset() function shall return a pointer to a string containing the name of the selected codeset. The string shall be allocated internally in the function and shall not be changed or freed by the user.

##### Parameters

domainname

  The *domainname* argument is applied to the currently active LC\_MESSAGE locale. It is equivalent in syntax and meaning to the *domainname* argument to *textdomain*, except that the selection of the domain is valid only for the duration of the call.

codeset

  The name of the output codeset for the selected domain, or NULL to select the current codeset.

If *domainname* is the null pointer, or is an empty string, bind\_textdomain\_codeset() shall fail, but need not set errno.

##### Return Value

Returns the currently selected codeset name. It returns a null pointer if no codeset has yet been selected.

##### Errors

ENOMEM

  Insufficient memory available to allocate return value.

##### See Also

gettext, dgettext, ngettext, dngettext, dcgettext, dcngettext, textdomain, bindtextdomain

#### bindresvport

##### Name

bindresvport — bind socket to privileged IP port

##### Synopsis

#include <sys/types.h>

#include <rpc/rpc.h>

int bindresvport(int *sd*, struct sockaddr\_in \* *sin*);

##### Description

If the process has appropriate privilege, the bindresvport() function shall bind a socket to an anonymous privileged IP port, that is, arbitrarily selected from the range 512 through 1023.

If the bind is successful and *sin* is not NULL, and the port number bound to is returned in the *sin\_port* member of *sin*. Any caller-supplied value of *sin\_port* is ignored.

If *sin* is NULL, the address family is taken to be AF\_INET and an available privileged port is bound to. Since there is no sockaddr\_in structure, the port number chosen cannot be returned. The getsockname() may be used to query for this information.

##### Return Value

On success, 0 is returned. On error, -1 is returned and errno is set to indicate the error.

##### Errors

bindresvport() may fail in the same way as bind() in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4). The following additional or differing failures may occur:

EADDRINUSE

  All privileged ports are in use.

EAFNOSUPPORT

  The specified address is not a valid address for the address family of the specified socket, or the address family is not supported.

EPFNOSUPPORT

  The same meaning as EAFNOSUPPORT. Some older implementations may return this error instead.

**Note:** At this time, only AF\_INET is supported. Applications should be prepared for either the EAFNOSUPPORT or EPFNOSUPPORT error to be indicated.

#### bindtextdomain

##### Name

bindtextdomain — specify the location of a message catalog

##### Synopsis

#include <libintl.h>

char \* bindtextdomain(const char \* *domainname*, const char \* *dirname*);

##### Description

The bindtextdomain() shall set the the base directory of the hierarchy containing message catalogs for a given message domain.

The bindtextdomain() function specifies that the *domainname* message catalog can be found in the *dirname* directory hierarchy, rather than in the system default locale data base.

If *dirname* is not NULL, the base directory for message catalogs belonging to domain *domainname* shall be set to *dirname*. If *dirname* is NULL, the base directory for message catalogs shall not be altered.

The function shall make copies of the argument strings as needed.

*dirname* can be an absolute or relative pathname.

**Note:** Applications that wish to use chdir() should always use absolute pathnames to avoid misadvertently selecting the wrong or non-existant directory.

If *domainname* is the null pointer, or is an empty string, bindtextdomain() shall fail, but need not set errno.

The bindtextdomain() function shall return a pointer to a string containing the name of the selected directory. The string shall be allocated internally in the function and shall not be changed or freed by the user.

##### Return Value

On success, bindtextdomain() shall return a pointer to a string containing the directory pathname currently bound to the domain. On failure, a NULL pointer is returned, and the global variable errno may be set to indicate the error.

##### Errors

ENOMEM

  Insufficient memory was available.

##### See Also

gettext, dgettext, ngettext, dngettext, dcgettext, dcngettext, textdomain, bind\_textdomain\_codeset

#### cfmakeraw

##### Name

cfmakeraw — get and set terminal attributes

##### Synopsis

#include <termios.h>

void cfmakeraw(struct termios \* *termios\_p*);

##### Description

The cfmakeraw() function shall set the attributes of the termios structure referenced by *termios\_p* as follows:

termios\_p->c\_iflag &= ~(IGNBRK|BRKINT|PARMRK|ISTRIP

|INLCR|IGNCR|ICRNL|IXON);

termios\_p->c\_oflag &= ~OPOST;

termios\_p->c\_lflag &= ~(ECHO|ECHONL|ICANON|ISIG|IEXTEN);

termios\_p->c\_cflag &= ~(CSIZE|PARENB);

termios\_p->c\_cflag |= CS8;

*termios\_p* shall point to a termios structure that contains the following members:

tcflag\_t c\_iflag; /\* input modes \*/

tcflag\_t c\_oflag; /\* output modes \*/

tcflag\_t c\_cflag; /\* control modes \*/

tcflag\_t c\_lflag; /\* local modes \*/

cc\_t c\_cc[NCCS]; /\* control chars \*/

#### cfsetspeed

##### Name

cfsetspeed — set terminal input and output data rate

##### Synopsis

#include <termios.h>

int cfsetspeed(struct termios \**t*, speed\_t *speed*);

##### Description

The cfsetspeed() function shall set the input and output speeds in *t* to the value specified by *speed*. The effects of the function on the terminal as described below do not become effective, nor are all errors detected, until the tcsetattr() function is called. Certain values for baud rates set in termios and passed to tcsetattr() have special meanings.

##### Return Value

On success, 0 is returned. On error, -1 is returned and the global variable errno is set appropriately.

##### Errors

EINVAL

  Invalid *speed* argument

#### clearerr\_unlocked

##### Name

clearerr\_unlocked — non-thread-safe clearerr

##### Description

clearerr\_unlocked() is the same as clearerr(), except that it need not be thread-safe. That is, it may only be invoked in the ways which are legal for getc\_unlocked().

#### daemon

##### Name

daemon — run in the background

##### Synopsis

#include <unistd.h>

int daemon(int *nochdir*, int *noclose*);

##### Description

The daemon() function shall create a new process, detached from the controlling terminal. If successful, the calling process shall exit and the new process shall continue to execute the application in the background. If *nochdir* evaluates to true, the current directory shall not be changed. Otherwise, daemon() shall change the current working directory to the root (`/'). If *noclose* evaluates to true the standard input, standard output, and standard error file descriptors shall not be altered. Otherwise, daemon() shall close the standard input, standard output and standard error file descriptors and reopen them attached to /dev/null.

##### Return Value

On error, -1 is returned, and the global variable errno is set to any of the errors specified for the library functions fork() and setsid().

#### dcgettext

##### Name

dcgettext — perform domain and category specific lookup in message catalog

##### Synopsis

#include <libintl.h>

#include <locale.h>

char \* dcgettext(const char \* *domainname*, const char \* *msgid*, int *category*);

##### Description

The dcgettext() function is a domain specified version of gettext().

The dcgettext() function shall lookup the translation in the current locale of the message identified by *msgid* in the domain specified by *domainname* and in the locale category specified by *category*. If *domainname* is NULL, the current default domain shall be used. The *msgid* argument shall be a NULL-terminated string to be matched in the catalogue. *category* shall specify the locale category to be used for retrieving message strings. The category parameter shall be one of *LC\_CTYPE*, *LC\_COLLATE*, *LC\_MESSAGES*, *LC\_MONETARY*, *LC\_NUMERIC*, or *LC\_TIME*. The default domain shall not be changed by a call to dcgettext().

##### Return Value

If a translation was found in one of the specified catalogs, it shall be converted to the current locale's codeset and returned. The resulting NULL-terminated string shall be allocated by the dcgettext function, and must not be modified or freed. If no translation was found, or category was invalid, *msgid* shall be returned.

##### Errors

dcgettext() shall not modify the errno global variable.

##### See Also

gettext, dgettext, ngettext, dngettext, dcngettext, textdomain, bindtextdomain, bind\_textdomain\_codeset

#### dcngettext

##### Name

dcngettext — perform domain and category specific lookup in message catalog with plural

##### Synopsis

#include <libintl.h>

#include <locale.h>

char \* dcngettext(const char \* *domainname*, const char \* *msgid1*, const char \* *msgid2*, unsigned long int *n*, int *category*);

##### Description

The dcngettext() function is a domain specific version of gettext, capable of returning either a singular or plural form of the message. The dcngettext() function shall lookup the translation in the current locale of the message identified by *msgid1* in the domain specified by *domainname* and in the locale category specified by *category*. If *domainname* is NULL, the current default domain shall be used. The *msgid1* argument shall be a NULL-terminated string to be matched in the catalogue. *category* shall specify the locale category to be used for retrieving message strings. The *category* parameter shall be one of *LC\_CTYPE*, *LC\_COLLATE*, *LC\_MESSAGES*, *LC\_MONETARY*, *LC\_NUMERIC*, or *LC\_TIME*. The default domain shall not be changed by a call to dcngettext(). If *n* is 1 then the singular version of the message is returned, otherwise one of the plural forms is returned, depending on the value of *n* and the current locale settings.

##### Return Value

If a translation corresponding to the value of *n* was found in one of the specified catalogs for *msgid1*, it shall be converted to the current locale's codeset and returned. The resulting NULL-terminated string shall be allocated by the dcngettext() function, and must not be modified or freed. If no translation was found, or *category* was invalid, *msgid1* shall be returned if *n* has the value 1, otherwise *msgid2* shall be returned.

##### Errors

dcngettext() shall not modify the errno global variable.

##### See Also

gettext, dgettext, ngettext, dngettext, dcgettext, textdomain, bindtextdomain, bind\_textdomain\_codeset

#### dgettext

##### Name

dgettext — perform lookup in message catalog for the current LC\_MESSAGES locale

##### Synopsis

#include <libintl.h>

char \* dgettext(const char \* *domainname*, const char \* *msgid*);

##### Description

dgettext() is a domain specified version of gettext().

The dgettext() function shall search the currently selected message catalogs in the domain *domainname* for a string identified by the string *msgid*. If a string is located, that string shall be returned. The domain specified by *domainname* applies to the currently active LC\_MESSAGE locale. The default domain shall not be changed by a call to dgettext().

**Note:** The usage of *domainanme* is equivalent in syntax and meaning to the textdomain() function's application of *domainname*, except that the selection of the domain in dgettext() is valid only for the duration of the call.

The dgettext() function is equivalent to dcgettext(domainname, msgid, LC\_MESSAGES).

##### Return Value

On success of a *msgid* query, the translated NULL-terminated string is returned. On error, the original *msgid* is returned. The length of the string returned is undetermined until dgettext() is called.

##### Errors

dgettext() shall not modify the errno global variable.

##### See Also

gettext, dgettext, ngettext, dngettext, dcgettext, dcngettext, textdomain, bindtextdomain, bind\_textdomain\_codeset

#### dl\_iterate\_phdr

##### Name

dl\_iterate\_phdr — iterate over a program's loaded shared objects

##### Synopsis

#include <link.h>

int dl\_iterate\_phdr(int(\**callback*) (struct dl\_phdr\_info \*, size\_t, void \*), void \**data*);

##### Description

dl\_iterate\_phdr() allows a program to iterate over the shared objects it has loaded. The function described by the *callback* parameter is called once for each loaded shared object, allowing an action to be taken for each one. *callback* is called with three arguments which are filled in by the implementation: a pointer to a structure of type dl\_phdr\_info containing information about the shared object; an integer size of the structure; and a copy of the *data* argument to dl\_iterate\_phdr(). If *callback* returns a non-zero value, dl\_iterate\_phdr() will stop processing, even if there are unprocessed shared objects. The order of processing is unspecified.

The dl\_phdr\_info structure has the following members (note that on 64-bit architectures the types here shown as Elf32\_*type* will instead be Elf64\_*type*):

Elf32\_Addr dlpi\_addr;

const char \*dlpi\_name;

const Elf32\_Phdr \*dlpi\_phdr;

Elf32\_Half dlpi\_phnum;

unsigned long long int dlpi\_adds;

unsigned long long int dlpi\_subs;

size\_t dlpi\_tls\_modid;

void \*dlpi\_tls\_data;

*dlpi\_addr* contains the base address of the shared object.

*dlpi\_name* is a null-terminated string giving the pathname from which the shared object was loaded.

*dlpi\_phdr* is a pointer to an array of program headers for this shared object, while *dlpi\_phnum* is the number of entries in this array.

*dlpi\_adds* and *dlpi\_subs* are incremented when shared objects are added or removed, respectively.

*dlpi\_tls\_modid* contains the module ID used in TLS relocations, if there is a PT\_TLS segment. Otherwise the value shall be zero.

*dlpi\_tls\_data* contains the address of the calling thread's instance of this module's PT\_TLS segment, if there is one and it has been allocated in the calling thread. Otherwise the value shall be a null pointer.

Some implementations may not provide all fields in dl\_phdr\_info, although the first four are always mandatory. Applications are advised to have the callback function check the size parameter before examining the later members.

##### Return Value

The dl\_iterate\_phdr() function returns whatever value was returned by the last call to *callback*. This will be zero if processing completed normally, since processing does not continue unless the callback function returns zero.

##### Errors

No errors are defined by dl\_iterate\_phdr(); as noted the callback function must use a zero return to indicate success but may assign any meaning it wishes to non-zero returns.

#### dngettext

##### Name

dngettext — perform lookup in message catalog for the current locale

##### Synopsis

#include <libintl.h>

char \* dngettext(const char \* *domainname*, const char \* *msgid1*, const char \* *msgid2*, unsigned long int *n*);

##### Description

dngettext() shall be equivalent to a call to

dcngettext(domainname, msgid1, msgid2, n, LC\_MESSAGES)

See dcngettext() for more information.

##### See Also

gettext, dgettext, ngettext, dcgettext, dcngettext, textdomain, bindtextdomain, bind\_textdomain\_codeset

#### drand48\_r

##### Name

drand48\_r — reentrantly generate pseudorandom numbers in a uniform distribution

##### Synopsis

#include <stdlib.h>

int drand48\_r(struct drand48\_data \* *buffer*, double \* *result*);

##### Description

The interface drand48\_r() shall function in the same way as the interface drand48(), except that drand48\_r() shall use the data in *buffer* instead of the global random number generator state.

Before it is used, *buffer* must be initialized, for example, by calling lcong48\_r(), seed48\_r(), or srand48\_r(), or by filling it with zeroes.

#### endutent

##### Name

endutent — access utmp file entries

##### Synopsis

#include <utmp.h>

void endutent(void);

##### Description

endutent() closes the utmp file. It should be called when the user code is done accessing the file with the other functions.

#### envz\_add, envz\_entry, envz\_get, envz\_merge, envz\_remove, envz\_strip

##### Name

envz\_add, envz\_entry, envz\_get, envz\_merge, envz\_remove, envz\_strip — Operate on environment vectors

##### Synopsis

#include <envz.h>

error\_t envz\_add(char \*\* *envz*, size\_t \* *envz\_len*, const char \* *name*, const char \* *value*);

char envz\_entry(const char \* *envz*, size\_t *envz\_len*, const char \* *name*);

char envz\_get(const char \* *envz*, size\_t *envz\_len*, const char \* *name*);

error\_t envz\_merge(char \*\* *envz*, size\_t \* *envz\_len*, const char \* *envz2*, size\_t *envz2\_len*, int *override*);

void envz\_remove(char \*\* *envz*, size\_t \* *envz\_len*, const char \* *name*);

void envz\_strip(char \*\* *envz*, size\_t \* *envz\_len*);

##### Description

The envz functions operate on envz vectors, which are typically used to manipulate program environment variables.

An envz vector is identical in makeup to an argz vector (see [argz\_add, argz\_add\_sep, argz\_append, argz\_count, argz\_create, argz\_create\_sep, argz\_delete, argz\_extract, argz\_insert, argz\_next, argz\_replace, argz\_stringify](#ID_BASELIB_45_ARGZ_45_ADD)) but has the constraint that each element is a name, value pair separated by an = character. Only the first = character in an element has special meaning, any subsequent instances are part of the value string. If no = character is present in an element, the value is taken to be NULL. If an element has an empty value (an = character is present), the value will return the empty string "" when queried.

Since an envz vector is an argz vector, the argz functions can be used where it makes sense. For example, converting from a program's environment variables (as described in Chapter 8 of the XBD volume of [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4)) to an envz vector is done with argz\_create().

The envz\_add() function adds a string constructed from name and value in the form "name=value" to the envz vector identified by envz and envz\_len, updating envz and envz\_len. If value is NULL it adds a string of the form "name". If an entry with the same name already exists, it is replaced..

The envz\_entry() function searches for name in the envz vector identified by envz and envz\_len, returning the full entry if found, or NULL if not.

The envz\_get() function searches for name in the envz vector identified by envz and envz\_len, returning the value part of the entry if found, or NULL if not. Note the value may be also NULL.

The envz\_merge() function adds each entry from the envz vector identified by envz2 and envz2\_len to the envz vector identified by envz and envz\_len, updating envz and envz\_len. The behavior is as if envz\_add() were called for each entry in envz2. If override is true, then values from envz2 will replace those with the same name in envz.

The envz\_remove() function removes the entry for name from the envz vector identified by envz and envz\_len if it exists, updating envz and envz\_len.

The envz\_strip() function removes all entries with value NULL.

##### Return Value

The envz functions that perform memory allocation (envz\_add() and envz\_merge()) return an error\_t type. These functions return 0 on success; if memory allocation fails, they return ENOMEM.

envz\_entry() and envz\_get() return a pointer to a substring in an envz vector, or NULL.

##### See Also

[argz\_add, argz\_add\_sep, argz\_append, argz\_count, argz\_create, argz\_create\_sep, argz\_delete, argz\_extract, argz\_insert, argz\_next, argz\_replace, argz\_stringify](#ID_BASELIB_45_ARGZ_45_ADD)

#### epoll\_create

##### Name

epoll\_create — open an epoll file descriptor

##### Synopsis

#include <sys/epoll.h>

int epoll\_create(int *size*);

##### Description

The epoll API, which consists of the interfaces epoll\_create(), epoll\_ctl(), and epoll\_wait(), shall support all file descriptors compatible with poll(). These interfaces shall be usable in either level-triggered or edge-triggered mode. In level-triggered mode, epoll has similar semantics to poll(), and can be used as a faster replacement for it. In edge-triggered mode, epoll shall only report events for a file descriptor when changes occur on it.

The epoll\_create() interface shall open an epoll file descriptor by allocating an event backing store of approximately size *size*. The *size* parameter is a hint to the kernel about how large the event storage should be, not a rigidly-defined maximum size.

##### Return Value

On success, epoll\_create() shall return the file descriptor, a non-negative integer that shall be used for subsequent epoll calls. It should be closed with the close() function.

On failure, epoll\_create() shall return -1 and set errno as follows.

##### Errors

EINVAL

  The *size* parameter is not positive.

ENFILE

  The maximum number of open files has been reached by the system.

ENOMEM

  Not enough memory to create the kernel object.

##### See Also

close(), epoll\_ctl(), epoll\_wait(), poll().

#### epoll\_ctl

##### Name

epoll\_ctl — control an epoll file descriptor

##### Synopsis

#include <sys/epoll.h>

int epoll\_ctl(int *epfd*, int *op*, int *fd*, struct epoll\_event \* *event*);

##### Description

The interface epoll\_ctl() shall control an epoll file descriptor.

The parameter *epfd* shall specify the epoll file descriptor to control.

The parameter *op* shall specify the operation to perform on the specified target file descriptor.

The parameter *fd* shall specify the target file descriptor on which to perform the specified operation.

The parameter *event* shall specify the object associated with the target file descriptor. The events member of the *event* parameter is a bit set composed of the event types listed below.

##### Event types

EPOLLERR

  An error condition occurred on the target file descriptor. It shall not be necessary to set this event in events; this interface shall always wait for it.

EPOLLET

  This event shall set edge-triggered behavior for the target file descriptor. The default epoll behavior shall be level-triggered.

EPOLLHUP

  A hang up occurred on the target file descriptor. It shall not be necessary to set this event in events; this interface shall always wait for it.

EPOLLIN

  The file is accessible to read() operations.

EPOLLONESHOT

  This event shall set one-shot behavior for the target file descriptor. After epoll\_wait() retrieves an event, the file descriptor shall be disabled and epoll shall not report any other events. To reenable the file descriptor with a new event mask, the user should invoke epoll\_ctl() with EPOLL\_CTL\_MOD in the *op* parameter.

EPOLLOUT

  The file is accessible to write() operations.

EPOLLPRI

  Urgent data exists for read() operations.

EPOLLRDHUP

  A stream socket peer closed the connection, or else the peer shut down the writing half of the connection.

##### Values of the op parameter

EPOLL\_CTL\_ADD

  Associate *event* with the file described by *fd*, and add *fd* to the epoll descriptor *epfd*.

EPOLL\_CTL\_DEL

  Remove *fd* from *epfd*, and ignore *event*, which can be NULL.

EPOLL\_CTL\_MOD

  Change the event *event* associated with *fd*.

##### Return Value

On success, epoll\_ctl() shall return 0.

On failure, epoll\_ctl() shall return -1 and set errno as follows.

##### Errors

EBADF

  The parameter *epfd* or the parameter *fd* is an invalid file descriptor.

EEXIST

  The parameter *op* was EPOLL\_CTL\_ADD, but the file descriptor *fd* is already in *epfd*.

EINVAL

  The parameter *epfd* is invalid, or it is the same as *fd*, or the operation specified by the parameter *op* is unsupported.

ENOENT

  The parameter *op* was EPOLL\_CTL\_MOD or EPOLL\_CTL\_DEL, but the file descriptor *fd* is not in *epfd*.

ENOMEM

  Not enough memory for the operation specified by the parameter *op*.

EPERM

  The file specified by *fd* does not support epoll.

##### See Also

close(), epoll\_create(), epoll\_wait(), poll().

#### epoll\_wait

##### Name

epoll\_wait — wait for I/O events on an epoll file descriptor

##### Synopsis

#include <sys/epoll.h>

int epoll\_wait(int *epfd*, struct epoll\_event \* *events*, int *maxevents*, int *timeout*);

##### Description

The interface epoll\_wait() shall wait for events on the epoll file descriptor specified by the parameter *epfd*.

Upon success, the output parameter *events* shall refer to an area of memory containing epoll\_event structures available to the caller. The data members of these structures shall contain the data set by the user with the interface epoll\_ctl(). The events members shall contain the event bit field that was returned.

The parameter *maxevents* shall specify the maximum number of events that epoll\_wait() may return in the output parameter *events*. The value of this parameter should be greater than 0.

The parameter *timeout* shall specify the maximum number of milliseconds that epoll\_wait() shall wait for events. If the value of this parameter is 0, then epoll\_wait() shall return immediately, even if no events are available, in which case the return code shall be 0. If the value of *timeout* is -1, then epoll\_wait() shall block until either a requested event occurs or the call is interrupted.

##### Return Value

On success, epoll\_wait() shall return the number of file descriptors that are ready for the I/O that was requested, or else 0 if no descriptors became ready during *timeout*.

On failure, epoll\_wait() shall return -1 and set errno as follows.

##### Errors

EBADF

  The parameter *epfd* is not a valid file descriptor.

EFAULT

  The area of memory referenced by the parameter *events* cannot be accessed with write permissions.

EINTR

  The call was interrupted by a signal handler before the *timeout* expired or any requested event took place.

EINVAL

  The parameter *epfd* is not a valid epoll file descriptor, or else the parameter *maxevents* is less than or equal to 0.

##### See Also

close(), epoll\_ctl(), epoll\_create(), poll().

#### erand48\_r

##### Name

erand48\_r — reentrantly generate pseudorandom numbers in a uniform distribution

##### Synopsis

#include <stdlib.h>

int erand48\_r(unsigned short[3] *xsubi*, struct drand48\_data \* *buffer*, double \* *result*);

##### Description

The interface erand48\_r() shall function in the same way as the interface erand48(), except that erand48\_r() shall use the data in *buffer* instead of the global random number generator state.

Before it is used, *buffer* must be initialized, for example, by calling lcong48\_r(), seed48\_r(), or srand48\_r(), or by filling it with zeroes.

#### err

##### Name

err — display formatted error messages

##### Synopsis

#include <err.h>

void err (int *eval* , const char \* *fmt* , ...);

##### Description

The err() function shall display a formatted error message on the standard error stream. First, err() shall write the last component of the program name, a colon character, and a space character. If *fmt* is non-NULL, it shall be used as a format string for the printf() family of functions, and err() shall write the formatted message, a colon character, and a space. Finally, the error message string affiliated with the current value of the global variable errno shall be written, followed by a newline character.

The err() function shall not return, the program shall terminate with the exit value of *eval*.

##### See Also

error(), errx()

##### Return Value

None.

##### Errors

None.

#### error

##### Name

error — print error message

##### Synopsis

#include <error.h>

void error (int *status* , int *errnum* , const char \* *format* , ...);

##### Description

error() shall print a message to standard error.

error() shall build the message from the following elements in their specified order:

1. the program name. If the application has provided a function named error\_print\_progname(), error() shall call this to supply the program name; otherwise, error() uses the content of the global variable program\_name.

2. the colon and space characters, then the result of using the printf-style *format* and the optional arguments.

3. if *errnum* is nonzero, error() shall add the colon and space characters, then the result of strerror(errnum).

4. a newline.

If *status* is nonzero, error() shall call exit(status).

##### See Also

err(), errx()

#### errx

##### Name

errx — display formatted error message and exit

##### Synopsis

#include <err.h>

void errx (int *eval* , const char \* *fmt* , ...);

##### Description

The errx() function shall display a formatted error message on the standard error stream. The last component of the program name, a colon character, and a space shall be output. If *fmt* is non-NULL, it shall be used as the format string for the printf() family of functions, and the formatted error message, a colon character, and a space shall be output. The output shall be followed by a newline character.

errx() does not return, but shall exit with the value of *eval*.

##### Return Value

None.

##### Errors

None.

##### See Also

error(), err()

#### fcntl

##### Name

fcntl — file control

##### Description

fcntl() is as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but with differences as listed below.

#### Implementation may set O\_LARGEFILE

According to [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), only an application sets fcntl() flags, for example O\_LARGEFILE. However, this specification also allows an implementation to set the O\_LARGEFILE flag in the case where the programming environment is one of \_POSIX\_V6\_ILP32\_OFFBIG, \_POSIX\_V6\_LP64\_OFF64, \_POSIX\_V6\_LPBIG\_OFFBIG. See **getconf** and **c99** in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) for a description of these environments. Thus, calling fcntl() with the *F\_GETFL* command may return O\_LARGEFILE as well as flags explicitly set by the application in the case that both the implementation and the application support an off\_t of at least 64 bits.

#### Additional flags

In addition to the available values for *cmd*, as documented in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), this specification permits the following constants.

F\_GETSIG shall get the number of the signal to be sent when input or output can occur. If the value is 0, then SIGIO shall be sent. Otherwise, the value retrieved shall be the signal sent, and the signal handler can discover more information when installed with the SA\_SIGINFO flag.

F\_SETSIG shall set the number of the signal to be sent when input or output can occur. If the value is 0, then SIGIO shall be sent. Otherwise, the value set shall be the signal to be sent, and the signal handler can discover more information when installed with the SA\_SIGINFO flag.

F\_GETLK64 is analogous to the F\_GETLK constant in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but shall provide a 64-bit interface on non-64-bit architectures. It is identical to F\_GETLK on a 64-bit machine, but is provided in 64-bit environments for source code consistency among architectures.

F\_SETLK64 is analogous to the F\_SETLK constant in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but shall provide a 64-bit interface on non-64-bit architectures. It is identical to F\_SETLK on a 64-bit machine, but is provided in 64-bit environments for source code consistency among architectures.

F\_SETLKW64 is analogous to the F\_SETLKW constant in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but provides a 64-bit interface on non-64-bit architectures. It is identical to F\_SETLKW on a 64-bit machine, but is provided in 64-bit environments for source code consistency among architectures.

#### feof\_unlocked

##### Name

feof\_unlocked — non-thread-safe feof

##### Description

feof\_unlocked() is the same as feof(), except that it need not be thread-safe. That is, it may only be invoked in the ways which are legal for getc\_unlocked().

#### ferror\_unlocked

##### Name

ferror\_unlocked — non-thread-safe ferror

##### Description

ferror\_unlocked() is the same as ferror(), except that it need not be thread-safe. That is, it may only be invoked in the ways which are legal for getc\_unlocked().

#### fflush\_unlocked

##### Name

fflush\_unlocked — non thread safe fflush

##### Description

fflush\_unlocked() is the same as fflush() except that it need not be thread safe. That is, it may only be invoked in the ways which are legal for getc\_unlocked().

#### fgetc\_unlocked

##### Name

fgetc\_unlocked — non-thread-safe fgetc

##### Description

fgetc\_unlocked() is the same as fgetc(), except that it need not be thread-safe. That is, it may only be invoked in the ways which are legal for getc\_unlocked().

#### fgets\_unlocked

##### Name

fgets\_unlocked — non-thread-safe fgets

##### Description

fgets\_unlocked() is the same as fgets(), except that it need not be thread-safe. That is, it may only be invoked in the ways which are legal for getc\_unlocked().

#### fgetwc\_unlocked

##### Name

fgetwc\_unlocked — non thread safe fgetwc

##### Description

fgetwc\_unlocked() is the same as fgetwc() except that it need not be thread safe. That is, it may only be invoked in the ways which are legal for getc\_unlocked().

#### fgetws\_unlocked

##### Name

fgetws\_unlocked — non-thread-safe fgetws

##### Description

fgetws\_unlocked() is the same as fgetws(), except that it need not be thread-safe. That is, it may only be invoked in the ways which are legal for getc\_unlocked().

#### fileno\_unlocked

##### Name

fileno\_unlocked — non-thread-safe fileno

##### Description

fileno\_unlocked() is the same as fileno(), except that it need not be thread-safe. That is, it may only be invoked in the ways which are legal for getc\_unlocked().

#### flock

##### Name

flock — apply or remove an advisory lock on an open file

##### Synopsis

int flock(int *fd*, int *operation*);

##### Description

flock() applies or removes an advisory lock on the open file *fd*. Valid *operation* types are:

LOCK\_SH

  Shared lock. More than one process may hold a shared lock for a given file at a given time.

LOCK\_EX

  Exclusive lock. Only one process may hold an exclusive lock for a given file at a given time.

LOCK\_UN

  Unlock.

LOCK\_NB

  Don't block when locking. May be specified (by *or*ing) along with one of the other operations.

A single file may not simultaneously have both shared and exclusive locks.

##### Return Value

On success, 0 is returned. On error, -1 is returned and the global variable errno is set appropriately.

##### Errors

EWOULDBLOCK

  The file is locked and the LOCK\_NB flag was selected.

EBADF

*fd* is not a not an open file descriptor.

EINTR

  While waiting to acquire a lock, the call was interrupted by delivery of a signal caught by a handler.

EINVAL

  The operation is invalid.

ENOLCK

  The implementation ran out of memory for allocating lock records.

#### fnmatch

##### Name

fnmatch — match a filename or a pathname

##### Description

fnmatch() is as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but with differences as listed below.

#### Additional flags

In addition to the available values that can be used to form *flags*, as documented in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), this specification permits the following constants.

FNM\_CASEFOLD

  If this flag is set, the pattern is matched case-insensitively.

FNM\_FILE\_NAME

  A synonym for FNM\_PATHNAME constant specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

#### fputc\_unlocked

##### Name

fputc\_unlocked — non-thread-safe fputc

##### Description

fputc\_unlocked() is the same as fputc(), except that it need not be thread-safe. That is, it may only be invoked in the ways which are legal for getc\_unlocked().

#### fputs\_unlocked

##### Name

fputs\_unlocked — non-thread-safe fputs

##### Description

fputs\_unlocked() is the same as fputs(), except that it need not be thread-safe. That is, it may only be invoked in the ways which are legal for getc\_unlocked().

#### fputwc\_unlocked

##### Name

fputwc\_unlocked — non-thread-safe fputwc

##### Description

fputwc\_unlocked() is the same as fputwc(), except that it need not be thread-safe. That is, it may only be invoked in the ways which are legal for getc\_unlocked().

#### fputws\_unlocked

##### Name

fputws\_unlocked — non-thread-safe fputws

##### Description

fputws\_unlocked() is the same as fputws(), except that it need not be thread-safe. That is, it may only be invoked in the ways which are legal for getc\_unlocked().

#### fread\_unlocked

##### Name

fread\_unlocked — non-thread-safe fread

##### Description

fread\_unlocked() is the same as fread(), except that it need not be thread-safe. That is, it may only be invoked in the ways which are legal for getc\_unlocked().

#### fscanf

##### Name

fscanf — convert formatted input

##### Description

The scanf() family of functions shall behave as described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except as noted below.

##### Differences

The %s, %S and %[ conversion specifiers shall accept an option length modifier a, which shall cause a memory buffer to be allocated to hold the string converted. In such a case, the argument corresponding to the conversion specifier should be a reference to a pointer value that will receive a pointer to the allocated buffer. If there is insufficient memory to allocate a buffer, the function may set errno to ENOMEM and a conversion error results.

**Note:** This directly conflicts with the [ISO C (1999)](#ID_STD_46_ISOC99) usage of %a as a conversion specifier for hexadecimal float values. While this conversion specifier should be supported, a format specifier such as "%aseconds" will have a different meaning on an LSB conforming system.

#### fstatfs

##### Name

fstatfs — (deprecated)

##### Synopsis

#include <sys/statfs.h>

int fstatfs(int *fd*, struct statfs \* *buf*);

##### Description

The fstatfs() function returns information about a mounted file system. The file system is identified by *fd*, a file descriptor of an open file within the mounted filesystem. The results are placed in the structure pointed to by *buf*.

Fields that are undefined for a particular file system shall be set to 0.

**Note:** Application developers should use the fstatvfs() function to obtain general file system information. Applications should only use the fstatfs() function if they must determine the file system type, which need not be provided by fstatvfs().

##### Return Value

On success, the fstatfs() function shall return 0 and set the fields of the structure idenfitied by *buf* accordingly. On error, the fstatfs() function shall return -1 and set errno accordingly.

##### Errors

EBADF

*fd* is not a valid open file descriptor.

EFAULT

*buf* points to an invalid address.

EIO

  An I/O error occurred while reading from or writing to the file system.

ENOSYS

  The filesystem *fd* is open on does not support statfs().

#### fstatfs64

##### Name

fstatfs64 — (deprecated)

##### Synopsis

#include <sys/statfs.h>

int fstatfs64(int *fd*, struct statfs64 \* *buf*);

##### Description

The fstatfs64() function returns information about a mounted file system. The file system is identified by *fd*, a file descriptor of an open file within the mounted filesystem. The results are placed in the structure pointed to by *buf*.

Fields that are undefined for a particular file system shall be set to 0.

fstatfs64() is a large-file version of the fstatfs() function.

**Note:** Application developers should use the fstatvfs64() function to obtain general file system information. Applications should only use the fstatfs64() function if they must determine the file system type, which need not be provided by fstatvfs64().

##### Return Value

On success, the fstatfs64() function shall return 0 and set the fields of the structure idenfitied by *buf* accordingly. On error, the fstatfs64() function shall return -1 and set errno accordingly.

##### Errors

See fstatfs().

#### futimes

##### Name

futimes, lutimes — set file access and modification times

##### Synopsis

#include <sys/time.h>

int futimes(int *fd*, const struct timeval *tv[2]*);

int lutimes(const char \* *filename*, const struct timeval *tv[2]*);

##### Description

The futimes() and lutimes() functions shall set the access and modification times of a file to the values of the tv argument, which is an array of two timeval structures. The behavior is as for utimes() in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

The futimes() function shall change the times of of the open file described by file descriptor fd.

The lutimes() function shall change the times of of the file pointed to by the filename argument, except that if filename refers to a symbolic link, then the link is not followed and the times of the symbolic link are changed. This is similar to supplying AT\_SYMLINK\_NOFOLLOW in the flag argument to the utimensat() function.

##### Errors

As for utimes(), but in addition:

ENOSYS

  Ths implementation does not support this function (for lutimes()).

The implementation could not access a resource needed to complete the function (for futimes()).

##### See Also

utimes(), utime(), utimensat().

#### fwrite\_unlocked

##### Name

fwrite\_unlocked — non-thread-safe fwrite

##### Description

fwrite\_unlocked() is the same as fwrite(), except that it need not be thread-safe. That is, it may only be invoked in the ways which are legal for getc\_unlocked().

#### fwscanf

##### Name

fwscanf — convert formatted input

##### Description

The scanf() family of functions shall behave as described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except as noted below.

##### Differences

The %s, %S and %[ conversion specifiers shall accept an option length modifier a, which shall cause a memory buffer to be allocated to hold the string converted. In such a case, the argument corresponding to the conversion specifier should be a reference to a pointer value that will receive a pointer to the allocated buffer. If there is insufficient memory to allocate a buffer, the function may set errno to ENOMEM and a conversion error results.

**Note:** This directly conflicts with the [ISO C (1999)](#ID_STD_46_ISOC99) usage of %a as a conversion specifier for hexadecimal float values. While this conversion specifier should be supported, a format specifier such as "%aseconds" will have a different meaning on an LSB conforming system.

#### getcwd

##### Name

getcwd — get the pathname of the current working directory

##### Synopsis

#include <unistd.h>

char \* getcwd(char \* *buf*, size\_t *size*);

##### Description

The getcwd() functions shall behave as described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except as noted below.

##### Differences

If buf is NULL, memory is allocated for buf. If size is 0, the allocation size will be the size of the pathname +1, else the requested size is allocated.

##### Changed or Added Errors

EINVAL

  The size argument is 0 and buf is not a null pointer.

ENOENT

  The current working directory has been unlinked.

#### getdomainname

##### Name

getdomainname — get NIS domain name (DEPRECATED).

##### Synopsis

#include <unistd.h>

int getdomainname (char \* *name* , size\_t *namelen* );

##### Description

If the Network Information System (NIS) is in use, getdomainname() shall copy the NIS domain name to the supplied buffer identified by *name*, with maximum length *namelen*. If the NIS domain name is not currently set, getdomainname() shall copy the string "(none)" to the *name*. If *namelen* is less than the length of the string to be copied, getdomainname() shall either truncate the string to *namelen* characters and place it in *name* (without a terminating null character), or shall fail with EINVAL.

**Note:** The NIS domain name is not the same as the domain portion of a fully qualified domain name (for example, in DNS).

The LSB does not include other NIS functions, nor does it specify how NIS may affect other database functions. No conforming application can make use of this information beyond noting whether or not the domain name has been set. If the name is set to a value other than the string "(none)", the application should not imply that NIS is in use. Similarly, if it is set to "(none)", the application should not assume that NIS is not in use, although NIS functionality may be restricted in this case.

##### Return Value

On success, getdomainname() shall return 0. Otherwise, it shall return -1 and set errno to indicate the error.

##### Errors

EINVAL

*name* is a null pointer.

EINVAL

  The buffer identified by *name* and *namelen* is of insufficient size to store the NIS domain name string, and the implementation considers this an error.

##### Future Directions

The LSB does not include other NIS interfaces, and a future version of this specification may remove this interface. Application developers should avoid using this interface where possible.

#### getdtablesize

##### Name

getdtablesize — get file descriptor table size (DEPRECATED)

##### Synopsis

#include <unistd.h>

int getdtablesize (void );

##### Description

The function getdtablesize() returns the number of files a process can have open.

**Note:** The getdtablesize() function is deprecated. Portable applications should call sysconf() with the \_SC\_OPEN\_MAX option instead.

##### Return Value

The getdtablesize() function returns the current soft limit as if obtained by a call to sysconf() with the \_SC\_OPEN\_MAX option.

##### Errors

No errors are defined.

#### getgrent\_r

##### Name

getgrent\_r — reentrantly get entry in group file

##### Synopsis

#include <grp.h>

int getgrent\_r(struct group \* *gbuf*, char \* *buf*, size\_t *buflen*, struct group \* \* *gbufp*);

##### Description

The reentrant interface getgrent\_r() shall function in the same way as the interface getgrent(), except that getgrent\_r() shall return the group name, group password, and group members in buffers provided by the caller, rather than as a pointer to static storage.

The parameter *gbuf* contains the struct group that was read from the stream, if any.

The parameter *buf* contains additional strings, if any.

The parameter *buflen* specifies the size of *buf*.

The parameter *\*gbufp* returns a pointer to the struct group in *\*gbuf*.

##### Return Value

On success, getgrent\_r() shall return 0, and *\*gbufp* shall contain a pointer to the result.

On failure, *\*gbufp* shall contain NULL, and getgrent\_r() shall return an error as follows.

##### Errors

ENOENT

  No more group entries.

ERANGE

  Not enough buffer space. Specify a larger buffer and try again.

#### getgrouplist

##### Name

getgrouplist — get groups a user belongs to

##### Synopsis

#include <grp.h>

int getgrouplist(const char \* *user*, gid\_t *group*, gid\_t \* *groups*, int \* *ngroups*);

##### Description

The getgrouplist() function shall fill in the array *groups* with the supplementary groups for the user specified by *user*. On entry, *ngroups* shall refer to an integer containing the maximum number of elements in the *groups* array. The group *group* shall also be included in the values returned in *groups*. It is expected that *group* would be specified as the user's primary group from the password file (obtainable via getpwnam() or a similar function).

##### Return Value

If on entry the value referenced by *ngroups* was greater than or equal to the number of supplementary group identifiers to be copied to the array identified by *groups*, getgrouplist() shall return the number of group identifiers actually copied, and shall set the value referenced by *ngroups* to this value.

If on entry the value referenced by *ngroups* was less than the number of supplementary group identifiers, getgrouplist() shall return -1. The initial *ngroups* entries in *groups* shall be overwritten.

If the number of groups exceeds the input *ngroups* value, then as well as returning -1, *ngroups* shall be set to the number of groups that would have been placed in *groups* if it had been large enough.

**Note:** In such a case, the caller can use the information returned to make a further getgrouplist() call with a correctly sized *groups* array.

If *user* does not refer to a valid user on the system, then the behavior of this function is undefined.

##### Errors

None defined.

##### See Also

getgroups()

#### gethostbyaddr\_r

##### Name

gethostbyaddr\_r — find network host database entry matching host name (DEPRECATED)

##### Synopsis

#include <netdb.h>

int gethostbyaddr\_r(const void \* restrict *addr*, socklen\_t *len*, int *type*, struct hostent \* restrict *result\_buf*, char \* restrict *buf*, size\_t *buflen*, struct hostent \* \* restrict *result*, int \* *h\_errnop*);

##### Description

**Note:** The gethostbyaddr\_r() function is deprecated; applications should use getaddrinfo() instead.

gethostbyaddr\_r() is a reentrant version of gethostbyaddr() that searches the network host database for a host address match.

The gethostbyaddr\_r() function shall search the network host database for an entry of address family *type* with the host with address *addr*. The *len* argument contains the length of the address referenced by *addr*.

If *type* is AF\_INET, the *addr* argument shall be an in\_addr structure. If *type* is AF\_INET6, the *addr* argument shall be an in6\_addr structure. If *type* is any other value, the behavior is unspecified.

The application must provide a buffer for the gethostbyaddr\_r() to use during the lookup process. The buffer is referenced by *buf*, and is of size *buflen*. If the buffer is not of sufficient size, gethostbyaddr\_r() may fail and return ERANGE. If a matching entry is found in the database, gethostbyaddr\_r() shall copy the relevant information to the application supplied hostent structure referenced by *result\_buf*, and return a pointer to this structure in \**result*. If no matching entry is found, \**result* shall be set to a null pointer. Additional error information shall be set in the variable referenced by *h\_errnop*.

##### Return Value

On success, the gethostbyaddr\_r() function shall return zero. If the return value was ERANGE, the size of the buffer *buf*, indicated by *buflen*, was too small. If the gethostbyaddr\_r() function returns returns any other value, then the variable referenced by *h\_errnop* shall be set to indicate the cause as for gethostbyaddr().

#### gethostbyname2

##### Name

gethostbyname2 — find network host database entry matching host name (DEPRECATED)

##### Synopsis

int gethostbyname2(const char \* restrict *name*, int *af*);

##### Description

**Note:** The gethostbyname2() function is deprecated; applications should use getaddrinfo() instead.

The gethostbyname2() function shall search the network host database for an entry with name *name*. This function is similar to the gethostbyname() function but additionally allows the search to be restricted to a particular address family specified by *af*.

##### Return Value

On success, the gethostbyname2() function shall return a pointer to a hostent structure if the requested entry was found, and a null pointer otherwise.

On unsuccessful completion, gethostbyname2() shall set h\_errno as for gethostbyname() in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

##### Errors

The gethostbyname2() shall set h\_errno as for gethostbyname() in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

#### gethostbyname2\_r

##### Name

gethostbyname2\_r — find network host database entry matching host name (DEPRECATED)

##### Synopsis

int gethostbyname2\_r(const char \* restrict *name*, int *af*, struct hostent \* restrict *result\_buf*, char \* restrict *buf*, size\_t *buflen*, struct hostent \*\* restrict *result*, int \* restrict *h\_errnop*);

##### Description

**Note:** The gethostbyname2\_r() function is deprecated; applications should use getaddrinfo() instead.

The gethostbyname2\_r() function shall search the network host database for an entry with name *name*. gethostbyname2\_r() is a reentrant version of gethostbyname2(). These functions are similar to the gethostbyname() and gethostbyname\_r() functions but additionally allow the search to be restricted to a particular address family specified by *af*.

The application must provide a buffer for the gethostbyname2\_r() function to use during the lookup process. The buffer is referenced by *buf*, and is of size *buflen*. If the buffer is not of sufficient size, gethostbyname\_r() may fail and return ERANGE. If a matching entry is found in the database, gethostbyname\_r() shall copy the relevant information to the application-supplied hostent structure referenced by *result\_buf*, and return a pointer to this structure in \**result*. If no matching entry is found, \**result* shall be set to a null pointer. Additional error information shall be set in the variable referenced by *h\_errnop*.

##### Return Value

On success, the gethostbyname2\_r() function shall return zero. If the return value was ERANGE, the size of the buffer *buf*, indicated by *buflen*, was too small. If the gethostbyname2\_r() function returns returns any other value, then the variable referenced by *h\_errnop* shall be set to indicate the cause as for gethostbyname\_r().

#### gethostbyname\_r

##### Name

gethostbyname\_r — find network host database entry matching host name (DEPRECATED)

##### Synopsis

int gethostbyname\_r(const char \* restrict *name*, struct hostent \* restrict *result\_buf*, char \* restrict *buf*, size\_t *buflen*, struct hostent \*\* restrict *result*, int \* restrict *h\_errnop*);

##### Description

**Note:** The gethostbyname\_r() function is deprecated; applications should use getaddrinfo() instead.

gethostbyname\_r() is a reentrant version of gethostbyname() that searches the network host database for a host name match.

The gethostbyname\_r() function shall search the network host database for an entry with name *name*.

The application must provide a buffer for the gethostbyname\_r() to use during the lookup process. The buffer is referenced by *buf*, and is of size *buflen*. If the buffer is not of sufficient size, gethostbyname\_r() may fail and return ERANGE. If a matching entry is found in the database, gethostbyname\_r() shall copy the relevant information to the application supplied hostent structure referenced by *result\_buf*, and return a pointer to this structure in \**result*. If no matching entry is found, \**result* shall be set to a null pointer. Additional error information shall be set in the variable referenced by *h\_errnop*.

##### Return Value

On success, the gethostbyname\_r() function shall return zero. If the return value was ERANGE, the size of the buffer *buf*, indicated by *buflen*, was too small. If the gethostbyname\_r() function returns returns any other value, then the variable referenced by *h\_errnop* shall be set to indicate the cause as for gethostbyname().

#### getifaddrs

##### Name

getifaddrs, freeifaddrs — get interface addresses

##### Synopsis

#include <ifaddrs.h>

int getifaddrs(struct ifaddrs \*\* *ifap*);

void freeifaddrs(struct ifaddrs \* *ifa*);

##### Description

The getifaddrs() function creates a linked list of structures describing the network interfaces of the local system. The address of the first item is stored in memory pointed to by ifap. The data returned is dynamically allocated, and should be freed using freeifaddrs().

The list consists of structures of type ifaddrs (see Data Definitions above).

##### Return Value

On success, getifaddrs() returns zero; on error, -1 is returned, and errno is set appropriately.

##### Errors

getifaddrs() may fail and set errno for any of the errors specified for socket(), bind(), getsockname(), recvmsg(), sendto(), malloc(), or realloc().

##### See Also

bind(), getsockname(), socket().

#### getloadavg

##### Name

getloadavg — get system load averages

##### Synopsis

#include <stdlib.h>

int getloadavg(double *loadavg[]*, int *nelem*);

##### Description

getloadavg() returns the number of processes in the system run queue averaged over various periods of time. Up to *nelem* samples are retrieved and assigned to successive elements of *loadavg*[]. The system imposes a maximum of 3 samples, representing averages over the last 1, 5, and 15 minutes, respectively.

##### Return Value

If the load average could not be obtained, -1 is returned. Otherwise, the number of samples actually retrieved is returned.

#### getopt

##### Name

getopt — parse command line options

##### Synopsis

#include <unistd.h>

int getopt(int *argc*, char \* const *argv[]*, const char \* *optstring*);

extern char \*optarg;

extern int optind, opterr, optopt;

##### Description

The getopt() function shall parse command line arguments as described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), with the following exceptions, where LSB and POSIX specifications vary. LSB systems shall implement the modified behaviors described below.

##### Argument Ordering

The getopt() function can process command line arguments referenced by *argv* in one of three ways:

PERMUTE

  the order of arguments in *argv* is altered so that all options (and their arguments) are moved in front of all of the operands. This is the default behavior.

**Note:** This behavior has undefined results if *argv* is not modifiable. This is to support historic behavior predating the use of const and [ISO C (1999)](#ID_STD_46_ISOC99). The function prototype was aligned with [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) despite the fact that it modifies *argv*, and the library maintainers are unwilling to change this.

REQUIRE\_ORDER

  The arguments in *argv* are processed in exactly the order given, and option processing stops when the first non-option argument is reached, or when the element of argv is "--". This ordering can be enforced either by setting the environment variable POSIXLY\_CORRECT, or by setting the first character of *optstring* to '+'.

RETURN\_IN\_ORDER

  The order of arguments is not altered, and all arguments are processed. Non-option arguments (operands) are handled as if they were the argument to an option with the value 1 ('\001'). This ordering is selected by setting the first character of *optstring* to '-';

##### Option Characteristics

*LSB* specifies that:

• an element of *argv* that starts with "-" (and is not exactly "-" or "--") is an option element.

• characters of an option element, aside from the initial "-", are option characters.

*POSIX* specifies that:

• applications using getopt() shall obey the following syntax guidelines:

• option name is a single alphanumeric character from the portable character set

• option is preceded by the '-' delimiter character

• options without option-arguments should be accepted when grouped behind one '-' delimiter

• each option and option-argument is a separate argument

• option-arguments are not optional

• all options should precede operands on the command line

• the argument "--" is accepted as a delimiter indicating the end of options and the consideration of subsequent arguments, if any, as operands

• historical implementations of getopt() support other characters as options as an allowed extension, but applications that use extensions are not maximally portable.

• support for multi-byte option characters is only possible when such characters can be represented as type int.

• applications that call any utility with a first operand starting with '-' should usually specify "--" to mark the end of the options. Standard utilities that do not support this guideline indicate that fact in the OPTIONS section of the utility description.

##### Extensions

*LSB* specifies that:

• if a character is followed by two colons, the option takes an optional argument; if there is text in the current *argv* element, it is returned in *optarg*, otherwise *optarg* is set to 0.

• if *optstring* contains W followed by a semi-colon (;), then -W foo is treated as the long option --foo.

**Note:** See getopt\_long() for a description of long options.

• The first character of *optstring* shall modify the behavior of getopt() as follows:

• if the first character is '+', then REQUIRE\_ORDER processing shall be in effect (see above)

• if the first character is '-', then RETURN\_IN\_ORDER processing shall be in effect (see above)

• if the first character is ':', then getopt() shall return ':' instead of '?' to indicate a missing option argument, and shall not print any diagnostic message to stderr.

*POSIX* specifies that:

• the -W option is reserved for implementation extensions.

##### Return Values

*LSB* specifies the following additional getopt() return values:

• '\001' is returned if RETURN\_IN\_ORDER argument ordering is in effect, and the next argument is an operand, not an option. The argument is available in optarg.

Any other return value has the same meaning as for *POSIX*.

*POSIX* specifies the following getopt() return values:

• the next option character is returned, if found successfully.

• ':' is returned if a parameter is missing for one of the options and the first character of optstring is ':'.

• '?' is returned if an unknown option character not in optstring is encountered, or if getopt() detects a missing argument and the first character of optstring is not ':'.

• -1 is returned for the end of the option list.

##### Environment Variables

*LSB* specifies that:

• if the variable POSIXLY\_CORRECT is set, option processing stops as soon as a non-option argument is encountered.

• the variable \_*[PID]*\_GNU\_nonoption\_argv\_flags\_ (where *[PID]* is the process ID for the current process), contains a space separated list of arguments that should not be treated as arguments even though they appear to be so.

**Rationale:** This was used by bash 2.0 to communicate to *GNU* libc which arguments resulted from wildcard expansion and so should not be considered as options. This behavior was removed in bash version 2.01, but the support remains in *GNU* libc.

This behavior is DEPRECATED in this version of the LSB; future revisions of this specification may not include this requirement.

#### getopt\_long

##### Name

getopt\_long — parse command line options

##### Synopsis

#define \_GNU\_SOURCE

#include <getopt.h>

int getopt\_long(int *argc*, char \* const *argv[]*, const char \* *opstring*, const struct option \* *longopts*, int \* *longindex*);

##### Description

getopt\_long() works like getopt() except that it also accepts long options, started out by two dashes. Long option names may be abbreviated if the abbreviation is unique or is an exact match for some defined option. A long option may take a parameter, of the form --arg=param or --arg param.

*longopts* is a pointer to the first element of an array of struct option declared in getopt.h as:

struct option {

const char \*name;

int has\_arg;

int \*flag;

int val;

};

The fields in this structure have the following meaning:

*name*

  The name of the long option.

*has\_arg*

  One of:

|  |
| --- |
| no\_argument (or 0) if the option does not take an argument, |
| required\_argument (or 1) if the option requires an argument, or |
| optional\_argument (or 2) if the option takes an optional argument. |

*flag*

  specifies how results are returned for a long option. If flag is NULL, then getopt\_long() shall return *val*. (For example, the calling program may set val to the equivalent short option character.) Otherwise, getopt\_long() returns 0, and *flag* shall point to a variable which shall be set to *val* if the option is found, but left unchanged if the option is not found.

*val*

  The value to return, or to load into the variable pointed to by flag.

If *longindex* is not NULL, it points to a variable which is set to the index of the long option relative to *longopts*.

##### Return Value

getopt\_long() returns the option character if a short option was found successfully, or ":" if there was a missing parameter for one of the options, or "?" for an unknown option character, or -1 for the end of the option list.

For a long option, getopt\_long() returns *val* if *flag* is NULL, and 0 otherwise. Error and -1 returns are the same as for getopt(), plus "?" for an ambiguous match or an extraneous parameter.

#### getopt\_long\_only

##### Name

getopt\_long\_only — parse command line options

##### Synopsis

#define \_GNU\_SOURCE

#include <getopt.h>

int getopt\_long\_only(int *argc*, char \* const *argv[]*, const char \* *optstring*, const struct option \* *longopts*, int \* *longindex*);

##### Description

getopt\_long\_only() is like getopt\_long(), but "-" as well as "--" can indicate a long option. If an option that starts with "-" (not "--") doesn't match a long option, but does match a short option, it is parsed as a short option instead.

**Note:** The getopt\_long\_only() function is intended only for supporting certain programs whose command line syntax was designed before the Utility Syntax Guidelines of [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) were developed. New programs should generally call getopt\_long() instead, which provides the --option syntax for long options, which is preferred by GNU and consistent with [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

##### Return Value

getopt\_long\_only() returns the option character if the option was found successfully, or ":" if there was a missing parameter for one of the options, or "?" for an unknown option character, or -1 for the end of the option list.

getopt\_long\_only() also returns the option character when a short option is recognized. For a long option, they return val if flag is NULL, and 0 otherwise. Error and -1 returns are the same as for getopt(), plus "?" for an ambiguous match or an extraneous parameter.

#### getpagesize

##### Name

getpagesize — get memory page size (DEPRECATED)

##### Synopsis

#include <unistd.h>

int getpagesize (void );

##### Description

The function getpagesize() returns the number of bytes in a meory page.

**Note:** The getpagesize() function is deprecated. Portable applications should use sysconf(\_SC\_PAGE\_SIZE) instead.

##### Return Value

The getpagesize() function returns the current page size.

##### Errors

No errors are defined.

#### getprotobyname\_r

##### Name

getprotobyname\_r — retrieve information from the network protocol database by protocol name, reentrantly

##### Synopsis

#include <netdb.h>

int getprotobyname\_r(const char \* *name*, struct protoent \* *result\_buf*, char \* *buf*, size\_t *buflen*, struct protoent \* \* *result*);

##### Description

The getprotobyname\_r() function is a reentrant version of the getprotobyname() function.

The getprotobyname\_r() function shall search the network protocol database for an entry with the name *name*.

If a matching entry is found in the database, this function shall copy the relevant information to the application-supplied protoent structure referenced by *result\_buf*, and return a pointer to this structure in \**result*. If no matching entry is found, \**result* shall be set to a null pointer.

The array *buf* shall contain the string fields referenced by the protoent structure that was returned. The parameter *buflen* shall specify the array's size. 1024 bytes should be enough for most uses.

##### Return Value

On success, the getprotobyname\_r() function shall return 0. If the return value was ERANGE, the size of the buffer *buf*, indicated by *buflen*, was too small.

#### getprotobynumber\_r

##### Name

getprotobynumber\_r — retrieve information from the network protocol database by protocol number, reentrantly

##### Synopsis

#include <netdb.h>

int getprotobynumber\_r(int *proto*, struct protoent \* *result\_buf*, char \* *buf*, size\_t *buflen*, struct protoent \* \* *result*);

##### Description

The getprotobynumber\_r() function is a reentrant version of the getprotobynumber() function.

The getprotobynumber\_r() function shall search the network protocol database for an entry with protocol number *proto*.

If a matching entry is found in the database, this function shall copy the relevant information to the application-supplied protoent structure referenced by *result\_buf*, and return a pointer to this structure in \**result*. If no matching entry is found, \**result* shall be set to a null pointer.

The array *buf* shall contain the string fields referenced by the protoent structure that was returned. The parameter *buflen* shall specify the array's size. 1024 bytes should be enough for most uses.

##### Return Value

On success, the getprotobynumber\_r() function shall return 0. If the return value was ERANGE, the size of the buffer *buf*, indicated by *buflen*, was too small.

#### getprotoent\_r

##### Name

getprotoent\_r — read the next entry of the protocol database, reentrantly

##### Synopsis

#include <netdb.h>

int getprotoent\_r(struct protoent \* *result\_buf*, char \* *buf*, size\_t *buflen*, struct protoent \* \* *result*);

##### Description

The getprotoent\_r() function is a reentrant version of the getprotoent() function.

The getprotoent\_r() function shall search the network protocol database for the next entry.

If the next entry is found in the database, this function shall copy the relevant information to the application-supplied protoent structure referenced by *result\_buf*, and return a pointer to this structure in \**result*. If no next entry is found, \**result* shall be set to a null pointer.

The array *buf* shall contain the string fields referenced by the protoent structure that was returned. The parameter *buflen* shall specify the array's size. 1024 bytes should be enough for most uses.

##### Return Value

On success, the getprotoent\_r() function shall return zero.

If the return value was ENOENT, there were no more entries in the database.

If the return value was ERANGE, the size of the buffer *buf*, indicated by *buflen*, was too small.

#### getpwent\_r

##### Name

getpwent\_r — reentrantly get entry in passwd file

##### Synopsis

#include <pwd.h>

int getpwent\_r(struct passwd \* *pwbuf*, char \* *buf*, size\_t *buflen*, struct passwd \* \* *pwbufp*);

##### Description

The reentrant interface getpwent\_r() shall function in the same way as the interface getpwent(), except that getpwent\_r() shall return the user name, user password, GECOS field, home directory, and shell program in buffers provided by the caller, rather than as a pointer to static storage.

The parameter *pwbuf* contains the struct passwd that was read from the stream, if any.

The parameter *buf* contains additional strings, if any.

The parameter *buflen* specifies the size of *buf*.

The parameter *\*pwbufp* returns a pointer to the struct passwd in *\*pwbuf*.

##### Return Value

On success, getpwent\_r() shall return 0, and *\*pwbufp* shall contain a pointer to the result.

On failure, *\*pwbufp* shall contain NULL, and getpwent\_r() shall return an error as follows.

##### Errors

ENOENT

  No more password entries.

ERANGE

  Not enough buffer space. Specify a larger buffer and try again.

#### getrlimit, setrlimit

##### Name

getrlimit, setrlimit — get resource consumption limits

##### Synopsis

#include <sys/resource.h>

int getrlimit(\_\_rlimit\_resource\_t *\_\_resource*, struct rlimit \* *\_\_rlimits*);

int setrlimit(\_\_rlimit\_resource\_t *\_\_resource*, const struct rlimit \* *\_\_rlimits*);

##### Description

getrlimit() and setrlimit() are as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but with differences as listed below.

#### Extra Resources

These additional resources extend the list in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

RLIMIT\_NPROC

  The maximum number of processes (or, more precisely on Linux, threads) that can be created for the real user ID of the calling process. Upon encountering this limit, fork() shall fail with the error EAGAIN.

RLIMIT\_MEMLOCK

  The maximum number of bytes of memory that may be locked into RAM. In effect this limit is rounded down to the nearest multiple of the system page size. This limit affects mlock() and mlockall(), the mmap() MAP\_LOCKED operation and the shmctl() SHM\_LOCK operation. The shmctl() SHM\_LOCK locks are accounted for separately from the per-process memory locks established by mlock(), mlockall(), and mmap() MAP\_LOCKED. In the former case, the limit sets a maximum on the total bytes in shared memory segments (see shmget()) that may be locked by the real user ID of the calling process. A process can lock bytes up to this limit in each of these two categories.

RLIMIT\_LOCKS

  A limit on the combined number of flock() locks and fcntl() leases that this process may establish. This limit is obsolete and should not be used; support depends heavily on the version of the operating system kernel.

RLIMIT\_RSS

  Specifies the limit (in pages) of the process's resident set. This limit is obsolete and should not be used; support depends heavily on the version of the operating system kernel. It affects only calls to madvise() specifying MADV\_WILLNEED.

RLIMIT\_SIGPENDING

  Specifies the limit on the number of signals that may be queued for the real user ID of the calling process. Both standard and real-time signals are counted for the purpose of checking this limit. However, the limit is enforced only for sigqueue(); it is always possible to use kill() to queue one instance of any of the signals that are not already queued to the process.

RLIMIT\_MSGQUEUE

  Specifies the limit on the number of bytes that can be allocated for POSIX message queues for the real user ID of the calling process. This limit is enforced for mq\_open(). Each message queue that the user creates counts (until it is removed) against this limit according to the formula:

bytes = attr.mq\_maxmsg \* sizeof(struct msg\_msg \*) +

attr.mq\_maxmsg \* attr.mq\_msgsize

where attr is the mq\_attr structure specified as the fourth argument to mq\_open(3).

The first addend in the formula, which includes sizeof(struct msg\_msg \*) (4 bytes on Linux/i386), ensures that the user cannot create an unlimited number of zero-length messages (such messages nevertheless each consume some system memory for bookkeeping overhead).

RLIMIT\_NICE

  Specifies a ceiling to which the process's nice value can be raised using setpriority() or nice(). The actual ceiling for the nice value is calculated as 20 minus the value of rlim\_cur.

RLIMIT\_RTPRIO

  Specifies a ceiling on the real-time priority that may be set for this process using sched\_setscheduler(2) and sched\_setparam(2).

RLIMIT\_RTTIME

  Specifies a limit (in microseconds) on the amount of CPU time that a process scheduled under a real-time scheduling policy may consume without making a blocking system call. For the purpose of this limit, each time a process makes a blocking system call, the count of its consumed CPU time is reset to zero. The CPU time count is not reset if the process continues trying to use the CPU but is preempted, its time slice expires, or it calls sched\_yield().

Upon reaching the soft limit, the process is sent a SIGXCPU signal. If the process catches or ignores this signal and continues consuming CPU time, then SIGXCPU will be generated once each second until the hard limit is reached, at which point the process is sent a SIGKILL signal.

The intended use of this limit is to stop a runaway real-time process from locking up the system.

##### Extra Errors

These additional error codes extend the list in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

EFAULT

  A pointer argument points to a location outside the accessible address space.

#### getservbyname\_r

##### Name

getservbyname\_r — retrieve information from the network services database by service name, reentrantly

##### Synopsis

#include <netdb.h>

int getservbyname\_r(const char \* *name*, const char \* *proto*, struct servent \* *result\_buf*, char \* *buf*, size\_t *buflen*, struct servent \* \* *result*);

##### Description

The getservbyname\_r() function is a reentrant version of the getservbyname() function.

The getservbyname\_r() function shall search the network services database for an entry with the name *name*. The *proto* parameter shall restrict the search to entries with the specified protocol. If *proto* is NULL, getservbyname\_r() may return entries with any protocol.

If a matching entry is found in the database, this function shall copy the relevant information to the application-supplied servent structure referenced by *result\_buf*, and return a pointer to this structure in \**result*. If no matching entry is found, \**result* shall be set to a null pointer.

The array *buf* shall contain the string fields referenced by the servent structure that was returned. The parameter *buflen* shall specify the array's size. 1024 bytes should be enough for most uses.

##### Return Value

On success, the getservbyname\_r() function shall return zero. If the return value was ERANGE, the size of the buffer *buf*, indicated by *buflen*, was too small.

#### getservbyport\_r

##### Name

getservbyport\_r — retrieve information from the network services database by service port, reentrantly

##### Synopsis

#include <netdb.h>

int getservbyport\_r(int *port*, const char \* *proto*, struct servent \* *result\_buf*, char \* *buf*, size\_t *buflen*, struct servent \* \* *result*);

##### Description

The getservbyport\_r() function is a reentrant version of the getservbyport() function.

The getservbyport\_r() function shall search the network services database for an entry with the port *port*. The *proto* parameter shall restrict the search to entries with the specified protocol. If *proto* is NULL, getservbyport\_r() may return entries with any protocol.

If a matching entry is found in the database, this function shall copy the relevant information to the application-supplied servent structure referenced by *result\_buf*, and return a pointer to this structure in \**result*. If no matching entry is found, \**result* shall be set to a null pointer.

The array *buf* shall contain the string fields referenced by the servent structure that was returned. The parameter *buflen* shall specify the array's size. 1024 bytes should be enough for most uses.

##### Return Value

On success, the getservbyport\_r() function shall return zero. If the return value was ERANGE, the size of the buffer *buf*, indicated by *buflen*, was too small.

#### getservent\_r

##### Name

getservent\_r — read the next entry of the network services database, reentrantly

##### Synopsis

#include <netdb.h>

int getservent\_r(struct servent \* *result\_buf*, char \* *buf*, size\_t *buflen*, struct servent \* \* *result*);

##### Description

The getservent\_r() function is a reentrant version of the getservent() function.

The getservent\_r() function shall search the network services database for the next entry.

If the next entry is found in the database, this function shall copy the relevant information to the application-supplied servent structure referenced by *result\_buf*, and return a pointer to this structure in \**result*. If no next entry is found, \**result* shall be set to a null pointer.

The array *buf* shall contain the string fields referenced by the servent structure that was returned. The parameter *buflen* shall specify the array's size. 1024 bytes should be enough for most uses.

##### Return Value

On success, the getservent\_r() function shall return 0.

If the return value was ENOENT, there were no more entries in the database.

If the return value was ERANGE, the size of the buffer *buf*, indicated by *buflen*, was too small.

#### getsockopt

##### Name

getsockopt — get socket options

##### Synopsis

#include <sys/socket.h>

#include <netinet/ip.h>

int getsockopt(int *socket*, int *level*, int *option\_name*, void \* restrict *option\_value*, socklen\_t \* restrict *option\_len*);

##### Description

The getsockopt() function shall behave as specified in [*POSIX 1003.1-2008 (ISO/IEC 9945-2009)*](#ID_STD_46_SUSV4), with the following extensions.

##### IP Protocol Level Options

If the *level* parameter is IPPROTO\_IP, the following values shall be supported for *option\_name* (see [RFC 791:Internet Protocol](#ID_STD_46_RFC791) for further details):

IP\_OPTIONS

  Get the Internet Protocol options sent with every packet from this socket. The *option\_value* shall point to a memory buffer in which the options shall be placed; on entry *option\_len* shall point to an integer value indicating the maximum size of the memory buffer, in bytes. On successful return, the value referenced by *option\_len* shall be updated to the size of data copied to the buffer. For IPv4, the maximum length of options is 40 bytes.

IP\_TTL

  Get the current unicast Internet Protocol Time To Live value used when sending packets with this socket. The *option\_value* shall point to a buffer large enough to hold the time to live value (at least 1 byte), and *option\_len* shall point to an integer value holding the maximum size of that buffer. On successful return, the value referenced by *option\_len* shall be updated to contain the number of bytes copied into the buffer, which shall be no larger than the initial value, and *option\_value* shall point to an integer containing the time to live value.

IP\_TOS

  Get the Internet Protocol type of service indicator used when sending packets with this socket. The *option\_value* shall point to a buffer large enough to hold the type of service indicator (at least 1 byte), and *option\_len* shall point to an integer value holding the maximum size of that buffer. On successful return, the value referenced by *option\_len* shall be updated to contain the number of bytes copied into the buffer, which shall be no larger than the initial value, and *option\_value* shall point to an integer containing the time to live value.

#### gettext

##### Name

gettext — search message catalogs for a string

##### Synopsis

#include <libintl.h>

char \* gettext(const char \* *msgid*);

##### Description

The gettext() function shall search the currently selected message catalogs for a string identified by the string *msgid*. If a string is located, that string shall be returned.

The gettext() function is equivalent to dcgettext(NULL, msgid, LC\_MESSAGES).

##### Return Value

If a string is found in the currently selected message catalogs for *msgid*, then a pointer to that string shall be returned. Otherwise, a pointer to *msgid* shall be returned.

Applications shall not modify the string returned by gettext().

##### Errors

None.

The gettext() function shall not modify errno.

##### See Also

dgettext, ngettext, dngettext, dcgettext, dcngettext, textdomain, bindtextdomain, bind\_textdomain\_codeset

#### getutent

##### Name

getutent — access user accounting database entries

##### Synopsis

#include <utmp.h>

struct utmp \*getutent(void);

##### Description

The getutent() function shall read the next entry from the user accounting database.

##### Return Value

Upon successful completion, getutent() shall return a pointer to a utmp structure containing a copy of the requested entry in the user accounting database. Otherwise, a null pointer shall be returned. The return value may point to a static area which is overwritten by a subsequent call to getutent().

##### Errors

None defined.

#### getutent\_r

##### Name

getutent\_r — access user accounting database entries

##### Synopsis

int getutent\_r(struct utmp \* *buffer*, struct utmp \*\* *result*);

##### Description

The getutent\_r() function is a reentrant version of the getutent() function. On entry, *buffer* should point to a user supplied buffer to which the next entry in the database will be copied, and *result* should point to a location where the result will be stored.

##### Return Value

On success, getutent\_r() shall return 0 and set the location referenced by *result* to a pointer to *buffer*. Otherwise, getutent\_r() shall return -1 and set the location referenced by *result* to NULL.

#### getwc\_unlocked

##### Name

getwc\_unlocked — non-thread-safe getwc

##### Description

getwc\_unlocked() is the same as getwc(), except that it need not be thread-safe. That is, it may only be invoked in the ways which are legal for getc\_unlocked().

#### getwchar\_unlocked

##### Name

getwchar\_unlocked — non-thread-safe getwchar

##### Description

getwchar\_unlocked() is the same as getwchar(), except that it need not be thread-safe. That is, it may only be invoked in the ways which are legal for getc\_unlocked().

#### glob64

##### Name

glob64 — find pathnames matching a pattern (Large File Support)

##### Synopsis

#include <glob.h>

int glob64(const char \* *pattern*, int *flags*, int (\**errfunc*) (const char \*, int), glob64\_t \* *pglob*);

##### Description

glob64() is a large-file version of the glob() function defined in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4). It shall search for pathnames matching *pattern* according to the rules used by the shell, /bin/sh. No tilde expansion or parameter substitution is done; see wordexp().

The results of a glob64() call are stored in the structure pointed to by *pglob*, which is a glob64\_t declared in glob.h with the following members:

typedef struct

{

size\_t *gl\_pathc*;

char \*\**gl\_pathv*;

size\_t *gl\_offs*;

int *gl\_flags*;

void (\**gl\_closedir*) (void \*);

struct dirent64 \*(\**gl\_readdir64*) (void \*);

void \*(\**gl\_opendir*) (const char \*);

int (\**gl\_lstat*) (const char \*, struct stat \*);

int (\**gl\_stat*) (const char \*, struct stat \*);

}

glob64\_t;

Structure members with the same name as corresponding members of a glob\_t as defined in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) shall have the same purpose.

Other members are defined as follows:

*gl\_flags*

  reserved for internal use

*gl\_closedir*

  pointer to a function capable of closing a directory opened by *gl\_opendir*

*gl\_readdir64*

  pointer to a function capable of reading entries in a large directory

*gl\_opendir*

  pointer to a function capable of opening a large directory

*gl\_stat*

  pointer to a function capable of returning file status for a large file

*gl\_lstat*

  pointer to a function capable of returning file status information for a large file or symbolic link

A large file or large directory is one with a size which cannot be represented by a variable of type off\_t.

##### Return Value

On success, 0 is returned. Other possible returns are:

GLOB\_NOSPACE

  out of memory

GLOB\_ABORTED

  read error

GLOB\_NOMATCH

  no match found

#### globfree64

##### Name

globfree64 — free memory from glob64() (Large File Support)

##### Synopsis

#include <glob.h>

void globfree64(glob64\_t \* *pglob*);

##### Description

globfree64() frees the dynamically allocated storage from an earlier call to glob64().

globfree64() is a large-file version of the globfree() function defined in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

#### gnu\_get\_libc\_version, gnu\_get\_libc\_release

##### Name

gnu\_get\_libc\_version, gnu\_get\_libc\_release — get glibc-specific version and release

##### Synopsis

#include <gnu/libc-version.h>

const char \* gnu\_get\_libc\_version(void);

const char \* gnu\_get\_libc\_release(void);

##### Description

gnu\_get\_libc\_version() returns a string that identifies the version of the C library running the program making the call.

gnu\_get\_libc\_release() returns a string indicates the release status of the C library running the program making the call. This will be a string such as "stable".

##### Return Value

The functions return strings. The contents of these strings are unspecified.

##### Errors

No errors are defined.

##### Notes

These functions are specific to GNU libc (glibc). This specification does not require the implementation of libc to be glibc, although it requires these functions.

The string returned by gnu\_get\_libc\_version() will be a dotted version string, which may have meaning to developers otherwise familiar with glibc. These functions have been requested to aid in portability of software which also runs in non-LSB contexts, but decisions based on the return value should be tempered by an understanding of what the behavioral requirements of this specification are. That is, it may or may not be useful to discover that a running system, for example, has version "2.10.1" if that implies different behavior than described by this specification.

#### hcreate\_r

##### Name

hcreate\_r — allocate space for a hash search table, reentrantly

##### Synopsis

#include <search.h>

int hcreate\_r(size\_t *nel*, struct hsearch\_data \* *htab*);

##### Description

The hcreate\_r() function is a reentrant version of the hcreate() function.

hcreate\_r() shall initialize the object referenced by *htab* with a hash table containing at least *nel* elements. Unlike its non-reentrant equivalent, hcreate(), the hcreate\_r() function may work with more than one hash table.

The memory for the *htab* object may be dynamically allocated. It must be initialized with 0 before hcreate\_r() is called.

##### Return Value

On success, hcreate\_r() shall return a non-zero value.

On failure, hcreate\_r() shall return 0. This usually happens because not enough memory was available.

#### hdestroy\_r

##### Name

hdestroy\_r — dispose of a hash search table, reentrantly

##### Synopsis

#include <search.h>

void hdestroy\_r(struct hsearch\_data \* *htab*);

##### Description

The hdestroy\_r() function is a reentrant version of the hdestroy() function.

hdestroy\_r() frees the resources allocated by hcreate\_r() for the object *htab*.

#### hsearch\_r

##### Name

hsearch\_r — search a hash table, reentrantly

##### Synopsis

#include <search.h>

int hsearch\_r(ENTRY *item*, ACTION *action*, ENTRY \* \* *retval*, struct hsearch\_data \* *htab*);

##### Description

The hsearch\_r() is a reentrant version of the hsearch() function, but instead of operating on a single global hash table, hsearch\_r() operates on the table described by the object that *htab* references. This object can be initialized with the function hcreate\_r().

Unlike the hsearch() function, hsearch\_r() returns a pointer to the found entry in the variable referred to by *retval*, rather than directly.

##### Return Value

On success, hsearch\_r() shall return a non-zero value.

On failure, hsearch\_r() shall return 0 and set errno to an appropriate value.

##### Errors

ENOMEM

*action* was set to ENTER, but the table was full.

ESRCH

*action* was set to FIND, but no matching element was found in the table.

#### inet\_aton

##### Name

inet\_aton — Internet address manipulation routine

##### Synopsis

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

int anet\_iton(const char \* *cp*, struct in\_addr \* *inp*);

##### Description

inet\_aton() converts the Internet host address *cp* from the standard IPv4 numbers-and-dots notation into binary data and stores it in the structure that *inp* points to.

inet\_aton() returns a nonzero value if the address is valid, 0 if not.

**Note:** Note that on some LSB architectures, the host byte order is Least Significant Byte first, whereas the network byte order, as used on the Internet, is Most Significant Byte first.

#### initgroups

##### Name

initgroups — initialize the supplementary group access list

##### Synopsis

#include <grp.h>

#include <sys/types.h>

int initgroups(const char \* *user*, gid\_t *group*);

##### Description

If the process has appropriate privilege, the initgroups() function shall initialize the Supplementary Group IDs for the current process by reading the group database and using all groups of which *user* is a member. The additional group *group* is also added to the list.

##### Return Value

On success, 0 is returned. On error, -1 is returned and the global variable errno is set appropriately.

##### Errors

EPERM

  The calling process does not have sufficient privileges.

ENOMEM

  Insufficient memory to allocate group information structure.

##### See Also

setgroups()

#### initstate\_r

##### Name

initstate\_r — reentrantly initialize a state array for random number generator functions

##### Synopsis

#include <stdlib.h>

int initstate\_r(unsigned int *seed*, char \* *statebuf*, size\_t *statelen*, struct random\_data \* *buffer*);

##### Description

The interface initstate\_r() shall function in the same way as the interface initstate(), except that initstate\_r() shall use the data in *buffer* instead of the global random number generator state.

#### inotify\_add\_watch

##### Name

inotify\_add\_watch — add a watch to a watch list

##### Synopsis

#include <sys/inotify.h>

int inotify\_add\_watch(int *fd*, const char \* *path*, uint32\_t *mask*);

##### Description

inotify\_add\_watch() shall add a watch to, or modify an existing watch on, the watch list of the inotify instance specified by the file descriptor *fd*, for the file specified by *path*, to monitor the events specified by the bitmask *mask*. The caller must have read access to the file.

##### Return Value

On success, inotify\_add\_watch() shall return the unique, non-negative watch descriptor associated with the file *path* and the inotify instance specified by the file descriptor *fd*.

If *path* was already on the watch list, then inotify\_add\_watch() shall return the existing watch descriptor.

If *path* was not already on the watch list, then inotify\_add\_watch() shall allocate a new watch descriptor.

inotify\_add\_watch() shall not work recursively. Monitoring subdirectories of *path* shall require adding watches to them.

On failure, inotify\_add\_watch() shall return -1 and set errno to an appropriate value.

##### Errors

EACCESS

  The caller does not have read access to *path*.

EBADF

  The file descriptor *fd* is invalid.

EFAULT

*path* is outside of the address space accessible by the process.

EINVAL

*mask* contains no legal events, or *fd* is not a valid inotify file descriptor.

ENOMEM

  There is not enough kernel memory available.

ENOSPC

  The maximum number of watches has been created for this user, or the kernel cannot allocate a resource.

##### Application Usage

###### Reading

The function read() can be used to determine which inotify events have occurred. A blocking file descriptor will make read() block until at least one event has occurred.

If successful, read() will return at least one of the following inotify\_event structures in a buffer:

struct inotify\_event {

int wd;

uint32\_t mask;

uint32\_t cookie;

uint32\_t len;

char path[];

};

wd is a watch descriptor that specifies the watch associated with the event. It is obtained from a previous invocation of inotify\_add\_watch().

mask is a bit mask describing inotify events. See the section on masks below.

cookie is an integer associating related inotify events. The integer value is unique, and currently only enables the application to associate IN\_MOVE\_FROM and IN\_MOVE\_TO rename events.

len is a count of the bytes in path, including null bytes. This means that the total length of an inotify\_event structure is

sizeof(inotify\_event)+len

path is only returned when an event occurs for a file within a watched directory. This string is null-terminated, and it may contain more null bytes so that future reads will be aligned properly on an address boundary.

In kernels before 2.6.21, read() returns 0 when the buffer given to it is too small to return data about the next event. In subsequent kernels, it fails with the error EINVAL.

For a given file descriptor, the inotify events are returned in an ordered queue. Events on a file descriptor will always be returned in the correct order of occurrence. If two or more inotify events for a given file descriptor have identical values for all fields, then only one inotify\_event will be returned to represent all of them.

The number of bytes that can be read from an inotify file descriptor can be determined by making a FIONREAD ioctl() call.

###### Masks

The *mask* argument of inotify\_add\_watch() and the mask field of the inotify\_event structure are bit masks that specify inotify events. The bits in the list below can be set in the *mask* argument of inotify\_add\_watch() and returned in the mask field of inotify\_event.

IN\_ACCESS

  File was read.

IN\_ALL\_EVENTS

  Bit mask of all events in this list.

IN\_ATTRIB

  File's metadata changed (including timestamps and permissions).

IN\_CLOSE

  Same as

IN\_CLOSE\_WRITE | IN\_CLOSE\_NOWRITE

IN\_CLOSE\_WRITE

  File that was opened for writing was closed.

IN\_CLOSE\_NOWRITE

  File that was not opened for writing was closed.

IN\_CREATE

  File or directory was created in a watched directory.

IN\_DELETE

  File or directory was deleted in a watched directory.

IN\_DELETE\_SELF

  Watched file or directory was deleted.

IN\_MODIFY

  File was changed.

IN\_MOVE

  Same as

IN\_MOVED\_FROM | IN\_MOVED\_TO

IN\_MOVE\_SELF

Watched file or directory was moved

IN\_MOVED\_FROM

  File was moved out of watched directory.

IN\_MOVED\_TO

  File was moved into watched directory.

IN\_OPEN

  File was opened.

All of the events above, except for IN\_DELETE\_SELF and IN\_MOVE\_SELF, cause the name field of the inotify\_event structure to contain the name of the file or directory being monitored.

The following bit is valid for inotify\_add\_watch() only.

IN\_ONESHOT

  Monitor path for an event, and then remove it from the watch list.

The following bits are valid for the inotify\_event structure only.

IN\_IGNORED

  Watch was removed, either explicitly (via inotify\_rm\_watch()) or implicitly (file deletion or file system unmounting).

IN\_ISDIR

  Object being watched is a directory.

IN\_Q\_OVERFLOW

  The event queue overflowed (*wd* is set to -1).

IN\_UNMOUNT

  File system of object being watched was unmounted.

###### Notes

It is possible to monitor file descriptors with the functions epoll(), poll(), and select().

When all of the file descriptors that point to an inotify instance have been closed, the instance and its associated resources and watches are freed by the kernel.

##### See Also

inotify\_init(), inotify\_rm\_watch()

#### inotify\_init

##### Name

inotify\_init — instantiate inotify

##### Synopsis

#include <sys/inotify.h>

int inotify\_init(void);

##### Description

inotify\_init() shall create one instance of inotify.

##### Return Value

On success, inotify\_init() shall return a file descriptor pointing to the new inotify instance.

On failure, inotify\_init() shall return -1 and set errno to an appropriate value.

##### Errors

EMFILE

  The maximum number of inotify instances has been created for this user.

ENFILE

  The maximum number of file descriptors has been created on the system.

ENOMEM

  There is not enough kernel memory available.

##### See Also

inotify\_add\_watch(), inotify\_rm\_watch()

#### inotify\_rm\_watch

##### Name

inotify\_rm\_watch — remove a watch from an inotify watch list

##### Synopsis

#include <sys/inotify.h>

int inotify\_rm\_watch(int *fd*, int *wd*);

##### Description

inotify\_rm\_watch() shall remove the watch associated with the watch descriptor *wd* from the watch list of the inotify instance associated with the file descriptor *fd*.

If a watch is removed, its watch descriptor shall generate the IN\_IGNORED event.

##### Return Value

On success, inotify\_rm\_watch() shall return 0.

On failure, inotify\_rm\_watch() shall return -1 and set errno to an appropriate value.

##### Errors

EBADF

  The file descriptor *fd* is invalid.

EINVAL

*wd* is invalid, or *fd* is not a valid inotify file descriptor.

##### See Also

inotify\_add\_watch(), inotify\_init()

#### ioctl

##### Name

ioctl — control device

##### Synopsis

#include <sys/ioctl.h>

int ioctl (int *fildes* , int *request* , ...);

##### Description

The ioctl() function shall manipulate the underlying device parameters of special files. *fildes* shall be an open file descriptor referring to a special file. The ioctl() function shall take three parameters; the type and value of the third parameter is dependent on the device and *request*.

Conforming LSB applications shall not call ioctl() except in situations explicitly stated in this specification.

##### Return Value

On success, 0 is returned. An ioctl() may use the return value as an output parameter and return a non-negative value on success. On error, -1 is returned and the global variable errno is set appropriately.

##### Errors

EBADF

*fildes* is not a valid descriptor.

EFAULT

  The third parameter references an inaccessible memory area.

ENOTTY

*fildes* is not associated with a character special device.

ENOTTY

  The specified request does not apply to the kind of object that *fildes* references.

EINVAL

*request* or the third parameter is not valid.

##### Relationship to POSIX (Informative)

It should be noted that [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) contains an interface named ioctl(). The LSB only defines behavior when *fildes* refers to a socket (see [sockio](#ID_BASELIB_45_SOCKIO_45_2)) or terminal device (see [ttyio](#ID_BASELIB_45_TTYIO_45_2)), while [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) only defines behavior when *fildes* refers to a STREAMS device. An implementation may support both behaviors; the LSB does not require any STREAMS support.

#### sockio

##### Name

sockio — socket ioctl commands

##### Synopsis

#include <sys/ioctl.h>

#include <sys/socket.h>

#include <net/if.h>

#include <netinet/in.h>

int ioctl(int *sockfd*, int *request*, void \* *argp*);

##### Description

Socket ioctl() commands are a subset of the ioctl() calls, which can perform a variety of functions on sockets. *sockfd* shall be an open file descriptor referring to a socket (see the socket() or accept() functions).

Socket ioctl() commands apply to the underlying network interfaces, and affect the entire system, not just the file descriptor used to issue the ioctl().

The following values for *request* are accepted:

SIOCGIFCONF (Deprecated)

  Get the interface configuration list for the system.

**Note:** The SIOCGIFCONF interface is superceded by the if\_nameindex() family of functions (see [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4)). A future version of this specification may withdraw this value for *request*.

*argp* shall point to a ifconf structure, as described in <net/if.h>. Before calling, the caller shall set the *ifc\_ifcu.ifcu\_req* field to point to an array of ifreq structures, and set *ifc\_len* to the size in bytes of this allocated array. Upon return, *ifc\_len* will contain the size in bytes of the array which was actually used. If it is the same as the length upon calling, the caller should assume that the array was too small and try again with a larger array.

On success, SIOCGIFCONF shall return a nonnegative value.

**Rationale:** Historical UNIX systems disagree on the meaning of the return value.

SIOCGIFFLAGS

  Get the interface flags for the indicated interface. *argp* shall point to a ifreq structure. Before calling, the caller should fill in the *ifr\_name* field with the interface name, and upon return, the *ifr\_ifru.ifru\_flags* field is set with the interface flags.

SIOCGIFADDR

  Get the interface address for the given interface. *argp* shall point to a ifreq structure. Before calling, the caller should fill in the *ifr\_name* field with the interface name, and upon return, the *ifr\_ifru.ifru\_addr* field is set with the interface address.

SIOCGIFBRDADDR

  Get the interface broadcast address for the given interface. *argp* shall point to a ifreq structure. Before calling, the caller should fill in the *ifr\_name* field with the interface name, and upon return, the *ifr\_ifru.ifru\_broadcast* field is set with the interface broadcast address.

SIOCGIFDSTADDR

  Get the point-to-point address for the given interface. *argp* shall point to a ifreq structure. Before calling, the caller should fill in the *ifr\_name* field with the interface name, and upon return, the *ifr\_dstaddr* field is set with the point-to-point address.

SIOCGIFNAME

  Get the name of an interface. *argp* shall point to a ifreq structure. Before calling, the caller should fill in the *ifr\_ifindex* field with the number (index) of the interface, and upon return, the *ifr\_name* field is set with the interface name.

SIOCGIFNETMASK

  Get the network mask for the given interface. *argp* shall point to a ifreq structure. Before calling, the caller should fill in the *ifr\_name* field with the interface name, and upon return, the *ifr\_ifru.ifru\_netmask* field is set with the network mask.

SIOCGIFMTU

  Get the Maximum Transmission Unit (MTU) size for the given interface. *argp* shall point to a ifreq structure. Before calling, the caller should fill in the *ifr\_name* field with the interface name, and upon return, the *ifr\_ifru.ifru\_mtu* field is set with the MTU. Note: The range of valid values for MTU varies for an interface depending on the interface type.

FIONREAD

  Get the amount of queued unread data in the receive buffer. *argp* shall point to an integer where the result is to be placed.

**Note:** Some implementations may also support the use of FIONREAD on other types of file descriptor. However, the LSB only specifies its behavior for a socket related file descriptor.

##### Return Value

On success, if *request* is SIOCGIFCONF, a non-negative integer shall be returned. If request is not SIOCGIFCONF, on success 0 is returned. On error, -1 is returned and the global variable errno is set appropriately.

##### Errors

EBADF

*sockfd* is not a valid descriptor.

EFAULT

*argp* references an inaccessible memory area.

ENOTTY

  The specified *request* does not apply to the kind of object that the descriptor *sockfd* references.

EINVAL

  Either *request* or *argp* is invalid.

ENOTCONN

  The operation is only defined on a connected socket, but the socket wasn't connected.

#### ttyio

##### Name

ttyio — tty ioctl commands

##### Synopsis

#include <sys/ioctl.h>

#include <fcntl.h>

int ioctl(int *fd*, unsigned long *request*, int \* *argp*);

##### Description

Tty *ioctl* commands are a subset of the ioctl() calls, which can perform a variety of functions on tty devices. *fd* shall be an open file descriptor referring to a terminal device.

The following ioctl()s are provided:

TIOCGWINSZ

  Get the size attributes of the terminal or pseudo-terminal identified by *fd*. On entry, *argp* shall reference a winsize structure. On return, the structure will have *ws\_row* set to the number of rows of text (i.e. lines of text) that can be viewed on the device, and *ws\_col* set to the number of columns (i.e. text width).

**Note:** The number of columns stored in *ws\_col* assumes that the terminal device is using a mono-spaced font.

TIOCSWINSZ

  Sets the size attributes of the terminal or pseudo-terminal identified by *fd*. On entry, *argp* shall reference a winsize structure. The value of the winsize structure's element *ws\_row* shall be the number of rows of text (i.e. lines of text) that can be viewed on the device, and the element *ws\_col* shall be the number of columns (i.e. text width). Note that this call merely sets the size attributes for the kernel driver, not the window size itself, and is intended to be used to update the kernel driver when the window size is changed.

##### Return Value

On success, 0 is returned. On error, -1 is returned and the global variable errno is set appropriately.

##### Errors

EBADF

*fd* is not a valid descriptor.

EFAULT

*argp* references an inaccessible memory area.

EINVAL

*request* and *argp* are not valid.

#### jrand48\_r

##### Name

jrand48\_r — reentrantly generate pseudorandom numbers in a uniform distribution

##### Synopsis

#include <stdlib.h>

int jrand48\_r(unsigned short[3] *xsubi*, struct drand48\_data \* *buffer*, long int \* *result*);

##### Description

The interface jrand48\_r() shall function in the same way as the interface jrand48(), except that jrand48\_r() shall use the data in *buffer* instead of the global random number generator state.

Before it is used, *buffer* must be initialized, for example, by calling lcong48\_r(), seed48\_r(), or srand48\_r(), or by filling it with zeroes.

#### kill

##### Name

kill — send a signal

##### Synopsis

#include <signal.h>

int kill(pid\_t *pid*, int *sig*);

##### Description

kill() is as specified in the [*POSIX 1003.1-2008 (ISO/IEC 9945-2009)*](#ID_STD_46_SUSV4), but with differences as listed below.

#### Process ID -1 doesn't affect calling process

If *pid* is specified as -1, *sig* shall not be sent to the calling process. Other than this, the rules in the [*POSIX 1003.1-2008 (ISO/IEC 9945-2009)*](#ID_STD_46_SUSV4) apply.

**Rationale:** This was a deliberate Linus decision after an unpopular experiment in including the calling process in the 2.5.1 kernel. See "What does it mean to signal everybody?", Linux Weekly News, 20 December 2001, http://lwn.net/2001/1220/kernel.php3

#### lcong48\_r

##### Name

lcong48\_r — reentrantly generate pseudorandom numbers in a uniform distribution

##### Synopsis

#include <stdlib.h>

int lcong48\_r(unsigned short[7] *param*, struct drand48\_data \* *buffer*);

##### Description

The interface lcong48\_r() shall function in the same way as the interface lcong48(), except that lcong48\_r() shall use the data in *buffer* instead of the global random number generator state.

#### link

##### Name

link — create a link to a file

##### Synopsis

#include <unistd.h>

int link(const char \* *path1*, const char \* *path2*);

##### Description

The link() function shall behave as specified in [*POSIX 1003.1-2008 (ISO/IEC 9945-2009)*](#ID_STD_46_SUSV4), except with differences as listed below.

#### Need Not Follow Symlinks

[POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) specifies that pathname resolution shall follow symbolic links during pathname resolution unless the function is required to act on the symbolic link itself, or certain arguments direct that the function act on the symbolic link itself. The link() function in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) contains no such requirement to operate on a symbolic link. However, a conforming LSB implementation need not follow a symbolic link for the *path1* argument.

#### lrand48\_r

##### Name

lrand48\_r — reentrantly generate pseudorandom numbers in a uniform distribution

##### Synopsis

#include <stdlib.h>

int lrand48\_r(struct drand48\_data \* *buffer*, long int \* *result*);

##### Description

The interface lrand48\_r() shall function in the same way as the interface lrand48(), except that lrand48\_r() shall use the data in *buffer* instead of the global random number generator state.

Before it is used, *buffer* must be initialized, for example, by calling lcong48\_r(), seed48\_r(), or srand48\_r(), or by filling it with zeroes.

#### memmem

##### Name

memmem — locate bytes

##### Synopsis

#define \_GNU\_SOURCE

#include <string.h>

void \* memmem(const void \* *haystack*, size\_t *haystacklen*, const void \* *needle*, size\_t *needlelen*);

##### Description

memmem() finds the start of the first occurrence of the byte array referenced by *needle* of length *needlelen* in the memory area *haystack* of length *haystacklen*.

##### Return Value

If *needle* is found, memmem() returns a pointer to it. If *needlelen* is 0, *memmem* returns *haystack*. If *needle* is not found in *haystack*, memmem() returns NULL.

##### Notes

Earlier versions of the C library (prior to glibc 2.1) contained a memmem() with various problems, and application developers should treat this function with care.

#### memrchr

##### Name

memrchr — scan memory for a character

##### Synopsis

#include <string.h>

void \* memrchr(const void \* *s*, int *c*, size\_t *n*);

##### Description

The memrchr() function shall locate the last occurence of *c* (converted to an unsigned char) in the initial *n* bytes (each interpreted as an unsigned char) of the object pointed to by *s*.

##### Return Value

The memrchr() shall return a pointer to the located byte, or a null pointer if the byte does not occur in the object.

##### Errors

No errors are defined.

##### See Also

memchr()

#### mkstemp64

##### Name

mkstemp64 — create a unique temporary file (Large File Support)

##### Synopsis

#include <stdio.h>

#include <stdlib.h>

int mkstemp64(char \* *template*);

##### Description

mkstemp64() shall generates a unique temporary file name from *template*. The last six characters of template shall be XXXXXX and these are replaced with a string that makes the file name unique; the file is then created and an open file descriptor returned as described for mkstemp().

mkstemp64() is a large-file version of the mkstemp() function as defined in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4). The only difference is that the temporary file is opened with open64() instead of with open().

##### Return Value

On success, mkstemp64() returns the file descriptor of the temporary file. Otherwise mkstemp64() shall return -1 and set errno to indicate the error.

##### Errors

See mkstemp() for possible error values.

#### mrand48\_r

##### Name

mrand48\_r — reentrantly generate pseudorandom numbers in a uniform distribution

##### Synopsis

#include <stdlib.h>

int mrand48\_r(struct drand48\_data \* *buffer*, long int \* *result*);

##### Description

The interface mrand48\_r() shall function in the same way as the interface mrand48(), except that mrand48\_r() shall use the data in *buffer* instead of the global random number generator state.

Before it is used, *buffer* must be initialized, for example, by calling lcong48\_r(), seed48\_r(), or srand48\_r(), or by filling it with zeroes.

#### Mremap

##### Name

mremap — remap a virtual memory address

##### Synopsis

#include <sys/mman.h>

void \* mremap(void \* *old\_address*, size\_t *old\_size*, size\_t *new\_size*, int *flags*);

##### Description

The mremap() function expands (or shrinks) an existing memory mapping, potentially moving it at the same time, depending on the flags argument and the available virtual address space.

old\_address is the old address of the virtual memory block to be resized. Note that old\_address must be page aligned. old\_size is the old size of the virtual memory block. new\_size is the requested size of the virtual memory block after the resize.

In Linux the memory is divided into pages. A user process has (one or) several linear virtual memory segments. Each virtual memory segment has one or more mappings to real memory pages (in the page table). Each virtual memory segment has its own protection (access rights), which may cause a segmentation violation if the memory is accessed incorrectly (e.g., writing to a read-only segment). Accessing virtual memory outside of the segments will also cause a segmentation violation.

mremap() uses the Linux page table scheme. mremap() changes the mapping between virtual addresses and memory pages. This can be used to implement a very efficient form of realloc().

The flags bit-mask argument may be 0, or include the following flag:

MREMAP\_MAYMOVE

  By default, if there is not sufficient space to expand a mapping at its current location, then mremap() fails. If this flag is specified, then the kernel is permitted to relocate the mapping to a new virtual address, if necessary. If the mapping is relocated, then absolute pointers into the old mapping location become invalid (offsets relative to the starting address of the mapping should be employed).

MREMAP\_FIXED

  This flag serves a similar purpose to the MAP\_FIXED flag of mmap(). If this flag is specified, then mremap()accepts a fifth argument, void \*new\_address, which specifies a pagealigned address to which the mapping must be moved. Any previous mapping at the address range specified by new\_address and new\_size is unmapped. If MREMAP\_FIXED is specified, then MREMAP\_MAYMOVE must also be specified.

If the memory segment specified by old\_address and old\_size is locked (using mlock() or similar), then this lock is maintained when the segment is resized and/or relocated. As a consequence, the amount of memory locked by the process may change.

##### Return Value

The mremap() function returns a pointer to the new virtual memory area on success. On error, the value MAP\_FAILED is returned, and errno is set appropriately.

##### Errors

EAGAIN

  The caller tried to expand a memory segment that is locked, but this was not possible without exceeding the RLIMIT\_MEMLOCK resource limit.

EFAULT

  "Segmentation fault." Some address in the range old\_address to old\_address+old\_size is an invalid virtual memory address for this process. You can also get EFAULT even if there exist mappings that cover the whole address space requested, but those mappings are of different types.

EINVAL

  An invalid argument was given. Possible causes are: old\_address was not page aligned; a value other than MREMAP\_MAYMOVE or MREMAP\_FIXED was specified in flags; new\_size was zero; new\_size or new\_address was invalid; or the new address range specified by new\_address and new\_size overlapped the old address range specified by old\_address and old\_size; or MREMAP\_FIXED was specified without also specifying MREMAP\_MAYMOVE.

ENOMEM

  The memory area cannot be expanded at the current virtual address, and the MREMAP\_MAYMOVE flag is not set in flags, or, there is not enough (virtual) memory available.

#### ngettext

##### Name

ngettext — search message catalogs for plural string

##### Synopsis

#include <libintl.h>

char \* ngettext(const char \* *msgid1*, const char \* *msgid2*, unsigned long int *n*);

##### Description

The ngettext() function shall search the currently selected message catalogs for a string matching the singular string *msgid1*. If a string is located, and if *n* is 1, that string shall be returned. If *n* is not 1, a pluralized version (dependent on *n*) of the string shall be returned.

The ngettext() function is equivalent to dcngettext(NULL, msgid1, msgid2, n, LC\_MESSAGES)().

##### Return Value

If a string is found in the currently selected message catalogs for *msgid1*, then if *n* is 1 a pointer to the located string shall be returned. If *n* is not 1, a pointer to an appropriately pluralized version of the string shall be returned. If no message could be found in the currently selected mesage catalogs, then if *n* is 1, a pointer to *msgid1* shall be returned, otherwise a pointer to *msgid2* shall be returned.

Applications shall not modify the string returned by ngettext().

##### Errors

None.

The ngettext() function shall not modify errno.

##### See Also

gettext, dgettext, ngettext, dngettext, dcgettext, dcngettext, textdomain, bindtextdomain, bind\_textdomain\_codeset

#### nrand48\_r

##### Name

nrand48\_r — reentrantly generate pseudorandom numbers in a uniform distribution

##### Synopsis

#include <stdlib.h>

int nrand48\_r(unsigned short[3] *xsubi*, struct drand48\_data \* *buffer*, long int \* *result*);

##### Description

The interface nrand48\_r() shall function in the same way as the interface nrand48(), except that nrand48\_r() shall use the data in *buffer* instead of the global random number generator state.

Before it is used, *buffer* must be initialized, for example, by calling lcong48\_r(), seed48\_r(), or srand48\_r(), or by filling it with zeroes.

#### openat64

##### Name

openat64 — open a file relative to a directory file descriptor (Large File Support)

##### Synopsis

#include <fcntl.h>

int openat64(int *fd*, const char \* *path*, int *oflag*, ...);

##### Description

openat64() shall establish a connection between a file and a file descriptor. It shall be identical open64() except in the case where *path* specifies a relative path. In this case, the file to be opened shall be determined relative to the directory associated with the file descriptor *fd* instead of the current working directory.

openat64() is a large-file version of the openat() function as defined in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4). It differs from openat() in the same way that open64() differs from open(), that the open is done in large-file mode.

##### Return Value

On success, openat64() returns a new file descriptor. Otherwise openat64() shall return -1 and set errno to indicate the error.

##### Errors

See openat() for possible error values.

#### pmap\_getport

##### Name

pmap\_getport — find the port number assigned to a service registered with a portmapper.

##### Synopsis

#include <rpc/pmap\_clnt.h>

u\_short \* pmap\_getport(struct sockaddr\_in \* *address*, const u\_long *program*, const u\_long \* *version*, u\_int *protocol*);

##### Description

The pmap\_getport() function shall return the port number assigned to a service registered with a RPC Binding service running on a given target system, using the protocol described in [RFC 1833: Binding Protocols for ONC RPC Version 2](#ID_STD_46_RFC1833). The pmap\_getport() function shall be called given the RPC program number *program*, the program version *version*, and transport protocol *protocol*. Conforming implementations shall support both IPPROTO\_UDP and IPPROTO\_TCP protocols. On entry, *address* shall specify the address of the system on which the portmapper to be contacted resides. The value of address->sin\_port shall be ignored, and the standard value for the portmapper port shall always be used.

**Note:** Security and network restrictions may prevent a conforming application from contacting a remote RPC Binding Service.

##### Return Value

On success, the pmap\_getport() function shall return the port number in host byte order of the RPC application registered with the remote portmapper. On failure, if either the program was not registered or the remote portmapper service could not be reached, the pmap\_getport() function shall return 0. If the remote portmap service could not be reached, the status is left in the global variable rpc\_createerr.

#### pmap\_set

##### Name

pmap\_set — establishes mapping to machine's RPC Bind service.

##### Synopsis

#include <rpc/pmap\_clnt.h>

bool\_t pmap\_set(const u\_long *program*, const u\_long *version*, int *protocol*, u\_short *port*);

##### Description

pmap\_set() establishes a mapping between the triple *[program,version,protocol]* and *port* on the machine's RPC Bind service. The value of *protocol* is most likely IPPROTO\_UDP or IPPROTO\_TCP. Automatically done by svc\_register().

##### Return Value

pmap\_set() returns non-zero if it suceeds, 0 otherwise.

#### pmap\_unset

##### Name

pmap\_unset — destroys RPC Binding

##### Synopsis

#include <rpc/pmap\_clnt.h>

bool\_t pmap\_unset(u\_long *prognum*, u\_long *versnum*);

##### Description

As a user interface to the RPC Bind service, pmap\_unset() destroys all mapping between the triple [*prognum*,*versnum*, *\**] and ports on the machine's RPC Bind service.

##### Return Value

pmap\_unset() returns non-zero if it succeeds, zero otherwise.

#### posix\_fadvise64

##### Name

posix\_fadvise64 — File advisory information (Large File Support)

##### Synopsis

#include <fcntl.h>

int posix\_fadvise64(int *fd*, off64\_t *offset*, off64\_t *len*, int *advice*);

##### Description

The posix\_fadvise64() function is a large-file version of the posix\_fadvise() function defined in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4). It shall advise the implementation on the expected behavior of the application with respect to the data in the file associated with the open file descriptor, *fd*, starting at *offset* and continuing for *len* bytes. The specified range need not currently exist in the file. If *len* is zero, all data following *offset* is specified. The implementation may use this information to optimize handling of the specified data. The posix\_fadvise() function shall have no effect on the semantics of other operations on the specified data, although it may affect the performance of other operations.

The advice to be applied to the data is specified by the *advice* parameter, as specified in posix\_fadvise().

##### Return Value

On success, posix\_fadvise64() shall return 0. Otherwise an error number shall be returned to indicate the error. See posix\_fadvise() for possible error values.

#### posix\_fallocate64

##### Name

posix\_fallocate64 — file space control (Large File Support)

##### Synopsis

#include <fcntl.h>

int posix\_fallocate64(int *fd*, off64\_t *offset*, off64\_t *len*);

##### Description

The posix\_fallocate64() function is a large file version of posix\_fallocate(). It shall behave as posix\_fallocate() in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except that the *offset* and *len* arguments are off64\_t objects rather than off\_t.

##### Return Value

See posix\_fallocate().

##### Errors

See posix\_fallocate().

#### pread64

##### Name

pread64 — read from a file (Large File Support)

##### Synopsis

#include <unistd.h>

ssize\_t pread64(int *fd*, void \* *buf*, size\_t *count*, off64\_t *offset*);

##### Description

pread64() shall read *count* bytes into *buf* from the file associated with the open file descriptor *fd*, at the position specified by *offset*, without changing the file position.

pread64() is a large-file version of the pread() function as defined in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4). It differs from pread() in that the *offset* parameter is an off64\_t instead of an off\_t

##### Return Value

On success, pread64() shall return the number of bytes actually read. Otherwise pread64() shall return -1 and set errno to indicate the error.

##### Errors

See pread() for possible error values.

#### ptrace

##### Name

ptrace — process trace

##### Synopsis

#include <sys/ptrace.h>

long ptrace(enum \_\_ptrace\_request *request*, pid\_t *pid*, void \* *addr*, void \* *data*);

##### Description

The ptrace() system call shall enable a process to observe and control the execution of another process, as well as examine and change certain attributes of that process.

This function operates via requests, which act on the traced process using the other parameters in ways unique to each request type. The tracing process must initiate tracing, either via the PTRACE\_TRACEME or PTRACE\_ATTACH requests, before other requests may be performed. Except for PTRACE\_TRACEME and PTRACE\_KILL, all requests must be performed on a traced process that has been stopped.

All signals, except one, delivered to the traced process cause it to stop, irrespective of its registered signal handling, and cause an event to be delivered to the tracing process which can be detected using the wait(2) system call. The exception is the SIGKILL signal, which is delivered immediately and performs its usual specified behavior.

The following requests are defined:

PTRACE\_TRACEME

  This request initates a trace from the perspective of the traced process, indicating that the parent of the current process shall be the tracing process. When this is called, a subsequent call to execve(2) shall cause the tracing process to receive a SIGTRAP signal, and shall stop the current process. This is the only request a traced process may perform, and a tracing process may not perform this request. The other parameters are ignored.

PTRACE\_ATTACH

  This request initates a trace from the perspective of the tracing process on the process specified by *pid*. After this call succeeds, the traced process will appear to be a child of the tracing process, although the original parent will still be returned to the traced process via getppid(2). The traced process will receive a SIGSTOP signal; the tracing process should use wait(2) to ensure that the traced process has stopped. A tracing process is only guaranteed to be able to trace its child processes; the tracing of other processes may not be allowed by the system, and the process with process ID 1 may not be traced under any circumstances. The *addr* and *data* parameters are ignored.

PTRACE\_CONT

  This request restarts a traced process, given in *pid*, which has been stopped. The *data* parameter may point to a signal ID to deliver to the traced process; if it is zero or SIGSTOP, no signal is delivered to the child. The *addr* is ignored.

PTRACE\_DETACH

  This request performs the same function, in the same way, as PTRACE\_CONT, except that the tracing relationship between the tracing and traced processes is also undone. If the trace was initiated using PTRACE\_ATTACH, the original parent-child relationships that existed beforehand are restored.

PTRACE\_KILL

  This request causes a SIGKILL signal to be sent to the traced process specified in *pid*. The *addr* and *data* parameters are ignored.

PTRACE\_PEEKTEXT

  This request reads a word at the location *addr* of the traced process *pid*, and returns it to the caller. The *data* parameter is ignored.

PTRACE\_PEEKDATA

  This request performs identically to the PTRACE\_PEEKTEXT request.

PTRACE\_PEEKUSER

  This request reads a word at offset *addr* in the USER area of the traced process *pid*. The offset must be word-aligned. The *data* parameter is ignored.

PTRACE\_POKETEXT

  This request writes the word pointed at by *data* to the location *addr* of the traced process *pid*.

PTRACE\_POKEDATA

  This request performs identically to the PTRACE\_POKETEXT request.

PTRACE\_POKEUSER

  This request writes the word pointed at by *data* to offset *addr* in the USER area of the traced process *pid*. The offset must be word-aligned. Implementations may choose to disallow some modifications to the USER area.

PTRACE\_GETREGS

  This request copies the general purpose registers from the traced process *pid* to the tracing process at location *data*. This parameter may not be available on all architectures. The *addr* parameter is ignored.

PTRACE\_GETFPREGS

  This request copies the floating point registers from the traced process *pid* to the tracing process at location *data*. This parameter may not be available on all architectures. The *addr* parameter is ignored.

PTRACE\_SETREGS

  This request writes the general purpose registers to the traced process *pid* from the tracing process at location *data*. This parameter may not be available on all architectures. Implementations may choose to disallow some register modifications. The *addr* parameter is ignored.

PTRACE\_SETFPREGS

  This request writes the floating point registers to the traced process *pid* from the tracing process at location *data*. This parameter may not be available on all architectures. Implementations may choose to disallow some register modifications. The *addr* parameter is ignored.

PTRACE\_GETSIGINFO

  This request writes information about the signal which caused the traced process *pid* to stop to the tracing process at location *data*, as a siginfo\_t. The *addr* parameter is ignored.

PTRACE\_SETSIGINFO

  This request writes signal information to the traced process *pid* from a siginfo\_t structure pointed at by *data*, such that it will be used as the signal information by the traced process when it is resumed. The *addr* parameter is ignored.

PTRACE\_GETEVENTMSG

  This request stores information about the most recent ptrace event for the traced process *pid* in the unsigned long pointed at by *data*. For PTRACE\_EVENT\_EXIT, this is the exit status of the traced process. For PTRACE\_EVENT\_FORK, PTRACE\_EVENT\_VFORK, or PTRACE\_EVENT\_CLONE, this is the PID of the newly created process. The *addr* parameter is ignored.

PTRACE\_SYSCALL

  This request performs the same function, in the same way, as PTRACE\_CONT, but with the additional step of causing the traced process to stop at the next entry to or exit from a system call. The usual events that would also cause the traced process to stop continue to do so.

PTRACE\_SINGLESTEP

  This request performs the same function, in the same way, as PTRACE\_CONT, but with the additional step of causing the traced process to stop after execution of a single instruction. The usual events that would also cause the traced process to stop continue to do so.

PTRACE\_SYSEMU

  This request performs the same function, in the same way, as PTRACE\_CONT, but with the additional step of causing the traced process to stop on entry to the next syscall, which will then not be executed.

PTRACE\_SYSEMU\_SINGLESTEP

  This request performs the same function, in the same way, as PTRACE\_CONT, but with the additional step of causing the traced process to stop on entry to the next syscall, which will then not be executed. If the next instruction is not itself a syscall, the traced process will stop after a single instruction is executed.

PTRACE\_SETOPTIONS

  This request sets ptrace() options for the traced process *pid* from the location pointed to by *data*. The *addr* is ignored. This location is interpreted as a bitmask of options, as defined by the following flags:

PTRACE\_O\_TRACESYSGOOD

  This option, when set, causes syscall traps to set bit 7 in the signal number.

PTRACE\_O\_TRACEFORK

  This option, when set, causes the traced process to stop when it calls fork(2). The original traced process will stop with SIGTRAP | PTRACE\_EVENT\_FORK << 8, and the new process will be stopped with SIGSTOP. The new process will also be traced by the tracing process, as if the tracing process had sent the PTRACE\_ATTACH request for that process. The PID of the new process may be retrieved with the PTRACE\_GETEVENTMSG request.

PTRACE\_O\_TRACEVFORK

  This option, when set, causes the traced process to stop when it calls vfork(2). The original traced process will stop with SIGTRAP | PTRACE\_EVENT\_VFORK << 8, and the new process will be stopped with SIGSTOP. The new process will also be traced by the tracing process, as if the tracing process had sent the PTRACE\_ATTACH request for that process. The PID of the new process may be retrieved with the PTRACE\_GETEVENTMSG request.

PTRACE\_O\_TRACECLONE

  This option, when set, causes the traced process to stop when it calls clone(2). The original traced process will stop with SIGTRAP | PTRACE\_EVENT\_CLONE << 8, and the new process will be stopped with SIGSTOP. The new process will also be traced by the tracing process, as if the tracing process had sent the PTRACE\_ATTACH request for that process. The PID of the new process may be retrieved with the PTRACE\_GETEVENTMSG request. Under certain circumstances, clone(2) calls by the traced process will generate events and information consistent with the PTRACE\_O\_TRACEVFORK or PTRACE\_O\_TRACEFORK options above.

PTRACE\_O\_TRACEEXEC

  This option, when set, causes the traced process to stop when it calls execve(2). The traced process will stop with SIGTRAP | PTRACE\_EVENT\_EXEC << 8.

PTRACE\_O\_TRACEVFORKDONE

  This option, when set, causes the traced process to stop at the completion of its next vfork(2) call. The traced process will stop with SIGTRAP | PTRACE\_EVENT\_EXEC << 8.

PTRACE\_O\_TRACEEXIT

  This option, when set, causes the traced process to stop upon exit. The traced process will stop with SIGTRAP | PTRACE\_EVENT\_EXIT << 8, and its exit status can be retrieved with the PTRACE\_GETEVENTMSG request. The stop is guaranteed to be early in the process exit process, meaning that information such as register status at exit is preserved. Upon continuing, the traced process will immediately exit.

##### Return Value

On success, ptrace() shall return the requested data for PTRACE\_PEEK requests, or zero for all other requests. On error, all requests return -1, with errno set to an appropriate value. Note that -1 may be a valid return value for PTRACE\_PEEK requests; the application is responsible for distinguishing between an error condition and a valid return value in that case.

##### Errors

On error, ptrace() shall set errno to one of the regular error values below:

EBUSY

  An error occurred while allocating or freeing a debug register.

EFAULT

  The request attempted to read from or write to an invalid area in the memory space of the tracing or traced process.

EIO

  The request was invalid, or it attempted to read from or write to an invalid area in the memory space of the tracing or traced process, or it violated a word-alignment boundary, or an invalid signal was given to continue the traced process.

EINVAL

  An attempt was made to set an invalid option.

EPERM

  The request to trace a process was denied by the system.

ESRCH

  The process requested does not exist, is not being traced by the current process, or is not stopped.

#### putwc\_unlocked

##### Name

putwc\_unlocked — non-thread-safe putwc

##### Description

putwc\_unlocked() is the same as putwc(), except that it need not be thread-safe. That is, it may only be invoked in the ways which are legal for getc\_unlocked().

#### putwchar\_unlocked

##### Name

putwchar\_unlocked — non-thread-safe putwchar

##### Description

putwchar\_unlocked() is the same as putwchar(), except that it need not be thread-safe. That is, it may only be invoked in the ways which are legal for getc\_unlocked().

#### pwrite64

##### Name

pwrite64 — write on a file (Large File Support)

##### Synopsis

#include <unistd.h>

ssize\_t pwrite64(int *fd*, const void \* *buf*, size\_t *count*, off64\_t *offset*);

##### Description

pwrite64() shall write *count* bytes from *buf* to the file associated with the open file descriptor *fd*, at the position specified by *offset*, without changing the file position.

pwrite64() is a large-file version of the pwrite() function as defined in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4). It differs from pwrite() in that the *offset* parameter is an off64\_t instead of an off\_t

##### Return Value

On success, pwrite64() shall return the number of bytes actually written. Otherwise pwrite() shall return -1 and set errno to indicate the error.

##### Errors

See pwrite() for possible error values.

#### random\_r

##### Name

random\_r — reentrantly generate pseudorandom numbers in a uniform distribution

##### Synopsis

#include <stdlib.h>

int random\_r(struct random\_data \* *buffer*, int32\_t \* *result*);

##### Description

The interface random\_r() shall function in the same way as the interface random(), except that random\_r() shall use the data in *buffer* instead of the global random number generator state.

Before it is used, *buffer* must be initialized, for example, by calling lcong48\_r(), seed48\_r(), or srand48\_r(), or by filling it with zeroes.

#### readdir64\_r

##### Name

readdir64\_r — read a directory (Large File Support)

##### Synopsis

#include <dirent.h>

int readdir64\_r(DIR \* *dirp*, struct dirent64 \* *entry*, struct dirent64 \* \* *result*);

##### Description

The readdir64\_r() function is a large file version of readdir\_r(). It shall behave as readdir\_r() in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except that the *entry* and *result* arguments are dirent64 structures rather than dirent.

##### Return Value

See readdir\_r().

##### Errors

See readdir\_r().

#### regexec

##### Name

regexec — regular expression matching

##### Description

The regexec() function shall behave as specified in [*POSIX 1003.1-2008 (ISO/IEC 9945-2009)*](#ID_STD_46_SUSV4), except with differences as listed below.

#### Differences

Certain aspects of regular expression matching are optional; see [Regular Expressions](#ID_LOCALIZATION_45_REGEX).

#### scandir64

##### Name

scandir64 — scan a directory (Large File Support)

##### Synopsis

#include <dirent.h>

int scandir64(const char \* *dir*, const struct dirent64 \*\* *namelist*, int (\*sel) (const struct dirent64 \*), int (\**compar*) (const struct dirent64 \*\*, const struct dirent64 \*\*));

##### Description

scandir64() is a large-file version of the scandir() function as defined in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4). If differs only in that the *namelist* and the paramters to the selection function *sel* and comparison function *compar* are of type dirent64 instead of type dirent.

#### scanf

##### Name

scanf — convert formatted input

##### Description

The scanf() family of functions shall behave as described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except as noted below.

##### Differences

The %s, %S and %[ conversion specifiers shall accept an option length modifier a, which shall cause a memory buffer to be allocated to hold the string converted. In such a case, the argument corresponding to the conversion specifier should be a reference to a pointer value that will receive a pointer to the allocated buffer. If there is insufficient memory to allocate a buffer, the function may set errno to ENOMEM and a conversion error results.

**Note:** This directly conflicts with the [ISO C (1999)](#ID_STD_46_ISOC99) usage of %a as a conversion specifier for hexadecimal float values. While this conversion specifier should be supported, a format specifier such as "%aseconds" will have a different meaning on an LSB conforming system.

#### sched\_getaffinity

##### Name

sched\_getaffinity — retrieve the affinity mask of a process

##### Synopsis

#include <sched.h>

int sched\_getaffinity(pid\_t *pid*, unsigned int *cpusetsize*, cpu\_set\_t \* *mask*);

##### Description

sched\_getaffinity() shall retrieve the affinity mask of a process.

The parameter *pid* specifies the ID for the process. If *pid* is 0, then the calling process is specified instead.

The parameter *cpusetsize* specifies the length of the data pointed to by *mask*, in bytes. Normally, this parameter is specified as sizeof(cpu\_set\_t).

##### Return Value

On success, sched\_getaffinity() shall return 0, and the structure pointed to by *mask* shall contain the affinity mask of the specified process.

On failure, sched\_getaffinity() shall return -1 and set errno as follows.

##### Errors

EFAULT

  Bad address.

EINVAL

*mask* does not specify any processors that exist in the system, or *cpusetsize* is smaller than the kernel's affinity mask.

ESRCH

  The specified process could not be found.

##### See Also

sched\_setscheduler(), sched\_setaffinity().

#### sched\_setaffinity

##### Name

sched\_setaffinity — set the CPU affinity mask for a process

##### Synopsis

#include <sched.h>

int sched\_setaffinity(pid\_t *pid*, unsigned int *cpusetsize*, cpu\_set\_t \* *mask*);

##### Description

sched\_setaffinity() shall set the CPU affinity mask for a process.

The parameter *pid* specifies the ID for the process. If *pid* is 0, then the calling process is specified instead.

The parameter *cpusetsize* specifies the length of the data pointed to by *mask*, in bytes. Normally, this parameter is specified as sizeof(cpu\_set\_t).

The parameter *mask* specifies the new value for the CPU affinity mask. The structure pointed to by *mask* represents the set of CPUs on which the process may run. If *mask* does not specify one of the CPUs on which the specified process is currently running, then sched\_setaffinity() shall migrate the process to one of those CPUs.

Setting the mask on a multiprocessor system can improve performance. For example, setting the mask for one process to specify a particular CPU, and then setting the mask of all other processes to exclude the CPU, dedicates the CPU to the process so that the process runs as fast as possible. This technique also prevents loss of performance in case the process terminates on one CPU and starts again on another, invalidating cache.

##### Return Value

On success, sched\_setaffinity() shall return 0.

On failure, sched\_setaffinity() shall return -1 and set errno as follows.

##### Errors

EFAULT

  Bad address.

EINVAL

*mask* does not specify any processors that exist in the system, or *cpusetsize* is smaller than the kernel's affinity mask.

EPERM

  Insufficient privileges. The effective user ID of the process calling sched\_setaffinity() is not equal to the user ID or effective user ID of the specified process, and the calling process does not have appropriate privileges.

ESRCH

  The specified process could not be found.

##### See Also

sched\_setscheduler(), sched\_getaffinity().

#### sched\_setscheduler

##### Name

sched\_setscheduler — set scheduling policy and parameters

##### Synopsis

#include <sched.h>

int sched\_setscheduler(pid\_t *pid*, int *policy*, const struct sched\_param \* *param*);

##### Description

The sched\_setscheduler() shall behave as described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except as noted below.

##### Return Value

On success, 0 is returned instead of the former scheduling policy.

#### seed48\_r

##### Name

seed48\_r — reentrantly generate pseudorandom numbers in a uniform distribution

##### Synopsis

#include <stdlib.h>

int seed48\_r(unsigned short[3] *seed16v*, struct drand48\_data \* *buffer*);

##### Description

The interface seed48\_r() shall function in the same way as the interface seed48(), except that seed48\_r() shall use the data in *buffer* instead of the global random number generator state.

#### sendfile

##### Name

sendfile — transfer data between two file descriptors

##### Synopsis

#include <sys/sendfile.h>

ssize\_t sendfile(int *out\_fd*, int *in\_fd*, off\_t \* *offset*, size\_t *count*);

##### Description

The sendfile() function shall copy data between the file descriptor *in\_fd*, which must not be a socket, and the file descriptor *out\_fd*, which must be a socket. *in\_fd* should be opened for reading, and *out\_fd* should be opened for writing.

The *offset* parameter points to a variable set to the file offset at which sendfile() shall start reading from *in\_fd*, unless it is NULL. On exit, this variable shall contain the offset of the byte immediately after the last byte read. sendfile() shall not change the current file offset of *in\_fd*, unless it is NULL. In that case, sendfile() shall adjust the current file offset to show how many bytes were read.

The *count* parameter specifies how many bytes to copy.

##### Return Value

On success, sendfile() shall return the number of bytes written to *out\_fd*.

On failure, sendfile() shall return -1 and set errno appropriately, as follows.

##### Errors

EAGAIN

  Non-blocking I/O with O\_NONBLOCK has been chosen, but the write would block.

EBADF

  The input file is not open for reading, or the output file is not open for writing.

EFAULT

  Bad address.

EINVAL

  An mmap()-like operation is unavailable for *in\_fd*, or file descriptor is locked or invalid.

EIO

  There was an unspecified error while reading.

ENOMEM

  There is not enough memory to read from *in\_fd*.

##### Notes

sendfile() is usually faster than combining read() and write() calls, because it is part of the kernel. However, if it fails with EINVAL, falling back to read() and write() may be advisable.

It is advisable for performance reasons to use the TCP\_CORK option of the tcp() function when sending header data with file contents to a TCP socket. This minimizes the number of packets.

##### See Also

mmap(), open(), socket(), splice().

#### sendfile64

##### Name

sendfile64 — transfer data between two file descriptors (Large File Support)

##### Synopsis

#include <sys/sendfile.h>

ssize\_t sendfile64(int *out\_fd*, int *in\_fd*, off64\_t \* *offset*, size\_t *count*);

##### Description

The sendfile64() function is a large-file version of the sendfile() function.

#### setbuffer

##### Name

setbuffer — stream buffering operation

##### Synopsis

#include <stdio.h>

void setbuffer(FILE \* *stream*, char \* *buf*, size\_t *size*);

##### Description

setbuffer() is an alias for the call to setvbuf(). It works the same, except that the size of the buffer in setbuffer() is up to the caller, rather than being determined by the default *BUFSIZ*.

#### setgroups

##### Name

setgroups — set list of supplementary group IDs

##### Synopsis

#include <grp.h>

int setgroups(size\_t *size*, const gid\_t \* *list*);

##### Description

If the process has appropriate privilege, the setgroups() function shall set the supplementary group IDs for the current process. *list* shall reference an array of *size* group IDs. A process may have at most NGROUPS\_MAX supplementary group IDs.

##### Return Value

On successful completion, 0 is returned. On error, -1 is returned and the errno is set to indicate the error.

##### Errors

EFAULT

*list* has an invalid address.

EPERM

  The process does not have appropriate privileges.

EINVAL

*size* is greater than NGROUPS\_MAX.

#### sethostname

##### Name

sethostname — set host name

##### Synopsis

#include <unistd.h>

#include <sys/param.h.h>

#include <sys/utsname.h>

int sethostname(const char \* *name*, size\_t *len*);

##### Description

If the process has appropriate privileges, the sethostname() function shall change the host name for the current machine. The *name* shall point to a null-terminated string of at most *len* bytes that holds the new hostname.

If the symbol HOST\_NAME\_MAX is defined, or if sysconf(\_SC\_HOST\_NAME\_MAX)() returns a value greater than 0, this value shall represent the maximum length of the new hostname. Otherwise, if the symbol MAXHOSTLEN is defined, this value shall represent the maximum length for the new hostname. If none of these values are defined, the maximum length shall be the size of the *nodename* field of the utsname structure.

##### Return Value

On success, 0 is returned. On error, -1 is returned and the global variable errno is set appropriately.

##### Errors

EINVAL

*len* is negative or larger than the maximum allowed size.

EPERM

  the process did not have appropriate privilege.

EFAULT

*name* is an invalid address.

##### Rationale

[POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) guarantees that:

Maximum length of a host name (not including the terminating null) as returned from the gethostname() function shall be at least 255 bytes.

The glibc C library does not currently define HOST\_NAME\_MAX, and although it provides the name \_SC\_HOST\_NAME\_MAX a call to sysconf() returns -1 and does not alter errno in this case (indicating that there is no restriction on the hostname length). However, the glibc manual idicates that some implementations may have MAXHOSTNAMELEN as a means of detecting the maximum length, while the Linux kernel at release 2.4 and 2.6 stores this hostname in the utsname structure. While the glibc manual suggests simply shortening the name until sethostname() succeeds, the LSB requires that one of the first four mechanisms works. Future versions of glibc may provide a more reasonable result from sysconf(\_SC\_HOST\_NAME\_MAX).

#### setsockopt

##### Name

setsockopt — set socket options

##### Synopsis

#include <sys/socket.h>

#include <netinet/ip.h>

int setsockopt(int *socket*, int *level*, int *option\_name*, const void \* *option\_value*, socklen\_t *option\_len*);

##### Description

The setsockopt() function shall behave as specified in [*POSIX 1003.1-2008 (ISO/IEC 9945-2009)*](#ID_STD_46_SUSV4), with the following extensions.

#### IP Protocol Level Options

If the *level* parameter is IPPROTO\_IP, the following values shall be supported for *option\_name* (see [RFC 791:Internet Protocol](#ID_STD_46_RFC791) for further details):

IP\_OPTIONS

  Set the Internet Protocol options sent with every packet from this socket. The *option\_value* shall point to a memory buffer containing the options and *option\_len* shall contain the size in bytes of that buffer. For IPv4, the maximum length of options is 40 bytes.

IP\_TOS

  Set the Type of Service flags to use when sending packets with this socket. The *option\_value* shall point to a value containing the type of service value. The least significant two bits of the value shall contain the new Type of Service indicator. Use of other bits in the value is unspecified. The *option\_len* parameter shall hold the size, in bytes, of the buffer referred to by *option\_value*.

IP\_TTL

  Set the current unicast Internet Protocol Time To Live value used when sending packets with this socket. The *option\_value* shall point to a value containing the time to live value, which shall be between 1 and 255. The *option\_len* parameter shall hold the size, in bytes, of the buffer referred to by *option\_value*.

IP\_MULTICAST\_TTL

  Sets the Time To Live value of outgoing multicast packets for this socket. *optval* shall point to an integer which contains the new TTL value. If the new TTL value is -1, the implementation should use an unspecified default TTL value. If the new TTL value is out of the range of acceptable values (0-255), setsockopt() shall return -1 and set errno to indicate the error.

IP\_MULTICAST\_LOOP

  Sets a boolean flag indicating whether multicast packets originating locally should be looped back to the local sockets. *optval* shall point to an integer which contains the new flag value.

IP\_ADD\_MEMBERSHIP

  Join a multicast group. *optval* shall point to a ip\_mreq structure. Before calling, the caller should fill in the *imr\_multiaddr* field with the multicast group address and the *imr\_address* field with the address of the local interface. If *imr\_address* is set to INADDR\_ANY, then an appropriate interface is chosen by the system.

IP\_DROP\_MEMBERSHIP

  Leave a multicast group. *optval* shall point to a ip\_mreq structure containing the same values as were used with IP\_ADD\_MEMBERSHIP.

IP\_MULTICAST\_IF

  Set the local device for a multicast socket. *optval* shall point to either an ip\_mreqn structure or an in\_addr structure. If using the ip\_mreqn structure, the *imr\_multiaddr* field should be set to multicast group address, the *imr\_address* field to the address of the local interface, and the *imr\_index* field to the interface index. If using the in\_addr structure, the address of the local interface shall be specified. If *in\_addr* or *imr\_address* is set to INADDR\_ANY, then an appropriate interface is chosen by the system. If *imr\_index* is zero, then an appropriate interface index is chosen by the implementation.

The ip\_mreq structure contains two struct in\_addr fields: *imr\_multiaddr* and *imr\_address*.

##### Return Value

On success, 0 is returned. On error, -1 is returned and the global variable errno is set appropriately.

##### Errors

As defined in [POSIX 1003.1-2001 (ISO/IEC 9945-2003)](#ID_STD_46_SUSV3).

#### setstate\_r

##### Name

setstate\_r — reentrantly change the state array used by random number generator functions

##### Synopsis

#include <stdlib.h>

int setstate\_r(char \* *statebuf*, struct random\_data \* *buf*);

##### Description

The interface setstate\_r() shall function in the same way as the interface setstate(), except that setstate\_r() shall use the data in *statebuf* instead of the global random number generator state.

#### setutent

##### Name

setutent — access user accounting database entries

##### Synopsis

#include <utmp.h>

void setutent(void);

##### Description

The setutent() function shall reset the user accounting database such that the next call to getutent() shall return the first record in the database. It is recommended to call it before any of the other functions that operate on the user accounting databases (e.g. getutent())

##### Return Value

None.

#### sigandset

##### Name

sigandset — build a new signal set by combining the two input sets using logical AND

##### Synopsis

#include <signal.h>

int sigandset(sigset\_t \* *set*, const sigset\_t \* *left*, const sigset\_t \* *right*);

##### Description

The sigandset() function shall combine the two signal sets referenced by *left* and *right*, using a logical AND operation, and shall place the result in the location referenced by *set*, The resulting signal set shall contain only signals that are in both the set referenced by *left* and the set referenced by *right*.

Applications shall call sigemptyset() or sigfillset() at least once for each object of type sigset\_t to initialize it. If an uninitialized or NULL object is passed to sigandset(), the results are undefined.

##### Return Value

sigandset() returns 0. There are no defined error returns.

##### See Also

sigorset()

#### sigisemptyset

##### Name

sigisemptyset — check for empty signal set

##### Synopsis

#include <signal.h>

int sigisemptyset(const sigset\_t \* *set*);

##### Description

The sigisemptyset() function shall check for empty signal set referenced by *set*.

Applications shall call sigemptyset() or sigfillset() at least once for each object of type sigset\_t to initialize it. If an uninitialized or NULL object is passed to sigisemptyset(), the results are undefined.

##### Return Value

The sigisemptyset() function shall return a positive non-zero value if the signal set referenced by *set* is empty, or zero if this set is empty. There are no defined error returns.

#### sigorset

##### Name

sigorset — build a new signal set by combining the two input sets using logical OR

##### Synopsis

#include <signal.h>

int sigorset(sigset\_t \* *set*, const sigset\_t \* *left*, const sigset\_t \* *right*);

##### Description

The sigorset() function shall combine the two signal sets referenced by *left* and *right*, using a logical OR operation, and shall place the result in the location referenced by *set*, The resulting signal set shall contain only signals that are in either the set referenced by *left* or the set referenced by *right*.

Applications shall call sigemptyset() or sigfillset() at least once for each object of type sigset\_t to initialize it. If an uninitialized or NULL object is passed to sigorset(), the results are undefined.

##### Return Value

sigorset() returns 0. There are no defined error returns.

##### See Also

sigandset()

#### sigpause

##### Name

sigpause — remove a signal from the signal mask and suspend the thread (deprecated)

##### Synopsis

#include <signal.h>

int sigpause(int *sig*);

##### Description

The sigpause() function is deprecated from the LSB and is expected to disappear from a future version of the LSB. Conforming applications should use sigsuspend() instead.

In the source standard, sigpause() is implemented as a macro causing it to behave as described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), and is equivalent to the function \_\_xpg\_sigpause(). If the macro is undefined, sigpause() from the binary standard is used, with differences as described here:

The sigpause() function shall block those signals indicated by *sig* and suspend execution of the thread until a signal is delivered. When a signal is delivered, the original signal mask shall be restored.

##### See Also

\_\_xpg\_sigpause()

#### sigreturn

##### Name

sigreturn — return from signal handler and cleanup stack frame

##### Synopsis

int sigreturn(struct sigcontext \* *scp*);

##### Description

The sigreturn() function is used by the system to cleanup after a signal handler has returned. This function is not in the source standard; it is only in the binary standard.

##### Return Value

sigreturn() never returns.

#### srand48\_r

##### Name

srand48\_r — reentrantly generate pseudorandom numbers in a uniform distribution

##### Synopsis

#include <stdlib.h>

int srand48\_r(long int *seedval*, struct drand48\_data \* *buffer*);

##### Description

The interface srand48\_r() shall function in the same way as the interface srand48(), except that srand48\_r() shall use the data in *buffer* instead of the global random number generator state.

#### srandom\_r

##### Name

srandom\_r — reentrantly set the seed for a new sequence of pseudorandom numbers

##### Synopsis

#include <stdlib.h>

int srandom\_r(unsigned int *seed*, struct random\_data \* *buffer*);

##### Description

The interface srandom\_r() shall function in the same way as the interface srandom(), except that srandom\_r() shall use the data in *buffer* instead of the global random number generator state.

#### sscanf

##### Name

sscanf — convert formatted input

##### Description

The scanf() family of functions shall behave as described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except as noted below.

##### Differences

The %s, %S and %[ conversion specifiers shall accept an option length modifier a, which shall cause a memory buffer to be allocated to hold the string converted. In such a case, the argument corresponding to the conversion specifier should be a reference to a pointer value that will receive a pointer to the allocated buffer. If there is insufficient memory to allocate a buffer, the function may set errno to ENOMEM and a conversion error results.

**Note:** This directly conflicts with the [ISO C (1999)](#ID_STD_46_ISOC99) usage of %a as a conversion specifier for hexadecimal float values. While this conversion specifier should be supported, a format specifier such as "%aseconds" will have a different meaning on an LSB conforming system.

#### statfs

##### Name

statfs — (deprecated)

##### Synopsis

#include <sys/statfs.h>

int statfs(const char \**path*, struct statfs \**buf*);

##### Description

The statfs() function returns information about a mounted file system. The file system is identified by *path*, a path name of a file within the mounted filesystem. The results are placed in the structure pointed to by

Fields that are undefined for a particular file system shall be set to 0.

**Note:** Application developers should use the statvfs() function to obtain general file system information. Applications should only use the statfs() function if they must determine the file system type, which need not be provided by statvfs().

##### Return Value

On success, the statfs() function shall return 0 and set the fields of the structure idenfitied by *buf* accordingly. On error, the statfs() function shall return -1 and set errno accordingly.

##### Errors

ENOTDIR

  A component of the path prefix of *path* is not a directory.

ENAMETOOLONG

*path* is too long.

ENOENT

  The file referred to by *path* does not exist.

EACCES

  Search permission is denied for a component of the path prefix of *path*.

ELOOP

  Too many symbolic links were encountered in translating *path*.

EFAULT

*buf* or *path* points to an invalid address.

EIO

  An I/O error occurred while reading from or writing to the file system.

ENOMEM

  Insufficient kernel memory was available.

ENOSYS

  The filesystem *path* is on does not support statfs().

#### statfs64

##### Name

statfs64 — (deprecated)

##### Synopsis

#include <sys/statfs.h>

int statfs64(const char \* *path*, struct statfs64 \**buf*);

##### Description

The statfs64() function returns information about a mounted file system. The file system is identified by *path*, a path name of a file within the mounted filesystem. The results are placed in the structure pointed to by *buf*.

statfs64() is a large-file version of the statfs() function.

Fields that are undefined for a particular file system shall be set to 0.

**Note:** Application developers should use the statvfs64() function to obtain general file system information. Applications should only use the statfs64() function if they must determine the file system type, which need not be provided by statvfs64().

##### Return Value

On success, the statfs64() function shall return 0 and set the fields of the structure idenfitied by *buf* accordingly. On error, the statfs64() function shall return -1 and set errno accordingly.

##### Errors

See fstatfs().

#### stime

##### Name

stime — set time

##### Synopsis

#define \_SVID\_SOURCE

#include <time.h>

int stime(const time\_t \* *t*);

##### Description

If the process has appropriate privilege, the stime() function shall set the system's idea of the time and date. Time, referenced by *t*, is measured in seconds from the epoch (defined in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) as 00:00:00 UTC January 1, 1970).

##### Return Value

On success, stime() shall return 0. Otherwise, stime() shall return -1 and errno shall be set to indicate the error.

##### Errors

EPERM

  The process does not have appropriate privilege.

EINVAL

*t* is a null pointer.

#### strcasestr

##### Name

strcasestr — locate a substring ignoring case

##### Synopsis

#include <string.h>

char \* strcasestr(const char \* *s1*, const char \* *s2*);

##### Description

The strcasestr() shall behave as strstr(), except that it shall ignore the case of both strings. The strcasestr() function shall be locale aware; that is strcasestr() shall behave as if both strings had been converted to lower case in the current locale before the comparison is performed.

##### Return Value

Upon successful completion, strcasestr() shall return a pointer to the located string or a null pointer if the string is not found. If *s2* points to a string with zero length, the function shall return *s1*.

#### strerror\_r

##### Name

strerror\_r — return string describing error number

##### Synopsis

#include <string.h>

char \* strerror\_r(int *errnum*, char \* *buf*, size\_t *buflen*);

##### Description

In the source standard, strerror\_r() is implemented as a macro causing it to behave as described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), and is equivalent to the function \_\_xpg\_strerror\_r(). If the macro is undefined, strerror\_r() from the binary standard is used, with differences as described here.

The strerror\_r() function shall return a pointer to the string corresponding to the error number *errnum*. The returned pointer may point within the buffer *buf* (at most *buflen* bytes).

##### Return Value

On success, strerror\_r() shall return a pointer to the generated message string (determined by the setting of the LC\_MESSAGES category in the current locale). Otherwise, strerror\_r() shall return the string corresponding to "Unknown error".

##### See Also

\_\_xpg\_strerror\_r()

#### strptime

##### Name

strptime — parse a time string

##### Description

The strptime() shall behave as specified in the [*POSIX 1003.1-2008 (ISO/IEC 9945-2009)*](#ID_STD_46_SUSV4) with differences as listed below.

#### Number of leading zeroes may be limited

The [*POSIX 1003.1-2008 (ISO/IEC 9945-2009)*](#ID_STD_46_SUSV4) specifies fields for which "leading zeros are permitted but not required"; however, applications shall not expect to be able to supply more leading zeroes for these fields than would be implied by the range of the field. Implementations may choose to either match an input with excess leading zeroes, or treat this as a non-matching input. For example, %j has a range of 001 to 366, so 0, 00, 000, 001, and 045 are acceptable inputs, but inputs such as 0000, 0366 and the like are not.

##### Rationale

*glibc* developers consider it appropriate behavior to forbid excess leading zeroes. When trying to parse a given input against several format strings, forbidding excess leading zeroes could be helpful. For example, if one matches 0011-12-26 against %m-%d-%Y and then against %Y-%m-%d, it seems useful for the first match to fail, as it would be perverse to parse that date as November 12, year 26. The second pattern parses it as December 26, year 11.

The [*POSIX 1003.1-2008 (ISO/IEC 9945-2009)*](#ID_STD_46_SUSV4) is not explicit that an unlimited number of leading zeroes are required, although it may imply this. The LSB explicitly allows implementations to have either behavior. Future versions of this standard may require implementations to forbid excess leading zeroes.

An Interpretation Request is currently pending against [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) for this matter.

#### strsep

##### Name

strsep — extract token from string

##### Synopsis

#include <string.h>

char \* strsep(char \* \* *stringp*, const char \* *delim*);

##### Description

The strsep() function shall find the first token in the string referenced by the pointer *stringp*, using the characters in *delim* as delimiters.

If *stringp* is NULL, strsep() shall return NULL and do nothing else.

If *stringp* is non-NULL, strsep() shall find the first token in the string referenced by *stringp*, where tokens are delimited by characters in the string *delim*. This token shall be terminated with a \0 character by overwriting the delimiter, and *stringp* shall be updated to point past the token. In case no delimiter was found, the token is taken to be the entire string referenced by *stringp*, and the location referenced by *stringp* is made NULL.

##### Return Value

strsep() shall return a pointer to the beginning of the token.

##### Notes

The strsep() function was introduced as a replacement for strtok(), since the latter cannot handle empty fields. However, strtok() conforms to [ISO C (1999)](#ID_STD_46_ISOC99) and to [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) and hence is more portable.

##### See Also

strtok(), strtok\_r().

#### strtoq

##### Name

strtoq — convert string value to a long or quad\_t integer

##### Synopsis

#include <sys/types.h>

#include <stdlib.h>

#include <limits.h>

long long strtoq(const char \* *nptr*, char \* \* *endptr*, int *base*);

##### Description

strtoq() converts the string *nptr* to a quadt value. The conversion is done according to the given base, which shall be between 2 and 36 inclusive, or be the special value 0.

*nptr* may begin with an arbitrary amount of white space (as determined by isspace()), followed by a single optional + or - sign character. If *base* is 0 or 16, the string may then include a 0x prefix, and the number will be read in base 16; otherwise, a 0 base is taken as 10 (decimal), unless the next character is 0, in which case it is taken as 8 (octal).

The remainder of the string is converted to a long value in the obvious manner, stopping at the first character which is not a valid digit in the given base. (In bases above 10, the letter A in either upper or lower case represents 10, B represents 11, and so forth, with Z representing 35.)

##### Return Value

strtoq() returns the result of the conversion, unless the value would underflow or overflow. If an underflow occurs, strtoq() returns QUAD\_MIN. If an overflow occurs, strtoq() returns QUAD\_MAX. In both cases, the global variable errno is set to ERANGE.

##### Errors

ERANGE

  The given string was out of range; the value converted has been clamped.

#### strtouq

##### Name

strtouq — convert a string to an unsigned long long

##### Synopsis

#include <sys/types.h>

#include <stdlib.h>

#include <limits.h>

unsigned long long strtouq(const char \* *nptr*, char \* \* *endptr*, int *base*);

##### Description

strtouq() converts the string *nptr* to an unsigned long long value. The conversion is done according to the given base, which shall be between 2 and 36 inclusive, or be the special value 0.

*nptr* may begin with an arbitrary amount of white space (as determined by isspace()), followed by a single optional + or - sign character. If *base* is 0 or 16, the string may then include a 0x prefix, and the number will be read in base 16; otherwise, a 0 base is taken as 10 (decimal), unless the next character is 0, in which case it is taken as 8 (octal).

The remainder of the string is converted to an unsigned long value in the obvious manner, stopping at the end of the string or at the first character that does not produce a valid digit in the given base. (In bases above 10, the letter A in either upper or lower case represents 10, B represents 11, and so forth, with Z representing 35.)

##### Return Value

On success, strtouq() returns either the result of the conversion or, if there was a leading minus sign, the negation of the result of the conversion, unless the original (non-negated) value would overflow. In the case of an overflow the function returns UQUAD\_MAX and the global variable errno is set to ERANGE.

##### Errors

ERANGE

  The given string was out of range; the value converted has been clamped.

#### svc\_register

##### Name

svc\_register — register Remote Procedure Call interface

##### Synopsis

#include <rpc/rpc.h>

bool\_t svc\_register(SVCXPRT \* *xprt*, rpcprog\_t *prognum*, rpcvers\_t *versnum*, \_\_dispatch\_fn\_t *dispatch*, rpcprot\_t *protocol*);

##### Description

The svc\_register() function shall associate the program identified by *prognum* at version *versnum* with the service dispatch procedure, *dispatch*. If *protocol* is zero, the service is not registered with the portmap service. If *protocol* is non-zero, then a mapping of the triple [*prognum*, *versnum*, *protocol*] to xprt->xp\_port is established with the local portmap service. The procedure *dispatch* has the following form:

int dispatch(struct svc\_req \* *request*, SVCXPRT \* *xprt*);

##### Return Value

svc\_register() returns 1 if it succeeds, and zero otherwise.

#### svc\_run

##### Name

svc\_run — waits for RPC requests to arrive and calls service procedure

##### Synopsis

#include <rpc/svc.h>

void svc\_run(void);

##### Description

The svc\_run() function shall wait for RPC requests to arrive, read and unpack each request, and dispatch it to the appropriate registered handler. Under normal conditions, svc\_run() shall not return; it shall only return if serious errors occur that prevent further processing.

#### svc\_sendreply

##### Name

svc\_sendreply — called by RPC service's dispatch routine

##### Synopsis

bool\_t svc\_sendreply(SVCXPRT \**xprt*, xdrproc\_t *outproc*, caddr\_t *out*);

##### Description

Called by an RPC service's dispatch routine to send the results of a remote procedure call. The parameter *xprt* is the request's associated transport handle; *outproc* is the XDR routine which is used to encode the results; and *out* is the address of the results. This routine returns one if it succeeds, zero otherwise.

#### svctcp\_create

##### Name

svctcp\_create — create a TCP/IP-based RPC service transport

##### Synopsis

#include <rpc/rpc.h>

SVCXPRT \* svctcp\_create(int *sock*, u\_int *send\_buf\_size*, u\_int *recv\_buf\_size*);

##### Description

svctcp\_create() creates a TCP/IP-based RPC service transport, to which it returns a pointer. The transport is associated with the socket *sock*, which may be RPC\_ANYSOCK, in which case a new socket is created. If the socket is not bound to a local TCP port, then this routine binds it to an arbitrary port. Upon completion, xprt->xp\_sock is the transport's socket descriptor, and xprt->xp\_port is the transport's port number. Since TCP-based RPC uses buffered I/O, users may specify the size of buffers; values of zero choose suitable defaults.

##### Return Value

svctcp\_create() returns NULL if it fails, or a pointer to the RPC service transport otherwise.

#### svcudp\_create

##### Name

svcudp\_create — create a UDP-based RPC service transport

##### Synopsis

SVCXPRT \*

svcudp\_create(int *sock*);

##### Description

The svcudp\_create() function shall create a UDP/IP-based RPC service transport, and return a pointer to its descriptor. The transport is associated with the socket *sock*, which may be RPC\_ANYSOCK, in which case a new socket shall be created. If the socket is not bound to a local UDP port, then svcudp\_create() shall bind it to an arbitrary port.

If svcudp\_create() returns successfully, then the *xp\_sock* field in the result shall be the transport's socket descriptor, and the *xp\_port* field shall be the transport's port number.

##### Return Value

Upon successful completion, svcudp\_create() shall return a pointer to a RPC service transport; otherwise, a null pointer shall be returned.

#### swscanf

##### Name

swscanf — convert formatted input

##### Description

The scanf() family of functions shall behave as described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except as noted below.

##### Differences

The %s, %S and %[ conversion specifiers shall accept an option length modifier a, which shall cause a memory buffer to be allocated to hold the string converted. In such a case, the argument corresponding to the conversion specifier should be a reference to a pointer value that will receive a pointer to the allocated buffer. If there is insufficient memory to allocate a buffer, the function may set errno to ENOMEM and a conversion error results.

**Note:** This directly conflicts with the [ISO C (1999)](#ID_STD_46_ISOC99) usage of %a as a conversion specifier for hexadecimal float values. While this conversion specifier should be supported, a format specifier such as "%aseconds" will have a different meaning on an LSB conforming system.

#### sysconf

##### Name

sysconf — Get configuration information at runtime

##### Synopsis

#include <unistd.h>

long sysconf(int *name*);

##### Description

sysconf() is as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but with differences as listed below.

Extra Variables

These additional values extend the list in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

- *\_SC\_PHYS\_PAGES*

  The number of pages of physical memory.

- *\_SC\_AVPHYS\_PAGES*

  The number of currently available pages of physical memory.

- *\_SC\_NPROCESSORS\_CONF*

  The number of processors configured.

- *\_SC\_NPROCESSORS\_ONLN*

  The number of processors currently online (available).

Extra Versions

While this specification only requires conformance with [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), implementations are not constrained from moving on and claiming conformance with a subsequent edition, [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4). Thus for run-time checks using sysconf(), the wording is amended to allow return values of 0, -1, 200112L or 200809L where formerly 200809L was not listed as allowed.

#### sysinfo

##### Name

sysinfo — return system information

##### Synopsis

#include <sys/sysinfo.h>

int sysinfo(struct sysinfo \**info*);

##### Description

sysinfo() provides a way to obtain certain system statistics. Statistics are written into a sysinfo structure pointed to by *info*. Elements which take a size are sized in units indicated by the value of the *mem\_unit* member of *info*. The other members have traditional meanings as indicated in Data Definitions, but are not formally part of this specification.

##### Return Value

Returns zero on success. On error, -1 is returned and errno is set to indicate the error.

##### Errors

EFAULT

  The *info* parameter does not point to a valid sysinfo structure.

#### system

##### Name

system — execute a shell command

##### Synopsis

#include <stdlib.h>

int system(const char \* *string*);

##### Description

The system() function shall behave as described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

##### Notes

The fact that system() ignores interrupts is often not what a program wants. [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) describes some of the consequences; an additional consequence is that a program calling system() from a loop cannot be reliably interrupted. Many programs will want to use the exec() family of functions instead.

Do not use system() from a program with suid or sgid privileges, because unexpected values for some environment variables might be used to subvert system integrity. Use the exec() family of functions instead, but not execlp() or execvp(). system() will not, in fact, work properly from programs with suid or sgid privileges on systems on which /bin/sh is **bash** version 2, since **bash** 2 drops privileges on startup. (Debian uses a modified **bash** which does not do this when invoked as **sh**.)

The check for the availability of /bin/sh is not actually performed; it is always assumed to be available. [ISO C (1999)](#ID_STD_46_ISOC99) specifies the check, but [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) specifies that the return shall always be nonzero, since a system without the shell is not conforming, and it is this that is implemented.

It is possible for the shell command to return 127, so that code is not a sure indication that the execve() call failed; check the global variable errno to make sure.

#### textdomain

##### Name

textdomain — set the current default message domain

##### Synopsis

#include <libintl.h>

char \* textdomain(const char \* *domainname*);

##### Description

The textdomain() function shall set the current default message domain to *domainname*. Subsequent calls to gettext() and ngettext() use the default message domain.

If *domainname* is NULL, the default message domain shall not be altered.

If *domainname* is "", textdomain() shall reset the default domain to the system default of "messages".

##### Return

On success, textdomain() shall return the currently selected domain. Otherwise, a null pointer shall be returned, and errno is set to indicate the error.

##### Errors

ENOMEM

  Insufficent memory available.

#### unlink

##### Name

unlink — remove a directory entry

##### Synopsis

int unlink(const char \* *path*);

##### Description

unlink() is as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but with differences as listed below.

See also [Section 18.1, Additional behaviors: unlink/link on directory](#ID_EXEC_45_LINK_45_UNLINK).

#### May return EISDIR on directories

If *path* specifies a directory, the implementation may return EISDIR instead of EPERM as specified by [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

**Rationale:** The Linux kernel has deliberately chosen EISDIR for this case and does not expect to change.

#### utmpname

##### Name

utmpname — set user accounting database

##### Synopsis

#include <utmp.h>

int utmpname(const char \* *dbname*);

##### Description

The utmpname() function shall cause the user accounting database used by the getutent(), getutent\_r(), getutxent(), getutxid(), getutxline(), and pututxline() functions to be that named by *dbname*, instead of the system default database. See [Section 18.3](#ID_FHS_45_USER_45_ACCOUNTING_45_DBS) for further information.

**Note:** The LSB does not specify the format of the user accounting database, nor the names of the file or files that may contain it.

##### Return Value

None.

##### Errors

None defined.

#### vasprintf

##### Name

vasprintf — write formatted output to a dynamically allocated string

##### Synopsis

#include <stdarg.h>

#include <stdio.h>

int vasprintf(char \* \* restrict *ptr*, const char \* restrict *format*, va\_list *arg*);

##### Description

The vasprintf() function shall write formatted output to a dynamically allocated string, and store the address of that string in the location referenced by *ptr*. It shall behave as asprintf(), except that instead of being called with a variable number of arguments, it is called with an argument list as defined by <stdarg.h>.

##### Return Value

Refer to fprintf().

##### Errors

Refer to fprintf().

#### verrx

##### Name

verrx — display formatted error message and exit

##### Synopsis

#include <stdarg.h>

#include <err.h>

void verrx (int *eval* , const char \* *fmt* , va\_list *args* );

##### Description

The verrx() shall behave as errx() except that instead of being called with a variable number of arguments, it is called with an argument list as defined by <stdarg.h>.

verrx() does not return, but exits with the value of *eval*.

##### Return Value

None.

##### Errors

None.

#### vfscanf

##### Name

vfscanf — convert formatted input

##### Description

The scanf() family of functions shall behave as described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except as noted below.

##### Differences

The %s, %S and %[ conversion specifiers shall accept an option length modifier a, which shall cause a memory buffer to be allocated to hold the string converted. In such a case, the argument corresponding to the conversion specifier should be a reference to a pointer value that will receive a pointer to the allocated buffer. If there is insufficient memory to allocate a buffer, the function may set errno to ENOMEM and a conversion error results.

**Note:** This directly conflicts with the [ISO C (1999)](#ID_STD_46_ISOC99) usage of %a as a conversion specifier for hexadecimal float values. While this conversion specifier should be supported, a format specifier such as "%aseconds" will have a different meaning on an LSB conforming system.

#### vfwscanf

##### Name

vfwscanf — convert formatted input

##### Description

The scanf() family of functions shall behave as described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except as noted below.

##### Differences

The %s, %S and %[ conversion specifiers shall accept an option length modifier a, which shall cause a memory buffer to be allocated to hold the string converted. In such a case, the argument corresponding to the conversion specifier should be a reference to a pointer value that will receive a pointer to the allocated buffer. If there is insufficient memory to allocate a buffer, the function may set errno to ENOMEM and a conversion error results.

**Note:** This directly conflicts with the [ISO C (1999)](#ID_STD_46_ISOC99) usage of %a as a conversion specifier for hexadecimal float values. While this conversion specifier should be supported, a format specifier such as "%aseconds" will have a different meaning on an LSB conforming system.

#### vscanf

##### Name

vscanf — convert formatted input

##### Description

The scanf() family of functions shall behave as described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except as noted below.

##### Differences

The %s, %S and %[ conversion specifiers shall accept an option length modifier a, which shall cause a memory buffer to be allocated to hold the string converted. In such a case, the argument corresponding to the conversion specifier should be a reference to a pointer value that will receive a pointer to the allocated buffer. If there is insufficient memory to allocate a buffer, the function may set errno to ENOMEM and a conversion error results.

**Note:** This directly conflicts with the [ISO C (1999)](#ID_STD_46_ISOC99) usage of %a as a conversion specifier for hexadecimal float values. While this conversion specifier should be supported, a format specifier such as "%aseconds" will have a different meaning on an LSB conforming system.

#### vsscanf

##### Name

vsscanf — convert formatted input

##### Description

The scanf() family of functions shall behave as described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except as noted below.

##### Differences

The %s, %S and %[ conversion specifiers shall accept an option length modifier a, which shall cause a memory buffer to be allocated to hold the string converted. In such a case, the argument corresponding to the conversion specifier should be a reference to a pointer value that will receive a pointer to the allocated buffer. If there is insufficient memory to allocate a buffer, the function may set errno to ENOMEM and a conversion error results.

**Note:** This directly conflicts with the [ISO C (1999)](#ID_STD_46_ISOC99) usage of %a as a conversion specifier for hexadecimal float values. While this conversion specifier should be supported, a format specifier such as "%aseconds" will have a different meaning on an LSB conforming system.

#### vswscanf

##### Name

vswscanf — convert formatted input

##### Description

The scanf() family of functions shall behave as described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except as noted below.

##### Differences

The %s, %S and %[ conversion specifiers shall accept an option length modifier a, which shall cause a memory buffer to be allocated to hold the string converted. In such a case, the argument corresponding to the conversion specifier should be a reference to a pointer value that will receive a pointer to the allocated buffer. If there is insufficient memory to allocate a buffer, the function may set errno to ENOMEM and a conversion error results.

**Note:** This directly conflicts with the [ISO C (1999)](#ID_STD_46_ISOC99) usage of %a as a conversion specifier for hexadecimal float values. While this conversion specifier should be supported, a format specifier such as "%aseconds" will have a different meaning on an LSB conforming system.

#### vsyslog

##### Name

vsyslog — log to system log

##### Synopsis

#include <stdarg.h>

#include <syslog.h>

void vsyslog(int *priority*, char \* *message*, va\_list *arglist*);

##### Description

The vsyslog() function is identical to syslog() as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except that *arglist* (as defined by stdarg.h) replaces the variable number of arguments.

#### vwscanf

##### Name

vwscanf — convert formatted input

##### Description

The scanf() family of functions shall behave as described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except as noted below.

##### Differences

The %s, %S and %[ conversion specifiers shall accept an option length modifier a, which shall cause a memory buffer to be allocated to hold the string converted. In such a case, the argument corresponding to the conversion specifier should be a reference to a pointer value that will receive a pointer to the allocated buffer. If there is insufficient memory to allocate a buffer, the function may set errno to ENOMEM and a conversion error results.

**Note:** This directly conflicts with the [ISO C (1999)](#ID_STD_46_ISOC99) usage of %a as a conversion specifier for hexadecimal float values. While this conversion specifier should be supported, a format specifier such as "%aseconds" will have a different meaning on an LSB conforming system.

#### wait4

##### Name

wait4 — wait for process termination, BSD style

##### Synopsis

#include <sys/types.h>

#include <sys/resource.h>

#include <sys/wait.h>

pid\_t wait4(pid\_t *pid*, int \* *status*, int *options*, struct rusage \* *rusage*);

**Description**

wait4() suspends execution of the current process until a child (as specified by *pid*) has exited, or until a signal is delivered whose action is to terminate the current process or to call a signal handling function. If a child (as requested by *pid*) has already exited by the time of the call (a so-called "zombie" process), the function returns immediately. Any system resources used by the child are freed.

The value of *pid* can be one of

< -1

 wait for any child process whose process group ID is equal to the absolute value of *pid*.

-1

wait for any child process; this is equivalent to calling wait3().

0

  wait for any child process whose process group ID is equal to that of the calling process.

> 0

  wait for the child whose process ID is equal to the value of *pid*.

The value of options is a bitwise or of zero or more of the following constants:

WNOHANG

  return immediately if no child is there to be waited for.

WUNTRACED

  return for children that are stopped, and whose status has not been reported.

If status is not NULL, wait4() stores status information in the location *status*. This status can be evaluated with the following macros:

**Note:** These macros take the status value (an int) as an argument -- not a pointer to the value!

WIFEXITED(status)

  is nonzero if the child exited normally.

WEXITSTATUS(status)

  evaluates to the least significant eight bits of the return code of the child that terminated, which may have been set as the argument to a call to exit() or as the argument for a return statement in the main program. This macro can only be evaluated if WIFEXITED() returned nonzero.

WIFSIGNALED(status)

  returns true if the child process exited because of a signal that was not caught.

WTERMSIG(status)

  returns the number of the signal that caused the child process to terminate. This macro can only be evaluated if WIFSIGNALED() returned nonzero.

WIFSTOPPED(status)

  returns true if the child process that caused the return is currently stopped; this is only possible if the call was done using WUNTRACED().

WSTOPSIG(status)

  returns the number of the signal that caused the child to stop. This macro can only be evaluated if WIFSTOPPED() returned nonzero.

If *rusage* is not NULL, the struct rusage (as defined in sys/resource.h) that it points to will be filled with accounting information. See getrusage() for details.

##### Return Value

On success, the process ID of the child that exited is returned. On error, -1 is returned (in particular, when no unwaited-for child processes of the specified kind exist), or 0 if WNOHANG() was used and no child was available yet. In the latter two cases, the global variable errno is set appropriately.

##### Errors

ECHILD

  No unwaited-for child process as specified does exist.

ERESTARTSYS

  A WNOHANG() was not set and an unblocked signal or a SIGCHILD was caught. This error is returned by the system call. The library interface is not allowed to return ERESTARTSYS, but will return EINTR.

#### warn

##### Name

warn — formatted error messages

##### Synopsis

#include <err.h>

void warn (const char \* *fmt* , ...);

##### Description

The warn() function shall display a formatted error message on the standard error stream. The output shall consist of the last component of the program name, a colon character, and a space character. If *fmt* is non-NULL, it shall be used as a format string for the printf() family of functions, and the formatted message, a colon character, and a space are written to stderr. Finally, the error message string affiliated with the current value of the global variable errno shall be written to stderr, followed by a newline character.

##### Return Value

None.

##### Errors

None.

#### warnx

##### Name

warnx — formatted error messages

##### Synopsis

#include <err.h>

void warnx (const char \* *fmt* , ...);

##### Description

The warnx() function shall display a formatted error message on the standard error stream. The last component of the program name, a colon character, and a space shall be output. If *fmt* is non-NULL, it shall be used as the format string for the printf() family of functions, and the formatted error message, a colon character, and a space shall be output. The output shall be followed by a newline character.

##### Return Value

None.

##### Errors

None.

#### wcstoq

##### Name

wcstoq — convert wide string to long long int representation

##### Synopsis

#include <wchar.h>

long long int wcstoq(const wchar\_t \* restrict *nptr*, wchar\_t \*\* restrict *endptr*, int *base*);

##### Description

The wcstoq() function shall convert the initial portion of the wide string *nptr* to long long int representation. It is identical to wcstoll().

##### Return Value

Refer to wcstoll().

##### Errors

Refer to wcstoll().

#### wcstouq

##### Name

wcstouq — convert wide string to unsigned long long int representation

##### Synopsis

#include <wchar.h>

unsigned long long wcstouq(const wchar\_t \* restrict *nptr*, wchar\_t \*\* restrict *endptr*, int *base*);

##### Description

The wcstouq() function shall convert the initial portion of the wide string *nptr* to unsigned long long int representation. It is identical to wcstoull().

##### Return Value

Refer to wcstoull().

##### Errors

Refer to wcstoull().

#### wscanf

##### Name

wscanf — convert formatted input

##### Description

The scanf() family of functions shall behave as described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except as noted below.

##### Differences

The %s, %S and %[ conversion specifiers shall accept an option length modifier a, which shall cause a memory buffer to be allocated to hold the string converted. In such a case, the argument corresponding to the conversion specifier should be a reference to a pointer value that will receive a pointer to the allocated buffer. If there is insufficient memory to allocate a buffer, the function may set errno to ENOMEM and a conversion error results.

**Note:** This directly conflicts with the [ISO C (1999)](#ID_STD_46_ISOC99) usage of %a as a conversion specifier for hexadecimal float values. While this conversion specifier should be supported, a format specifier such as "%aseconds" will have a different meaning on an LSB conforming system.

#### xdr\_u\_int

##### Name

xdr\_u\_int — library routines for external data representation

##### Synopsis

int xdr\_u\_int(XDR \* *xdrs*, unsigned int \* *up*);

##### Description

xdr\_u\_int() is a filter primitive that translates between C unsigned integers and their external representations.

##### Return Value

On success, 1 is returned. On error, 0 is returned.

#### xdrstdio\_create

##### Name

xdrstdio\_create — library routines for external data representation

##### Synopsis

#include <rpc/xdr.h>

void xdrstdio\_create(XDR \* *xdrs*, FILE \* *file*, enum xdr\_op *op*);

##### Description

The xdrstdio\_create() function shall initialize the XDR stream object referred to by *xdrs*. The XDR stream data shall be written to, or read from, the standard I/O stream associated with *file*. If the operation *op* is XDR\_ENCODE, encoded data shall be written to *file*. If *op* is XDR\_DECODE, encoded data shall be read from *file*. If *op* is XDR\_FREE, the XDR stream object may be used to deallocate storage allocated by a previous XDR\_DECODE.

The associated destroy function shall flush the *file* I/O stream, but not close it.

##### Return Value

None.

## **14.6 Interfaces for libm**

[Table 14-38](#ID_LIB_45_LIBM_45_DEF) defines the library name and shared object name for the libm library

**Table 14-38 libm Definition**

|  |  |
| --- | --- |
| Library: | libm |
| SONAME: | See architecture specific part. |

The behavior of the interfaces in this library is specified by the following specifications:

|  |
| --- |
| [LSB] [This Specification](#ID_STD_46_LSB) |
| [SUSv3] [POSIX 1003.1-2001 (ISO/IEC 9945-2003)](#ID_STD_46_SUSV3) |
| [SUSv4] [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) |

### **14.6.1 Math**

#### 14.6.1.1 Interfaces for Math

An LSB conforming implementation shall provide the generic functions for Math specified in [Table 14-39](#ID_TBL_45_LIBM_45_MATH_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-39 libm - Math Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| \_\_finite [[LSB]](#ID_REFSTD_46_LIBM_46_1) | \_\_finitef [[LSB]](#ID_REFSTD_46_LIBM_46_1) | \_\_finitel [[LSB]](#ID_REFSTD_46_LIBM_46_1) | \_\_fpclassify [[LSB]](#ID_REFSTD_46_LIBM_46_1) |
| \_\_fpclassifyf [[LSB]](#ID_REFSTD_46_LIBM_46_1) | \_\_signbit [[LSB]](#ID_REFSTD_46_LIBM_46_1) | \_\_signbitf [[LSB]](#ID_REFSTD_46_LIBM_46_1) | acos [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| acosf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | acosh [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | acoshf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | acoshl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| acosl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | asin [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | asinf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | asinh [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| asinhf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | asinhl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | asinl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | atan [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| atan2 [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | atan2f [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | atan2l [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | atanf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| atanh [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | atanhf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | atanhl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | atanl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| cabs [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | cabsf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | cabsl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | cacos [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| cacosf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | cacosh [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | cacoshf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | cacoshl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| cacosl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | carg [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | cargf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | cargl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| casin [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | casinf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | casinh [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | casinhf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| casinhl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | casinl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | catan [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | catanf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| catanh [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | catanhf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | catanhl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | catanl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| cbrt [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | cbrtf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | cbrtl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | ccos [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| ccosf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | ccosh [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | ccoshf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | ccoshl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| ccosl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | ceil [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | ceilf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | ceill [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| cexp [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | cexpf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | cexpl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | cimag [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| cimagf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | cimagl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | clog [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | clog10 [[LSB]](#ID_REFSTD_46_LIBM_46_1) |
| clog10f [[LSB]](#ID_REFSTD_46_LIBM_46_1) | clog10l [[LSB]](#ID_REFSTD_46_LIBM_46_1) | clogf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | clogl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| conj [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | conjf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | conjl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | copysign [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| copysignf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | copysignl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | cos [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | cosf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| cosh [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | coshf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | coshl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | cosl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| cpow [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | cpowf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | cpowl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | cproj [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| cprojf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | cprojl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | creal [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | crealf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| creall [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | csin [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | csinf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | csinh [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| csinhf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | csinhl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | csinl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | csqrt [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| csqrtf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | csqrtl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | ctan [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | ctanf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| ctanh [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | ctanhf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | ctanhl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | ctanl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| drem [[LSB]](#ID_REFSTD_46_LIBM_46_1) | dremf [[LSB]](#ID_REFSTD_46_LIBM_46_1) | dreml [[LSB]](#ID_REFSTD_46_LIBM_46_1) | erf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| erfc [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | erfcf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | erfcl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | erff [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| erfl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | exp [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | exp10 [[LSB]](#ID_REFSTD_46_LIBM_46_1) | exp10f [[LSB]](#ID_REFSTD_46_LIBM_46_1) |
| exp10l [[LSB]](#ID_REFSTD_46_LIBM_46_1) | exp2 [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | exp2f [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | expf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| expl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | expm1 [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | expm1f [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | expm1l [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| fabs [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | fabsf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | fabsl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | fdim [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| fdimf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | fdiml [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | feclearexcept [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | fedisableexcept [[LSB]](#ID_REFSTD_46_LIBM_46_1) |
| feenableexcept [[LSB]](#ID_REFSTD_46_LIBM_46_1) | fegetenv [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | fegetexcept [[LSB]](#ID_REFSTD_46_LIBM_46_1) | fegetexceptflag [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| fegetround [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | feholdexcept [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | feraiseexcept [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | fesetenv [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| fesetexceptflag [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | fesetround [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | fetestexcept [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | feupdateenv [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| finite [[LSB]](#ID_REFSTD_46_LIBM_46_1) | finitef [[LSB]](#ID_REFSTD_46_LIBM_46_1) | finitel [[LSB]](#ID_REFSTD_46_LIBM_46_1) | floor [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| floorf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | floorl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | fma [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | fmaf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| fmal [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | fmax [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | fmaxf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | fmaxl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| fmin [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | fminf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | fminl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | fmod [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| fmodf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | fmodl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | frexp [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | frexpf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| frexpl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | gamma [[LSB]](#ID_REFSTD_46_LIBM_46_1) | gammaf [[LSB]](#ID_REFSTD_46_LIBM_46_1) | gammal [[LSB]](#ID_REFSTD_46_LIBM_46_1) |
| hypot [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | hypotf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | hypotl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | ilogb [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| ilogbf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | ilogbl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | j0 [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | j0f [[LSB]](#ID_REFSTD_46_LIBM_46_1) |
| j0l [[LSB]](#ID_REFSTD_46_LIBM_46_1) | j1 [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | j1f [[LSB]](#ID_REFSTD_46_LIBM_46_1) | j1l [[LSB]](#ID_REFSTD_46_LIBM_46_1) |
| jn [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | jnf [[LSB]](#ID_REFSTD_46_LIBM_46_1) | jnl [[LSB]](#ID_REFSTD_46_LIBM_46_1) | ldexp [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| ldexpf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | ldexpl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | lgamma [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | lgamma\_r [[LSB]](#ID_REFSTD_46_LIBM_46_1) |
| lgammaf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | lgammaf\_r [[LSB]](#ID_REFSTD_46_LIBM_46_1) | lgammal [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | lgammal\_r [[LSB]](#ID_REFSTD_46_LIBM_46_1) |
| llrint [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | llrintf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | llrintl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | llround [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| llroundf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | llroundl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | log [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | log10 [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| log10f [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | log10l [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | log1p [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | log1pf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| log1pl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | log2 [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | log2f [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | log2l [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| logb [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | logbf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | logbl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | logf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| logl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | lrint [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | lrintf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | lrintl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| lround [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | lroundf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | lroundl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | matherr [[LSB]](#ID_REFSTD_46_LIBM_46_1) |
| modf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | modff [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | modfl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | nan [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| nanf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | nanl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | nearbyint [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | nearbyintf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| nearbyintl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | nextafter [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | nextafterf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | nextafterl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| nexttoward [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | nexttowardf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | nexttowardl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | pow [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| pow10 [[LSB]](#ID_REFSTD_46_LIBM_46_1) | pow10f [[LSB]](#ID_REFSTD_46_LIBM_46_1) | pow10l [[LSB]](#ID_REFSTD_46_LIBM_46_1) | powf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| powl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | remainder [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | remainderf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | remainderl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| remquo [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | remquof [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | remquol [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | rint [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| rintf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | rintl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | round [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | roundf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| roundl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | scalb [[SUSv3]](#ID_REFSTD_46_LIBM_46_2) | scalbf [[LSB]](#ID_REFSTD_46_LIBM_46_1) | scalbl [[LSB]](#ID_REFSTD_46_LIBM_46_1) |
| scalbln [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | scalblnf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | scalblnl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | scalbn [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| scalbnf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | scalbnl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | significand [[LSB]](#ID_REFSTD_46_LIBM_46_1) | significandf [[LSB]](#ID_REFSTD_46_LIBM_46_1) |
| significandl [[LSB]](#ID_REFSTD_46_LIBM_46_1) | sin [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | sincos [[LSB]](#ID_REFSTD_46_LIBM_46_1) | sincosf [[LSB]](#ID_REFSTD_46_LIBM_46_1) |
| sincosl [[LSB]](#ID_REFSTD_46_LIBM_46_1) | sinf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | sinh [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | sinhf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| sinhl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | sinl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | sqrt [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | sqrtf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| sqrtl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | tan [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | tanf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | tanh [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| tanhf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | tanhl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | tanl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | tgamma [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| tgammaf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | tgammal [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | trunc [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | truncf [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| truncl [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | y0 [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | y0f [[LSB]](#ID_REFSTD_46_LIBM_46_1) | y0l [[LSB]](#ID_REFSTD_46_LIBM_46_1) |
| y1 [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) | y1f [[LSB]](#ID_REFSTD_46_LIBM_46_1) | y1l [[LSB]](#ID_REFSTD_46_LIBM_46_1) | yn [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |
| ynf [[LSB]](#ID_REFSTD_46_LIBM_46_1) | ynl [[LSB]](#ID_REFSTD_46_LIBM_46_1) |  |  |

An LSB conforming implementation shall provide the generic deprecated functions for Math specified in [Table 14-40](#ID_TBL_45_LIBM_45_MATH_45_DEPINTS), with the full mandatory functionality as described in the referenced underlying specification.

**Note:** These interfaces are deprecated, and applications should avoid using them. These interfaces may be withdrawn in future releases of this specification.

**Table 14-40 libm - Math Deprecated Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| drem [[LSB]](#ID_REFSTD_46_LIBM_46_1) | dremf [[LSB]](#ID_REFSTD_46_LIBM_46_1) | dreml [[LSB]](#ID_REFSTD_46_LIBM_46_1) | finite [[LSB]](#ID_REFSTD_46_LIBM_46_1) |
| finitef [[LSB]](#ID_REFSTD_46_LIBM_46_1) | finitel [[LSB]](#ID_REFSTD_46_LIBM_46_1) | gamma [[LSB]](#ID_REFSTD_46_LIBM_46_1) | gammaf [[LSB]](#ID_REFSTD_46_LIBM_46_1) |
| gammal [[LSB]](#ID_REFSTD_46_LIBM_46_1) | matherr [[LSB]](#ID_REFSTD_46_LIBM_46_1) |  |  |

An LSB conforming implementation shall provide the generic data interfaces for Math specified in [Table 14-41](#ID_TBL_45_LIBM_45_MATH_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-41 libm - Math Data Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| signgam [[SUSv4]](#ID_REFSTD_46_LIBM_46_3) |  |  |  |

## **14.7 Data Definitions for libm**

This section defines global identifiers and their values that are associated with interfaces contained in libm. These definitions are organized into groups that correspond to system headers. This convention is used as a convenience for the reader, and does not imply the existence of these headers, or their content. Where an interface is defined as requiring a particular system header file all of the data definitions for that system header file presented here shall be in effect.

This section gives data definitions to promote binary application portability, not to repeat source interface definitions available elsewhere. System providers and application developers should use this ABI to supplement - not to replace - source interface definition specifications.

This specification uses the [ISO C (1999)](#ID_STD_46_ISOC99) C Language as the reference programming language, and data definitions are specified in ISO C format. The C language is used here as a convenient notation. Using a C language description of these data objects does not preclude their use by other programming languages.

### **14.7.1 complex.h**

#define complex \_Complex

extern double cabs(double complex);

extern float cabsf(float complex);

extern long double cabsl(long double complex);

extern double complex cacos(double complex);

extern float complex cacosf(float complex);

extern double complex cacosh(double complex);

extern float complex cacoshf(float complex);

extern long double complex cacoshl(long double complex);

extern long double complex cacosl(long double complex);

extern double carg(double complex);

extern float cargf(float complex);

extern long double cargl(long double complex);

extern double complex casin(double complex);

extern float complex casinf(float complex);

extern double complex casinh(double complex);

extern float complex casinhf(float complex);

extern long double complex casinhl(long double complex);

extern long double complex casinl(long double complex);

extern double complex catan(double complex);

extern float complex catanf(float complex);

extern double complex catanh(double complex);

extern float complex catanhf(float complex);

extern long double complex catanhl(long double complex);

extern long double complex catanl(long double complex);

extern double complex ccos(double complex);

extern float complex ccosf(float complex);

extern double complex ccosh(double complex);

extern float complex ccoshf(float complex);

extern long double complex ccoshl(long double complex);

extern long double complex ccosl(long double complex);

extern double complex cexp(double complex);

extern float complex cexpf(float complex);

extern long double complex cexpl(long double complex);

extern double cimag(double complex);

extern float cimagf(float complex);

extern long double cimagl(long double complex);

extern double complex clog(double complex);

extern double complex clog10(double complex);

extern float complex clog10f(float complex);

extern long double complex clog10l(long double complex);

extern float complex clogf(float complex);

extern long double complex clogl(long double complex);

extern double complex conj(double complex);

extern float complex conjf(float complex);

extern long double complex conjl(long double complex);

extern double complex cpow(double complex, double complex);

extern float complex cpowf(float complex, float complex);

extern long double complex cpowl(long double complex, long double complex);

extern double complex cproj(double complex);

extern float complex cprojf(float complex);

extern long double complex cprojl(long double complex);

extern double creal(double complex);

extern float crealf(float complex);

extern long double creall(long double complex);

extern double complex csin(double complex);

extern float complex csinf(float complex);

extern double complex csinh(double complex);

extern float complex csinhf(float complex);

extern long double complex csinhl(long double complex);

extern long double complex csinl(long double complex);

extern double complex csqrt(double complex);

extern float complex csqrtf(float complex);

extern long double complex csqrtl(long double complex);

extern double complex ctan(double complex);

extern float complex ctanf(float complex);

extern double complex ctanh(double complex);

extern float complex ctanhf(float complex);

extern long double complex ctanhl(long double complex);

extern long double complex ctanl(long double complex);

### **14.7.2 fenv.h**

extern int feclearexcept(int \_\_excepts);

extern int fedisableexcept(int \_\_excepts);

extern int feenableexcept(int \_\_excepts);

extern int fegetenv(fenv\_t \* \_\_envp);

extern int fegetexcept(void);

extern int fegetexceptflag(fexcept\_t \* \_\_flagp, int \_\_excepts);

extern int fegetround(void);

extern int feholdexcept(fenv\_t \* \_\_envp);

extern int feraiseexcept(int \_\_excepts);

extern int fesetenv(const fenv\_t \* \_\_envp);

extern int fesetexceptflag(const fexcept\_t \* \_\_flagp, int \_\_excepts);

extern int fesetround(int \_\_rounding\_direction);

extern int fetestexcept(int \_\_excepts);

extern int feupdateenv(const fenv\_t \* \_\_envp);

### **14.7.3 math.h**

#define DOMAIN 1

#define SING 2

#define FP\_NAN 0

#define FP\_INFINITE 1

#define FP\_ZERO 2

#define FP\_SUBNORMAL 3

#define FP\_NORMAL 4

#define isnormal(x) (fpclassify (x) == FP\_NORMAL) /\* Return nonzero value if X is neither zero, subnormal, Inf, n \*/

#define HUGE\_VAL 0x1.0p2047

#define HUGE\_VALF 0x1.0p255f

#define NAN ((float)0x7fc00000UL)

#define M\_1\_PI 0.31830988618379067154

#define M\_LOG10E 0.43429448190325182765

#define M\_2\_PI 0.63661977236758134308

#define M\_LN2 0.69314718055994530942

#define M\_SQRT1\_2 0.70710678118654752440

#define M\_PI\_4 0.78539816339744830962

#define M\_2\_SQRTPI 1.12837916709551257390

#define M\_SQRT2 1.41421356237309504880

#define M\_LOG2E 1.4426950408889634074

#define M\_PI\_2 1.57079632679489661923

#define M\_LN10 2.30258509299404568402

#define M\_E 2.7182818284590452354

#define M\_PI 3.14159265358979323846

#define INFINITY HUGE\_VALF

#define MATH\_ERRNO 1 /\* errno set by math functions. \*/

#define MATH\_ERREXCEPT 2 /\* Exceptions raised by math functions. \*/

#define isunordered(u, v) \

(\_\_extension\_\_({ \_\_typeof\_\_(u) \_\_u = (u); \_\_typeof\_\_(v) \_\_v = (v);fpclassify (\_\_u) == FP\_NAN || fpclassify (\_\_v) == FP\_NAN; })) /\* Return nonzero value if arguments are unordered. \*/

#define islessgreater(x, y) \

(\_\_extension\_\_({ \_\_typeof\_\_(x) \_\_x = (x); \_\_typeof\_\_(y) \_\_y = (y);!isunordered (\_\_x, \_\_y) && (\_\_x < \_\_y || \_\_y < \_\_x); })) /\* Return nonzero value if either X is less than Y or Y is less \*/

#define isless(x,y) \

(\_\_extension\_\_({ \_\_typeof\_\_(x) \_\_x = (x); \_\_typeof\_\_(y) \_\_y = (y);!isunordered (\_\_x, \_\_y) && \_\_x < \_\_y; })) /\* Return nonzero value if X is less than Y. \*/

#define islessequal(x, y) \

(\_\_extension\_\_({ \_\_typeof\_\_(x) \_\_x = (x); \_\_typeof\_\_(y) \_\_y = (y);!isunordered (\_\_x, \_\_y) && \_\_x <= \_\_y; })) /\* Return nonzero value if X is less than or equal to Y. \*/

#define isgreater(x,y) \

(\_\_extension\_\_({ \_\_typeof\_\_(x) \_\_x = (x); \_\_typeof\_\_(y) \_\_y = (y);!isunordered (\_\_x, \_\_y) && \_\_x > \_\_y; })) /\* Return nonzero value if X is greater than Y. \*/

#define isgreaterequal(x,y) \

(\_\_extension\_\_({ \_\_typeof\_\_(x) \_\_x = (x); \_\_typeof\_\_(y) \_\_y = (y);!isunordered (\_\_x, \_\_y) && \_\_x >= \_\_y; })) /\* Return nonzero value if X is greater than or equal to Y. \*/

extern int \_\_finite(double);

extern int \_\_finitef(float);

extern int \_\_finitel(long double);

extern int \_\_fpclassify(double);

extern int \_\_fpclassifyf(float);

extern int \_\_isinf(double);

extern int \_\_isinff(float);

extern int \_\_isinfl(long double);

extern int \_\_isnan(double);

extern int \_\_isnanf(float);

extern int \_\_isnanl(long double);

extern int \_\_signbit(double);

extern int \_\_signbitf(float);

extern double acos(double);

extern float acosf(float);

extern double acosh(double);

extern float acoshf(float);

extern long double acoshl(long double);

extern long double acosl(long double);

extern double asin(double);

extern float asinf(float);

extern double asinh(double);

extern float asinhf(float);

extern long double asinhl(long double);

extern long double asinl(long double);

extern double atan(double);

extern double atan2(double, double);

extern float atan2f(float, float);

extern long double atan2l(long double, long double);

extern float atanf(float);

extern double atanh(double);

extern float atanhf(float);

extern long double atanhl(long double);

extern long double atanl(long double);

extern double cbrt(double);

extern float cbrtf(float);

extern long double cbrtl(long double);

extern double ceil(double);

extern float ceilf(float);

extern long double ceill(long double);

extern double copysign(double, double);

extern float copysignf(float, float);

extern long double copysignl(long double, long double);

extern double cos(double);

extern float cosf(float);

extern double cosh(double);

extern float coshf(float);

extern long double coshl(long double);

extern long double cosl(long double);

extern double drem(double, double);

extern float dremf(float, float);

extern long double dreml(long double, long double);

extern double erf(double);

extern double erfc(double);

extern float erfcf(float);

extern long double erfcl(long double);

extern float erff(float);

extern long double erfl(long double);

extern double exp(double);

extern double exp10(double);

extern float exp10f(float);

extern long double exp10l(long double);

extern double exp2(double);

extern float exp2f(float);

extern float expf(float);

extern long double expl(long double);

extern double expm1(double);

extern float expm1f(float);

extern long double expm1l(long double);

extern double fabs(double);

extern float fabsf(float);

extern long double fabsl(long double);

extern double fdim(double, double);

extern float fdimf(float, float);

extern long double fdiml(long double, long double);

extern int finite(double);

extern int finitef(float);

extern int finitel(long double);

extern double floor(double);

extern float floorf(float);

extern long double floorl(long double);

extern double fma(double, double, double);

extern float fmaf(float, float, float);

extern long double fmal(long double, long double, long double);

extern double fmax(double, double);

extern float fmaxf(float, float);

extern long double fmaxl(long double, long double);

extern double fmin(double, double);

extern float fminf(float, float);

extern long double fminl(long double, long double);

extern double fmod(double, double);

extern float fmodf(float, float);

extern long double fmodl(long double, long double);

extern double frexp(double, int \*);

extern float frexpf(float, int \*);

extern long double frexpl(long double, int \*);

extern double gamma(double);

extern float gammaf(float);

extern long double gammal(long double);

extern double hypot(double, double);

extern float hypotf(float, float);

extern long double hypotl(long double, long double);

extern int ilogb(double);

extern int ilogbf(float);

extern int ilogbl(long double);

extern double j0(double);

extern float j0f(float);

extern long double j0l(long double);

extern double j1(double);

extern float j1f(float);

extern long double j1l(long double);

extern double jn(int, double);

extern float jnf(int, float);

extern long double jnl(int, long double);

extern double ldexp(double, int);

extern float ldexpf(float, int);

extern long double ldexpl(long double, int);

extern double lgamma(double);

extern double lgamma\_r(double, int \*);

extern float lgammaf(float);

extern float lgammaf\_r(float, int \*);

extern long double lgammal(long double);

extern long double lgammal\_r(long double, int \*);

extern long long int llrint(double);

extern long long int llrintf(float);

extern long long int llrintl(long double);

extern long long int llround(double);

extern long long int llroundf(float);

extern long long int llroundl(long double);

extern double log(double);

extern double log10(double);

extern float log10f(float);

extern long double log10l(long double);

extern double log1p(double);

extern float log1pf(float);

extern long double log1pl(long double);

extern double log2(double);

extern float log2f(float);

extern long double log2l(long double);

extern double logb(double);

extern float logbf(float);

extern long double logbl(long double);

extern float logf(float);

extern long double logl(long double);

extern long int lrint(double);

extern long int lrintf(float);

extern long int lrintl(long double);

extern long int lround(double);

extern long int lroundf(float);

extern long int lroundl(long double);

extern double modf(double, double \*);

extern float modff(float, float \*);

extern long double modfl(long double, long double \*);

extern double nan(const char \*);

extern float nanf(const char \*);

extern long double nanl(const char \*);

extern double nearbyint(double);

extern float nearbyintf(float);

extern long double nearbyintl(long double);

extern double nextafter(double, double);

extern float nextafterf(float, float);

extern long double nextafterl(long double, long double);

extern double nexttoward(double, long double);

extern float nexttowardf(float, long double);

extern long double nexttowardl(long double, long double);

extern double pow(double, double);

extern double pow10(double);

extern float pow10f(float);

extern long double pow10l(long double);

extern float powf(float, float);

extern long double powl(long double, long double);

extern double remainder(double, double);

extern float remainderf(float, float);

extern long double remainderl(long double, long double);

extern double remquo(double, double, int \*);

extern float remquof(float, float, int \*);

extern long double remquol(long double, long double, int \*);

extern double rint(double);

extern float rintf(float);

extern long double rintl(long double);

extern double round(double);

extern float roundf(float);

extern long double roundl(long double);

extern double scalb(double, double);

extern float scalbf(float, float);

extern long double scalbl(long double, long double);

extern double scalbln(double, long int);

extern float scalblnf(float, long int);

extern long double scalblnl(long double, long int);

extern double scalbn(double, int);

extern float scalbnf(float, int);

extern long double scalbnl(long double, int);

extern int signgam;

extern double significand(double);

extern float significandf(float);

extern long double significandl(long double);

extern double sin(double);

extern void sincos(double, double \*, double \*);

extern void sincosf(float, float \*, float \*);

extern void sincosl(long double, long double \*, long double \*);

extern float sinf(float);

extern double sinh(double);

extern float sinhf(float);

extern long double sinhl(long double);

extern long double sinl(long double);

extern double sqrt(double);

extern float sqrtf(float);

extern long double sqrtl(long double);

extern double tan(double);

extern float tanf(float);

extern double tanh(double);

extern float tanhf(float);

extern long double tanhl(long double);

extern long double tanl(long double);

extern double tgamma(double);

extern float tgammaf(float);

extern long double tgammal(long double);

extern double trunc(double);

extern float truncf(float);

extern long double truncl(long double);

extern double y0(double);

extern float y0f(float);

extern long double y0l(long double);

extern double y1(double);

extern float y1f(float);

extern long double y1l(long double);

extern double yn(int, double);

extern float ynf(int, float);

extern long double ynl(int, long double);

## **14.8 Interface Definitions for libm**

The interfaces defined on the following pages are included in libm and are defined by this specification. Unless otherwise noted, these interfaces shall be included in the source standard.

Other interfaces listed in [Section 14.6](#ID_LIBM) shall behave as described in the referenced base document.

#### \_\_finite

##### Name

\_\_finite — test for infinity

##### Synopsis

#include <math.h>

int \_\_finite(double *arg*);

##### Description

\_\_finite() has the same specification as isfinite() in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except that the argument type for \_\_finite() is known to be double.

\_\_finite() is not in the source standard; it is only in the binary standard.

#### \_\_finitef

##### Name

\_\_finitef — test for infinity

##### Synopsis

#include <math.h>  
int \_\_finitef(float *arg*);

##### Description

\_\_finitef() has the same specification as isfinite() in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) except that the argument type for \_\_finitef() is known to be float.

\_\_finitef() is not in the source standard; it is only in the binary standard.

#### \_\_finitel

##### Name

\_\_finitel — test for infinity

##### Synopsis

#include <math.h>

int \_\_finitel(long double *arg*);

##### Description

\_\_finitel() has the same specification as isfinite() in the [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except that the argument type for \_\_finitel() is known to be long double.

\_\_finitel() is not in the source standard; it is only in the binary standard.

#### \_\_fpclassify

##### Name

\_\_fpclassify — Classify real floating type

##### Synopsis

int \_\_fpclassify(double *arg*);

##### Description

\_\_fpclassify() has the same specification as fpclassify() in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except that the argument type for \_\_fpclassify() is known to be double.

\_\_fpclassify() is not in the source standard; it is only in the binary standard.

#### \_\_fpclassifyf

##### Name

\_\_fpclassifyf — Classify real floating type

##### Synopsis

int \_\_fpclassifyf(float *arg*);

##### Description

\_\_fpclassifyf() has the same specification as fpclassify() in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except that the argument type for \_\_fpclassifyf() is known to be float.

\_\_fpclassifyf() is not in the source standard; it is only in the binary standard.

#### \_\_signbit

##### Name

\_\_signbit — test sign of floating point value

##### Synopsis

#include <math.h>

int \_\_signbit(double *arg*);

##### Description

\_\_signbit() has the same specification as signbit() in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except that the argument type for \_\_signbit() is known to be double.

\_\_signbit() is not in the source standard; it is only in the binary standard.

#### \_\_signbitf

##### Name

\_\_signbitf — test sign of floating point value

##### Synopsis

#include <math.h>

int \_\_signbitf(float *arg*);

##### Description

\_\_signbitf() has the same specification as signbit() in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except that the argument type for \_\_signbitf() is known to be float.

\_\_signbitf() is not in the source standard; it is only in the binary standard.

#### clog10

##### Name

clog10 — Logarithm of a Complex Number

##### Synopsis

#include <complex.h>

double complex clog10(double complex *z*);

##### Description

The clog10() function shall compute the base 10 logarithm of the complex number *z*.

##### Return Value

The clog10() function shall return the base 10 logarithm.

#### clog10f

##### Name

clog10f — Logarithm of a Complex Number

##### Synopsis

#include <complex.h>

float complex clog10f(float complex *z*);

##### Description

The clog10f() function shall compute the base 10 logarithm of the complex number *z*.

##### Return Value

The clog10f() function shall return the base 10 logarithm.

#### clog10l

##### Name

clog10l — Logarithm of a Complex Number

##### Synopsis

#include <complex.h>

long double complex clog10l(long double complex *z*);

##### Description

The clog10l() function shall compute the base 10 logarithm of the complex number *z*.

##### Return Value

The clog10l() function shall return the base 10 logarithm.

#### drem

##### Name

drem — Floating Point Remainder (DEPRECATED)

##### Synopsis

#include <math.h>

double drem(double *x*, double *y*);

##### Description

The drem() function shall return the floating point remainder, *x* REM *y* as required by [IEC 60559/IEEE 754 Floating Point](#ID_STD_46_FLTPNT) in the same way as remainder().

**Note:** This function is included only for backwards compatibility; applications should use remainder() instead.

##### Returns

See remainder().

##### See Also

remainder(), dremf(), dreml()

#### dremf

##### Name

dremf — Floating Point Remainder (DEPRECATED)

##### Synopsis

#include <math.h>

double dremf(double *x*, double *y*);

##### Description

The dremf() function shall return the floating point remainder, *x* REM *y* as required by [IEC 60559/IEEE 754 Floating Point](#ID_STD_46_FLTPNT) in the same way as remainderf().

**Note:** This function is included only for backwards compatibility; applications should use remainderf() instead.

##### Returns

See remainderf().

##### See Also

remainderf(), drem(), dreml()

#### dreml

##### Name

dreml — Floating Point Remainder (DEPRECATED)

##### Synopsis

#include <math.h>

double dreml(double *x*, double *y*);

##### Description

The dreml() function shall return the floating point remainder, *x* REM *y* as required by [IEC 60559/IEEE 754 Floating Point](#ID_STD_46_FLTPNT) in the same way as remainderl().

**Note:** This function is included only for backwards compatibility; applications should use remainderl() instead.

##### Returns

See remainderl().

##### See Also

remainderl(), drem(), dremf()

#### exp10

##### Name

exp10 — Base-10 power function

##### Synopsis

#include <math.h>

double exp10(double *x*);

##### Description

The exp10() function shall return 10x.

**Note:** This function is identical to pow10().

##### Returns

Upon successful completion, exp10() shall return 10 rised to the power of x.

If the correct value would cause overflow, a range error shall occur and exp10() shall return ±HUGE\_VAL, with the same sign as the correct value of the function.

##### See Also

pow10(), exp10f(), exp10l()

#### exp10f

##### Name

exp10f — Base-10 power function

##### Synopsis

#include <math.h>

float exp10f(float *x*);

##### Description

The exp10f() function shall return 10x.

**Note:** This function is identical to pow10f().

##### Returns

Upon successful completion, exp10f() shall return 10 rised to the power of x.

If the correct value would cause overflow, a range error shall occur and exp10f() shall return ±HUGE\_VALF, with the same sign as the correct value of the function.

##### See Also

pow10f(), exp10(), exp10l()

#### exp10l

##### Name

exp10l — Base-10 power function

##### Synopsis

#include <math.h>

long double exp10l(long double *x*);

##### Description

The exp10l() function shall return 10x.

**Note:** This function is identical to pow10l().

##### Returns

Upon successful completion, exp10l() shall return 10 rised to the power of x.

If the correct value would cause overflow, a range error shall occur and exp10l() shall return ±HUGE\_VALL, with the same sign as the correct value of the function.

##### See Also

pow10l(), exp10(), exp10f()

#### fedisableexcept

##### Name

fedisableexcept — disable floating point exceptions

##### Synopsis

#include <fenv.h>

int fedisableexcept(int *excepts*);

##### Description

The fedisableexcept() function disables traps for each of the exceptions represented by the mask excepts.

##### Return Value

The fedisableexcept() function returns the previous set of enabled exceptions on success. On error, -1 is returned.

##### Errors

No errors are defined, but the function will fail if not supported on the architecture.

#### feenableexcept

##### Name

feenableexcept — enable floating point exceptions

##### Synopsis

#include <fenv.h>  
int feenableexcept(int *excepts*);

##### Description

The feenableexcept() function enables traps for each of the exceptions represented by the mask excepts.

##### Return Value

The feenableexcept() function returns the previous set of enabled exceptions on success. On error, -1 is returned.

##### Errors

No errors are defined, but the function will fail if not supported on the architecture.

#### fegetexcept

##### Name

fegetexcept — query floating point exception handling state

##### Synopsis

#include <fenv.h>

int fegetexcept(void);

##### Description

The fegetexcept() function returns the set of all currently enabled exceptions.

##### Return Value

The fegetexcept() function returns the set of all currently enabled exceptions.

##### Errors

No errors are defined, but the function will fail if not supported on the architecture.

#### finite

##### Name

finite — test for infinity (DEPRECATED)

##### Synopsis

#define \_SVID\_SOURCE

#include <math.h>

int finite(double *arg*);

##### Description

The finite() function shall test whether its argument is neither INFINITY nor not a number (NaN).

##### Returns

On success, finite() shall return 1. Otherwise the function shall return 0.

**Note:** The [ISO C (1999)](#ID_STD_46_ISOC99) standard defines the function isfinite(), which is more general purpose. The finite() function is deprecated, and applications should use isfinite() instead. A future revision of this standard may remove this function.

##### See Also

isfinite(), finitef(), finitel()

#### 

#### finitef

##### Name

finitef — test for infinity (DEPRECATED)

##### Synopsis

#define \_SVID\_SOURCE

#include <math.h>

int finitef(float *arg*);

##### Description

The finitef() function shall test whether its argument is neither INFINITY nor not a number (NaN).

##### Returns

On success, finitef() shall return 1. Otherwise the function shall return 0.

**Note:** The [ISO C (1999)](#ID_STD_46_ISOC99) standard defines the function isfinite(), which is more general purpose. The finitef() function is deprecated, and applications should use isfinite() instead. A future revision of this standard may remove this function.

##### See Also

isfinite(), finite(), finitel()

#### finitel

##### Name

finitel — test for infinity (DEPRECATED)

##### Synopsis

#define \_SVID\_SOURCE

#include <math.h>

int finitel(long double *arg*);

##### Description

The finitel() function shall test whether its argument is neither INFINITY nor not a number (NaN).

##### Returns

On success, finitel() shall return 1. Otherwise the function shall return 0.

**Note:** The [ISO C (1999)](#ID_STD_46_ISOC99) standard defines the function isfinite(), which is more general purpose. The finitel() function is deprecated, and applications should use isfinite() instead. A future revision of this standard may remove this function.

##### See Also

isfinite(), finite(), finitef()

#### gamma

##### Name

gamma — log gamma function (DEPRECATED)

##### Synopsis

#include <math.h>

double gammaf(double *x*);

##### Description

The gamma() function is identical to lgamma() in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

**Note:** The name gamma() for this function is deprecated and should not be used.

##### Returns

See lgamma().

##### See Also

lgamma(), lgammaf(), lgammal(), gammaf(), gammal()

#### gammaf

##### Name

gammaf — log gamma function (DEPRECATED)

##### Synopsis

#include <math.h>

float gammaf(float *x*);

##### Description

The gammaf() function is identical to lgammaf() in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

**Note:** The name gammaf() for this function is deprecated and should not be used.

##### Returns

See lgammaf().

##### See Also

lgamma(), lgammaf(), lgammal(), gamma(), gammal()

#### gammal

##### Name

gammal — log gamma function (DEPRECATED)

##### Synopsis

#include <math.h>

long double gammal(long double *x*);

##### Description

The gammal() function is identical to lgammal() in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

**Note:** The name gammal() for this function is deprecated and should not be used.

##### Returns

See lgammal().

##### See Also

lgamma(), lgammaf(), lgammal(), gamma(), gammaf()

#### j0f

##### Name

j0f — Bessel functions

##### Synopsis

#include <math.h>

float j0f(float *x*);

##### Description

The j0f() function is identical to j0(), except that the argument *x* and the return value is a float.

##### Returns

See j0().

##### See Also

j0(), j0l(), j1(), j1f(), j1l(), jn(), jnf(), jnl(), y0(), y0f(), y0l(), y1(), y1f(), y1l(), yn(), ynf(), ynl()

#### j0l

##### Name

j0l — Bessel functions

##### Synopsis

#include <math.h>

long double j0l(long double *x*);

##### Description

The j0l() function is identical to j0(), except that the argument *x* and the return value is a long double.

##### Returns

See j0().

##### See Also

j0(), j0f(), j1(), j1f(), j1l(), jn(), jnf(), jnl(), y0(), y0f(), y0l(), y1(), y1f(), y1l(), yn(), ynf(), ynl()

#### j1f

##### Name

j1f — Bessel functions

##### Synopsis

#include <math.h>

float j1f(float *x*);

##### Description

The j1f() function is identical to j1(), except that the argument *x* and the return value is a float.

##### Returns

See j1().

##### See Also

j0(), j0f(), j0l(), j1(), j1l(), jn(), jnf(), jnl(), y0(), y0f(), y0l(), y1(), y1f(), y1l(), yn(), ynf(), ynl()

#### j1l

##### Name

j1l — Bessel functions

##### Synopsis

#include <math.h>

long double j1l(long double *x*);

##### Description

The j1l() function is identical to j1(), except that the argument *x* and the return value is a long double.

##### Returns

See j0().

##### See Also

j0(), j0f(), j0l(), j1(), j1f(), jn(), jnf(), jnl(), y0(), y0f(), y0l(), y1(), y1f(), y1l(), yn(), ynf(), ynl()

#### jnf

##### Name

jnf — Bessel functions

##### Synopsis

#include <math.h>

float jnf(float *x*);

##### Description

The jnf() function is identical to jn(), except that the argument *x* and the return value is a float.

##### Returns

See jn().

##### See Also

j0(), j0f(), j0l(), j1(), j1f(), j1l(), jn(), jnl(), y0(), y0f(), y0l(), y1(), y1f(), y1l(), yn(), ynf(), ynl()

#### jnl

##### Name

jnl — Bessel functions

##### Synopsis

#include <math.h>

long double jnl(long double *x*);

##### Description

The jnl() function is identical to jn(), except that the argument *x* and the return value is a long double.

##### Returns

See jn().

##### See Also

j0(), j0f(), j0l(), j1(), j1f(), j1l(), jn(), jnf(), y0(), y0f(), y0l(), y1(), y1f(), y1l(), yn(), ynf(), ynl()

#### lgamma\_r

##### Name

lgamma\_r — log gamma functions

##### Synopsis

#include <math.h>

double lgamma\_r(double *x*, int \* *signp*);

##### Description

The lgamma\_r() function shall compute the natural logarithm of the absolute value of the Gamma function, as lgamma(). However, instead of setting the external integer signgam to the sign of the Gamma function, lgamma\_r() shall set the integer referenced by *signp* to the sign.

##### Returns

See lgamma() and signgam.

##### See Also

lgamma(), lgammaf\_r(), lgammal\_r(), signgam

#### lgammaf\_r

##### Name

lgammaf\_r — log gamma functions

##### Synopsis

#include <math.h>

float lgammaf\_r(float *x*, int \* *signp*);

##### Description

The lgammaf\_r() function shall compute the natural logarithm of the absolute value of the Gamma function, as lgammaf(). However, instead of setting the external integer signgam to the sign of the Gamma function, lgammaf\_r() shall set the integer referenced by *signp* to the sign.

##### Returns

See lgammaf() and signgam.

##### See Also

lgamma(), lgamma\_r(), lgammal\_r(), signgam

#### lgammal\_r

##### Name

lgammal\_r — log gamma functions

##### Synopsis

#include <math.h>

double lgammal\_r(double *x*, int \* *signp*);

##### Description

The lgammal\_r() function shall compute the natural logarithm of the absolute value of the Gamma function, as lgammal(). However, instead of setting the external integer signgam to the sign of the Gamma function, lgammal\_r() shall set the integer referenced by *signp* to the sign.

##### Returns

See lgammal() and signgam.

##### See Also

lgamma(), lgamma\_r(), lgammaf\_r(), signgam

#### matherr

##### Name

matherr — math library exception handling

##### Synopsis

#include <math.h>

int matherr(struct exception \**\_\_exc*);

##### Description

The System V Interface Definition (SVID) Issue 3 specifies that various math functions should invoke a function called matherr() if a math exception is detected. This function is called before the math function returns; after matherr() returns, the system then returns to the math function, which in turn returns to the caller.

matherr() is obsolete; indeed it was withdrawn in the System V Interface Definition (SVID) Issue 4, and is required only by this specification for historical compatibility, and will be removed in a future version. The floating point environment function group including fesetenv() should be used instead.

matherr() is not in the source standard; it is only in the binary standard.

##### See Also

fesetenv(), fegetenv(), feupdateenv().

#### pow10

##### Name

pow10 — Base-10 power function

##### Synopsis

#include <math.h>

double pow10(double *x*);

##### Description

The pow10() function shall return 10x.

**Note:** This function is identical to exp10().

##### Returns

Upon successful completion, pow10() shall return 10 rised to the power of x.

If the correct value would cause overflow, a range error shall occur and pow10() shall return ±HUGE\_VAL, with the same sign as the correct value of the function.

##### See Also

exp10(), pow10f(), pow10l()

#### pow10f

##### Name

pow10f — Base-10 power function

##### Synopsis

#include <math.h>

float pow10f(float *x*);

##### Description

The pow10f() function shall return 10x.

**Note:** This function is identical to exp10f().

##### Returns

Upon successful completion, pow10f() shall return 10 rised to the power of x.

If the correct value would cause overflow, a range error shall occur and pow10f() shall return ±HUGE\_VALF, with the same sign as the correct value of the function.

##### See Also

exp10f(), pow10(), pow10l()

#### pow10l

##### Name

pow10l — Base-10 power function

##### Synopsis

#include <math.h>

long double pow10l(long double *x*);

##### Description

The pow10l() function shall return 10x.

**Note:** This function is identical to exp10l().

##### Returns

Upon successful completion, pow10l() shall return 10 rised to the power of x.

If the correct value would cause overflow, a range error shall occur and pow10l() shall return ±HUGE\_VALL, with the same sign as the correct value of the function.

##### See Also

exp10l(), pow10(), pow10f()

#### scalbf

##### Name

scalbf — load exponent of radix-independent floating point number

##### Synopsis

#include <math.h>

float scalbf(float *x*, double *exp*);

##### Description

The scalbf() function is identical to scalb(), except that the argument *x* and the return value is of type float.

##### Returns

See scalb().

#### scalbl

##### Name

scalbl — load exponent of radix-independent floating point number

##### Synopsis

#include <math.h>

long double scalbl(long double *x*, double *exp*);

##### Description

The scalbl() function is identical to scalb(), except that the argument *x* and the return value is of type long double.

##### Returns

See scalb().

#### significand

##### Name

significand — floating point mantissa

##### Synopsis

#include <math.h>

double significand(double *x*);

##### Description

The significand() function shall return the mantissa of *x*, sig such that *x*  sig × 2n scaled such that 1  sig < 2.

**Note:** This function is intended for testing conformance to [IEC 60559/IEEE 754 Floating Point](#ID_STD_46_FLTPNT), and its use is not otherwise recommended.

This function is equivalent to scalb(x, (double)-ilogb(x)).

##### Returns

Upon successful completion, significand() shall return the mantissa of *x* in the range 1  sig < 2.

If *x* is 0, ±HUGE\_VAL, or NaN, the result is undefined.

##### See Also

significandf(), significandl()

#### significandf

##### Name

significandf — floating point mantissa

##### Synopsis

#include <math.h>

float significandf(float *x*);

##### Description

The significandf() function shall return the mantissa of *x*, sig such that *x*  sig × 2n scaled such that 1  sig < 2.

**Note:** This function is intended for testing conformance to [IEC 60559/IEEE 754 Floating Point](#ID_STD_46_FLTPNT), and its use is not otherwise recommended.

This function is equivalent to scalb(x, (double)-ilogb(x)).

##### Returns

Upon successful completion, significandf() shall return the mantissa of *x* in the range 1  sig < 2.

If *x* is 0, ±HUGE\_VALF, or NaN, the result is undefined.

##### See Also

significand(), significandl()

#### significandl

##### Name

significandl — floating point mantissa

##### Synopsis

#include <math.h>

long double significandl(long double *x*);

##### Description

The significandl() function shall return the mantissa of *x*, sig such that *x*  sig × 2n scaled such that 1  sig < 2.

**Note:** This function is intended for testing conformance to [IEC 60559/IEEE 754 Floating Point](#ID_STD_46_FLTPNT), and its use is not otherwise recommended.

This function is equivalent to scalb(x, (double)-ilogb(x)).

##### Returns

Upon successful completion, significandl() shall return the mantissa of *x* in the range 1  sig < 2.

If *x* is 0, ±HUGE\_VALL, or NaN, the result is undefined.

##### See Also

significand(), significandf()

#### sincos

##### Name

sincos — trigonometric functions

##### Synopsis

#define \_GNU\_SOURCE

#include <math.h>

void sincos(double *x*, double \* *sin*, double \* *cos*);

##### Description

The sincos() function shall calculate both the sine and cosine of *x*. The sine shall be stored in the location referenced by *sin*, and the cosine in the location referenced by *cosine*.

##### Returns

None. See sin() and cos() for possible error conditions.

##### See Also

cos(), sin(), sincosf(), sincosl()

#### sincosf

##### Name

sincosf — trigonometric functions

##### Synopsis

#define \_GNU\_SOURCE

#include <math.h>

void sincosf(float *x*, float \* *sin*, float \* *cos*);

##### Description

The sincosf() function shall calculate both the sine and cosine of *x*. The sine shall be stored in the location referenced by *sin*, and the cosine in the location referenced by *cosine*.

##### Returns

None. See sin() and cos() for possible error conditions.

##### See Also

cos(), sin(), sincos(), sincosl()

#### sincosl

##### Name

sincosl — trigonometric functions

##### Synopsis

#define \_GNU\_SOURCE

#include <math.h>

void sincosl(long double *x*, long double \* *sin*, long double \* *cos*);

##### Description

The sincosl() function shall calculate both the sine and cosine of *x*. The sine shall be stored in the location referenced by *sin*, and the cosine in the location referenced by *cosine*.

##### Returns

None. See sin() and cos() for possible error conditions.

##### See Also

cos(), sin(), sincos(), sincosl()

#### y0f

##### Name

y0f — Bessel functions

##### Synopsis

#include <math.h>

float y0f(float *x*);

##### Description

The y0f() function is identical to y0(), except that the argument *x* and the return value is a float.

##### Returns

See y0().

##### See Also

j0(), j0f(), j0l(), j1(), j1f(), j1l(), jn(), jnf(), jnl(), y0(), y0l(), y1(), y1f(), y1l(), yn(), ynf(), ynl()

#### y0l

##### Name

y0l — Bessel functions

##### Synopsis

#include <math.h>

long double y0l(long double *x*);

##### Description

The y0l() function is identical to y0(), except that the argument *x* and the return value is a long double.

##### Returns

See y0().

##### See Also

j0(), j0f(), j0l(), j1(), j1f(), j1l(), jn(), jnf(), jnl(), y0(), y0f(), y1(), y1f(), y1l(), yn(), ynf(), ynl()

#### y1f

##### Name

y1f — Bessel functions

##### Synopsis

#include <math.h>

float y1f(float *x*);

##### Description

The y1f() function is identical to y1(), except that the argument *x* and the return value is a float.

##### Returns

See y1().

##### See Also

j0(), j0f(), j0l(), j1(), j1f(), j1l(), jn(), jnf(), jnl(), y0(), y0f(), y0l(), y1(), y1l(), yn(), ynf(), ynl()

#### y1l

##### Name

y1l — Bessel functions

##### Synopsis

#include <math.h>

long double y1l(long double *x*);

##### Description

The y1l() function is identical to y1(), except that the argument *x* and the return value is a long double.

##### Returns

See j0().

##### See Also

j0(), j0f(), j0l(), j1(), j1f(), j1l(), jn(), jnf(), jnl(), y0(), y0f(), y0l(), y1(), y1f(), yn(), ynf(), ynl()

#### ynf

##### Name

ynf — Bessel functions

##### Synopsis

#include <math.h>

float ynf(float *x*);

##### Description

The ynf() function is identical to yn(), except that the argument *x* and the return value is a float.

##### Returns

See yn().

##### See Also

j0(), j0f(), j0l(), j1(), j1f(), j1l(), jn(), jnf(), jnl(), y0(), y0f(), y0l(), y1(), y1f(), y1l(), yn(), ynl()

#### ynl

##### Name

ynl — Bessel functions

##### Synopsis

#include <math.h>

long double ynl(long double *x*);

##### Description

The ynl() function is identical to yn(), except that the argument *x* and the return value is a long double.

##### Returns

See yn().

##### See Also

j0(), j0f(), j0l(), j1(), j1f(), j1l(), jn(), jnf(), jnl(), y0(), y0f(), y0l(), y1(), y1f(), y1l(), yn(), ynf()

## **14.9 Interfaces for libpthread**

[Table 14-42](#ID_LIB_45_LIBPTHREAD_45_DEF) defines the library name and shared object name for the libpthread library

**Table 14-42 libpthread Definition**

|  |  |
| --- | --- |
| Library: | libpthread |
| SONAME: | libpthread.so.0 |

The behavior of the interfaces in this library is specified by the following specifications:

|  |
| --- |
| [LFS] [Large File Support](#ID_STD_46_LFS) |
| [LSB] [This Specification](#ID_STD_46_LSB) |
| [SUSv3] [POSIX 1003.1-2001 (ISO/IEC 9945-2003)](#ID_STD_46_SUSV3) |
| [SUSv4] [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) |

### **14.9.1 Realtime Threads**

#### 14.9.1.1 Interfaces for Realtime Threads

An LSB conforming implementation shall provide the generic functions for Realtime Threads specified in [Table 14-43](#ID_TBL_45_LIBPTHREAD_45_REALT_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-43 libpthread - Realtime Threads Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| pthread\_attr\_getinheritsched [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_attr\_getschedpolicy [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_attr\_getscope [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_attr\_setinheritsched [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| pthread\_attr\_setschedpolicy [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_attr\_setscope [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_getschedparam [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_mutex\_getprioceiling(GLIBC\_2.4) [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| pthread\_mutex\_setprioceiling(GLIBC\_2.4) [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_mutexattr\_getprioceiling(GLIBC\_2.4) [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_mutexattr\_getprotocol(GLIBC\_2.4) [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_mutexattr\_setprioceiling(GLIBC\_2.4) [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| pthread\_mutexattr\_setprotocol(GLIBC\_2.4) [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_setschedparam [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_setschedprio(GLIBC\_2.3.4) [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |  |

### **14.9.2 Advanced Realtime Threads**

#### 14.9.2.1 Interfaces for Advanced Realtime Threads

An LSB conforming implementation shall provide the generic functions for Advanced Realtime Threads specified in [Table 14-44](#ID_TBL_45_LIBPTHREAD_45_ADVAN_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-44 libpthread - Advanced Realtime Threads Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| pthread\_barrier\_destroy [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_barrier\_init [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_barrier\_wait [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_barrierattr\_destroy [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| pthread\_barrierattr\_getpshared(GLIBC\_2.3.3) [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_barrierattr\_init [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_barrierattr\_setpshared [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_getcpuclockid [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| pthread\_spin\_destroy [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_spin\_init [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_spin\_lock [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_spin\_trylock [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| pthread\_spin\_unlock [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |  |  |  |

### **14.9.3 Posix Threads**

#### 14.9.3.1 Interfaces for Posix Threads

An LSB conforming implementation shall provide the generic functions for Posix Threads specified in [Table 14-45](#ID_TBL_45_LIBPTHREAD_45_POSIX_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-45 libpthread - Posix Threads Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| \_pthread\_cleanup\_pop [[LSB]](#ID_REFSTD_46_LIBPTHREAD_46_2) | \_pthread\_cleanup\_push [[LSB]](#ID_REFSTD_46_LIBPTHREAD_46_2) | pthread\_attr\_destroy [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_attr\_getdetachstate [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| pthread\_attr\_getguardsize [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_attr\_getschedparam [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_attr\_getstack [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_attr\_getstackaddr [[SUSv3]](#ID_REFSTD_46_LIBPTHREAD_46_3) |
| pthread\_attr\_getstacksize [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_attr\_init [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_attr\_setdetachstate [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_attr\_setguardsize [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| pthread\_attr\_setschedparam [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_attr\_setstack [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_attr\_setstackaddr [[SUSv3]](#ID_REFSTD_46_LIBPTHREAD_46_3) | pthread\_attr\_setstacksize [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| pthread\_cancel [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_cond\_broadcast [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_cond\_destroy [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_cond\_init [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| pthread\_cond\_signal [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_cond\_timedwait [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_cond\_wait [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_condattr\_destroy [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| pthread\_condattr\_getclock(GLIBC\_2.3.3) [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_condattr\_getpshared [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_condattr\_init [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_condattr\_setclock(GLIBC\_2.3.3) [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| pthread\_condattr\_setpshared [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_create [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_detach [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_equal [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| pthread\_exit [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_getconcurrency [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_getspecific [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_join [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| pthread\_key\_create [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_key\_delete [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_kill [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_mutex\_consistent(GLIBC\_2.12) [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| pthread\_mutex\_destroy [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_mutex\_init [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_mutex\_lock [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_mutex\_timedlock [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| pthread\_mutex\_trylock [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_mutex\_unlock [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_mutexattr\_destroy [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_mutexattr\_getpshared [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| pthread\_mutexattr\_getrobust(GLIBC\_2.12) [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_mutexattr\_gettype [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_mutexattr\_init [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_mutexattr\_setpshared [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| pthread\_mutexattr\_setrobust(GLIBC\_2.12) [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_mutexattr\_settype [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_once [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_rwlock\_destroy [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| pthread\_rwlock\_init [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_rwlock\_rdlock [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_rwlock\_timedrdlock [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_rwlock\_timedwrlock [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| pthread\_rwlock\_tryrdlock [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_rwlock\_trywrlock [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_rwlock\_unlock [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_rwlock\_wrlock [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| pthread\_rwlockattr\_destroy [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_rwlockattr\_getpshared [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_rwlockattr\_init [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_rwlockattr\_setpshared [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| pthread\_self [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_setcancelstate [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_setcanceltype [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_setconcurrency [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| pthread\_setspecific [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_sigmask [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pthread\_testcancel [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | sem\_close [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| sem\_destroy [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | sem\_getvalue [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | sem\_init [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | sem\_open [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| sem\_post [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | sem\_timedwait [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | sem\_trywait [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | sem\_unlink [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| sem\_wait [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |  |  |  |

An LSB conforming implementation shall provide the generic deprecated functions for Posix Threads specified in [Table 14-46](#ID_TBL_45_LIBPTHREAD_45_POSIX_45_DEPINTS), with the full mandatory functionality as described in the referenced underlying specification.

**Note:** These interfaces are deprecated, and applications should avoid using them. These interfaces may be withdrawn in future releases of this specification.

**Table 14-46 libpthread - Posix Threads Deprecated Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| pthread\_attr\_getstackaddr [[SUSv3]](#ID_REFSTD_46_LIBPTHREAD_46_3) | pthread\_attr\_setstackaddr [[SUSv3]](#ID_REFSTD_46_LIBPTHREAD_46_3) |  |  |

### **14.9.4 Thread aware versions of libc interfaces**

#### 14.9.4.1 Interfaces for Thread aware versions of libc interfaces

An LSB conforming implementation shall provide the generic functions for Thread aware versions of libc interfaces specified in [Table 14-47](#ID_TBL_45_LIBPTHREAD_45_THREA_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-47 libpthread - Thread aware versions of libc interfaces Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| lseek64 [[LFS]](#ID_REFSTD_46_LIBPTHREAD_46_1) | open64 [[LFS]](#ID_REFSTD_46_LIBPTHREAD_46_1) | pread [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pread64 [[LSB]](#ID_REFSTD_46_LIBPTHREAD_46_2) |
| pwrite [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | pwrite64 [[LSB]](#ID_REFSTD_46_LIBPTHREAD_46_2) |  |  |

### **14.9.5 GNU Extensions for libpthread**

#### 14.9.5.1 Interfaces for GNU Extensions for libpthread

An LSB conforming implementation shall provide the generic functions for GNU Extensions for libpthread specified in [Table 14-48](#ID_TBL_45_LIBPTHREAD_45_GNU_45_E_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-48 libpthread - GNU Extensions for libpthread Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| pthread\_getattr\_np [[LSB]](#ID_REFSTD_46_LIBPTHREAD_46_2) | pthread\_mutex\_consistent\_np [[LSB]](#ID_REFSTD_46_LIBPTHREAD_46_2) | pthread\_mutexattr\_getrobust\_np [[LSB]](#ID_REFSTD_46_LIBPTHREAD_46_2) | pthread\_mutexattr\_setrobust\_np [[LSB]](#ID_REFSTD_46_LIBPTHREAD_46_2) |
| pthread\_rwlockattr\_getkind\_np [[LSB]](#ID_REFSTD_46_LIBPTHREAD_46_2) | pthread\_rwlockattr\_setkind\_np [[LSB]](#ID_REFSTD_46_LIBPTHREAD_46_2) |  |  |

### **14.9.6 System Calls**

#### 14.9.6.1 Interfaces for System Calls

An LSB conforming implementation shall provide the generic functions for System Calls specified in [Table 14-49](#ID_TBL_45_LIBPTHREAD_45_SYSTE_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-49 libpthread - System Calls Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| close [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | fcntl [[LSB]](#ID_REFSTD_46_LIBPTHREAD_46_2) | fork [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | fsync [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| lseek [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | msync [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | nanosleep [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | open [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| pause [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | read [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | vfork [[SUSv3]](#ID_REFSTD_46_LIBPTHREAD_46_3) | wait [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| waitpid [[LSB]](#ID_REFSTD_46_LIBPTHREAD_46_2) | write [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |  |  |

### **14.9.7 Standard I/O**

#### 14.9.7.1 Interfaces for Standard I/O

An LSB conforming implementation shall provide the generic functions for Standard I/O specified in [Table 14-50](#ID_TBL_45_LIBPTHREAD_45_STAND_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-50 libpthread - Standard I/O Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| flockfile [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |  |  |  |

### **14.9.8 Signal Handling**

#### 14.9.8.1 Interfaces for Signal Handling

An LSB conforming implementation shall provide the generic functions for Signal Handling specified in [Table 14-51](#ID_TBL_45_LIBPTHREAD_45_SIGNA_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-51 libpthread - Signal Handling Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| \_\_libc\_current\_sigrtmax [[LSB]](#ID_REFSTD_46_LIBPTHREAD_46_2) | \_\_libc\_current\_sigrtmin [[LSB]](#ID_REFSTD_46_LIBPTHREAD_46_2) | raise [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | sigaction [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| siglongjmp [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | sigwait [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |  |  |

### **14.9.9 Standard Library**

#### 14.9.9.1 Interfaces for Standard Library

An LSB conforming implementation shall provide the generic functions for Standard Library specified in [Table 14-52](#ID_TBL_45_LIBPTHREAD_45_STANE_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-52 libpthread - Standard Library Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| \_\_errno\_location [[LSB]](#ID_REFSTD_46_LIBPTHREAD_46_2) | ftrylockfile [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | funlockfile [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | longjmp [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| system [[LSB]](#ID_REFSTD_46_LIBPTHREAD_46_2) |  |  |  |

### **14.9.10 Socket Interface**

#### 14.9.10.1 Interfaces for Socket Interface

An LSB conforming implementation shall provide the generic functions for Socket Interface specified in [Table 14-53](#ID_TBL_45_LIBPTHREAD_45_SOCKE_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-53 libpthread - Socket Interface Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| \_\_h\_errno\_location [[LSB]](#ID_REFSTD_46_LIBPTHREAD_46_2) | accept [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | connect [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | recv [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| recvfrom [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | recvmsg [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | send [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) | sendmsg [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |
| sendto [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |  |  |  |

### **14.9.11 Terminal Interface Functions**

#### 14.9.11.1 Interfaces for Terminal Interface Functions

An LSB conforming implementation shall provide the generic functions for Terminal Interface Functions specified in [Table 14-54](#ID_TBL_45_LIBPTHREAD_45_TERMI_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-54 libpthread - Terminal Interface Functions Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| tcdrain [[SUSv4]](#ID_REFSTD_46_LIBPTHREAD_46_4) |  |  |  |

## **14.10 Data Definitions for libpthread**

This section defines global identifiers and their values that are associated with interfaces contained in libpthread. These definitions are organized into groups that correspond to system headers. This convention is used as a convenience for the reader, and does not imply the existence of these headers, or their content. Where an interface is defined as requiring a particular system header file all of the data definitions for that system header file presented here shall be in effect.

This section gives data definitions to promote binary application portability, not to repeat source interface definitions available elsewhere. System providers and application developers should use this ABI to supplement - not to replace - source interface definition specifications.

This specification uses the [ISO C (1999)](#ID_STD_46_ISOC99) C Language as the reference programming language, and data definitions are specified in ISO C format. The C language is used here as a convenient notation. Using a C language description of these data objects does not preclude their use by other programming languages.

### **14.10.1 lsb/pthread.h**

typedef unsigned long int pthread\_t;

### **14.10.2 pthread.h**

#define PTHREAD\_MUTEX\_DEFAULT 0

#define PTHREAD\_MUTEX\_NORMAL 0

#define PTHREAD\_SCOPE\_SYSTEM 0

#define PTHREAD\_MUTEX\_RECURSIVE 1

#define PTHREAD\_SCOPE\_PROCESS 1

#define PTHREAD\_MUTEX\_ERRORCHECK 2

#define \_\_SIZEOF\_PTHREAD\_BARRIERATTR\_T 4

#define \_\_SIZEOF\_PTHREAD\_CONDATTR\_T 4

#define \_\_SIZEOF\_PTHREAD\_MUTEXATTR\_T 4

#define \_\_SIZEOF\_PTHREAD\_COND\_T 48

#define \_\_SIZEOF\_PTHREAD\_RWLOCKATTR\_T 8

#define pthread\_cleanup\_push(routine,arg) \

{struct \_pthread\_cleanup\_buffer \_buffer;\

\_pthread\_cleanup\_push(&\_buffer,(routine),(arg));

#define pthread\_cleanup\_pop(execute) \_pthread\_cleanup\_pop(&\_buffer,(execute));}

#define PTHREAD\_COND\_INITIALIZER { { 0, 0, 0, 0, 0, (void \*) 0, 0, 0 } }

struct \_pthread\_cleanup\_buffer {

void (\*\_\_routine) (void \*);

void \*\_\_arg;

int \_\_canceltype;

struct \_pthread\_cleanup\_buffer \*\_\_prev;

};

typedef unsigned int pthread\_key\_t;

typedef int pthread\_once\_t;

typedef volatile int pthread\_spinlock\_t;

typedef union {

char \_\_size[\_\_SIZEOF\_PTHREAD\_BARRIERATTR\_T];

int \_\_align;

} pthread\_barrierattr\_t;

enum {

PTHREAD\_PRIO\_NONE,

PTHREAD\_PRIO\_INHERIT,

PTHREAD\_PRIO\_PROTECT

};

enum {

PTHREAD\_MUTEX\_STALLED = 0,

PTHREAD\_MUTEX\_STALLED\_NP = 0,

PTHREAD\_MUTEX\_ROBUST = 1,

PTHREAD\_MUTEX\_ROBUST\_NP = 1

};

enum {

PTHREAD\_RWLOCK\_PREFER\_READER\_NP,

PTHREAD\_RWLOCK\_PREFER\_WRITER\_NP,

PTHREAD\_RWLOCK\_PREFER\_WRITER\_NONRECURSIVE\_NP,

PTHREAD\_RWLOCK\_DEFAULT\_NP = PTHREAD\_RWLOCK\_PREFER\_READER\_NP

};

typedef union {

struct \_\_pthread\_mutex\_s \_\_data;

char \_\_size[\_\_SIZEOF\_PTHREAD\_MUTEX\_T];

long int \_\_align;

} pthread\_mutex\_t;

typedef union {

char \_\_size[\_\_SIZEOF\_PTHREAD\_MUTEXATTR\_T];

int \_\_align;

} pthread\_mutexattr\_t;

typedef union {

char \_\_size[\_\_SIZEOF\_PTHREAD\_ATTR\_T];

long int \_\_align;

} pthread\_attr\_t;

typedef union {

struct {

int \_\_lock;

unsigned int \_\_futex;

unsigned long long int \_\_total\_seq;

unsigned long long int \_\_wakeup\_seq;

unsigned long long int \_\_woken\_seq;

void \*\_\_mutex;

unsigned int \_\_nwaiters;

unsigned int \_\_broadcast\_seq;

} \_\_data;

char \_\_size[\_\_SIZEOF\_PTHREAD\_COND\_T];

long long int \_\_align;

} pthread\_cond\_t;

typedef union {

char \_\_size[\_\_SIZEOF\_PTHREAD\_CONDATTR\_T];

int \_\_align;

} pthread\_condattr\_t;

typedef union {

char \_\_size[\_\_SIZEOF\_PTHREAD\_RWLOCKATTR\_T];

long int \_\_align;

} pthread\_rwlockattr\_t;

#define PTHREAD\_CREATE\_JOINABLE 0

#define PTHREAD\_INHERIT\_SCHED 0

#define PTHREAD\_ONCE\_INIT 0

#define PTHREAD\_PROCESS\_PRIVATE 0

#define PTHREAD\_CREATE\_DETACHED 1

#define PTHREAD\_EXPLICIT\_SCHED 1

#define PTHREAD\_PROCESS\_SHARED 1

#define PTHREAD\_CANCELED ((void\*)-1)

#define PTHREAD\_CANCEL\_DEFERRED 0

#define PTHREAD\_CANCEL\_ENABLE 0

#define PTHREAD\_CANCEL\_ASYNCHRONOUS 1

#define PTHREAD\_CANCEL\_DISABLE 1

extern int \_\_register\_atfork(void (\*)(void), void (\*)(void),

void (\*)(void), void \*);

extern void \_pthread\_cleanup\_pop(struct \_pthread\_cleanup\_buffer \*, int);

extern void \_pthread\_cleanup\_push(struct \_pthread\_cleanup\_buffer \*,

void (\*)(void \*), void \*);

extern int pthread\_atfork(void (\*\_\_prepare) (void),

void (\*\_\_parent) (void), void (\*\_\_child) (void));

extern int pthread\_attr\_destroy(pthread\_attr\_t \* \_\_attr);

extern int pthread\_attr\_getdetachstate(const pthread\_attr\_t \* \_\_attr,

int \*\_\_detachstate);

extern int pthread\_attr\_getguardsize(const pthread\_attr\_t \* \_\_attr,

size\_t \* \_\_guardsize);

extern int pthread\_attr\_getinheritsched(const pthread\_attr\_t \* \_\_attr,

int \*\_\_inherit);

extern int pthread\_attr\_getschedparam(const pthread\_attr\_t \* \_\_attr,

struct sched\_param \*\_\_param);

extern int pthread\_attr\_getschedpolicy(const pthread\_attr\_t \* \_\_attr,

int \*\_\_policy);

extern int pthread\_attr\_getscope(const pthread\_attr\_t \* \_\_attr,

int \*\_\_scope);

extern int pthread\_attr\_getstack(const pthread\_attr\_t \* \_\_attr,

void \*\*\_\_stackaddr, size\_t \* \_\_stacksize);

extern int pthread\_attr\_getstackaddr(const pthread\_attr\_t \* \_\_attr,

void \*\*\_\_stackaddr);

extern int pthread\_attr\_getstacksize(const pthread\_attr\_t \* \_\_attr,

size\_t \* \_\_stacksize);

extern int pthread\_attr\_init(pthread\_attr\_t \* \_\_attr);

extern int pthread\_attr\_setdetachstate(pthread\_attr\_t \* \_\_attr,

int \_\_detachstate);

extern int pthread\_attr\_setguardsize(pthread\_attr\_t \* \_\_attr,

size\_t \_\_guardsize);

extern int pthread\_attr\_setinheritsched(pthread\_attr\_t \* \_\_attr,

int \_\_inherit);

extern int pthread\_attr\_setschedparam(pthread\_attr\_t \* \_\_attr,

const struct sched\_param \*\_\_param);

extern int pthread\_attr\_setschedpolicy(pthread\_attr\_t \* \_\_attr,

int \_\_policy);

extern int pthread\_attr\_setscope(pthread\_attr\_t \* \_\_attr, int \_\_scope);

extern int pthread\_attr\_setstack(pthread\_attr\_t \* \_\_attr,

void \*\_\_stackaddr, size\_t \_\_stacksize);

extern int pthread\_attr\_setstackaddr(pthread\_attr\_t \* \_\_attr,

void \*\_\_stackaddr);

extern int pthread\_attr\_setstacksize(pthread\_attr\_t \* \_\_attr,

size\_t \_\_stacksize);

extern int pthread\_barrier\_destroy(pthread\_barrier\_t \* \_\_barrier);

extern int pthread\_barrier\_init(pthread\_barrier\_t \* \_\_barrier,

const pthread\_barrierattr\_t \* \_\_attr,

unsigned int \_\_count);

extern int pthread\_barrier\_wait(pthread\_barrier\_t \* \_\_barrier);

extern int pthread\_barrierattr\_destroy(pthread\_barrierattr\_t \* \_\_attr);

extern int pthread\_barrierattr\_getpshared(const pthread\_barrierattr\_t \*

\_\_attr, int \*\_\_pshared);

extern int pthread\_barrierattr\_init(pthread\_barrierattr\_t \* \_\_attr);

extern int pthread\_barrierattr\_setpshared(pthread\_barrierattr\_t \* \_\_attr,

int \_\_pshared);

extern int pthread\_cancel(pthread\_t \_\_th);

extern int pthread\_cond\_broadcast(pthread\_cond\_t \* \_\_cond);

extern int pthread\_cond\_destroy(pthread\_cond\_t \* \_\_cond);

extern int pthread\_cond\_init(pthread\_cond\_t \* \_\_cond,

const pthread\_condattr\_t \* \_\_cond\_attr);

extern int pthread\_cond\_signal(pthread\_cond\_t \* \_\_cond);

extern int pthread\_cond\_timedwait(pthread\_cond\_t \* \_\_cond,

pthread\_mutex\_t \* \_\_mutex,

const struct timespec \*\_\_abstime);

extern int pthread\_cond\_wait(pthread\_cond\_t \* \_\_cond,

pthread\_mutex\_t \* \_\_mutex);

extern int pthread\_condattr\_destroy(pthread\_condattr\_t \* \_\_attr);

extern int pthread\_condattr\_getclock(const pthread\_condattr\_t \* attr,

clockid\_t \* clock\_id);

extern int pthread\_condattr\_getpshared(const pthread\_condattr\_t \* \_\_attr,

int \*\_\_pshared);

extern int pthread\_condattr\_init(pthread\_condattr\_t \* \_\_attr);

extern int pthread\_condattr\_setclock(pthread\_condattr\_t \* attr,

clockid\_t clock\_id);

extern int pthread\_condattr\_setpshared(pthread\_condattr\_t \* \_\_attr,

int \_\_pshared);

extern int pthread\_create(pthread\_t \* \_\_newthread,

const pthread\_attr\_t \* \_\_attr,

void \*(\*\_\_start\_routine) (void \*), void \*\_\_arg);

extern int pthread\_detach(pthread\_t \_\_th);

extern int pthread\_equal(pthread\_t \_\_thread1, pthread\_t \_\_thread2);

extern void pthread\_exit(void \*\_\_retval);

extern int pthread\_getattr\_np(pthread\_t thread, pthread\_attr\_t \* attr);

extern int pthread\_getconcurrency(void);

extern int pthread\_getcpuclockid(pthread\_t \_\_thread\_id,

clockid\_t \* \_\_clock\_id);

extern int pthread\_getschedparam(pthread\_t \_\_target\_thread, int \*\_\_policy,

struct sched\_param \*\_\_param);

extern void \*pthread\_getspecific(pthread\_key\_t \_\_key);

extern int pthread\_join(pthread\_t \_\_th, void \*\*\_\_thread\_return);

extern int pthread\_key\_create(pthread\_key\_t \* \_\_key,

void (\*\_\_destr\_function) (void \*));

extern int pthread\_key\_delete(pthread\_key\_t \_\_key);

extern int pthread\_mutex\_consistent(pthread\_mutex\_t \* mutex);

extern int pthread\_mutex\_consistent\_np(pthread\_mutex\_t \* \_\_mutex);

extern int pthread\_mutex\_destroy(pthread\_mutex\_t \* \_\_mutex);

extern int pthread\_mutex\_getprioceiling(const pthread\_mutex\_t \* \_\_mutex,

int \*\_\_prioceiling);

extern int pthread\_mutex\_init(pthread\_mutex\_t \* \_\_mutex,

const pthread\_mutexattr\_t \* \_\_mutexattr);

extern int pthread\_mutex\_lock(pthread\_mutex\_t \* \_\_mutex);

extern int pthread\_mutex\_setprioceiling(pthread\_mutex\_t \* \_\_mutex,

int \_\_prioceiling,

int \*\_\_old\_ceiling);

extern int pthread\_mutex\_timedlock(pthread\_mutex\_t \* \_\_mutex,

const struct timespec \*\_\_abstime);

extern int pthread\_mutex\_trylock(pthread\_mutex\_t \* \_\_mutex);

extern int pthread\_mutex\_unlock(pthread\_mutex\_t \* \_\_mutex);

extern int pthread\_mutexattr\_destroy(pthread\_mutexattr\_t \* \_\_attr);

extern int pthread\_mutexattr\_getprioceiling(const pthread\_mutexattr\_t \*

\_\_attr, int \*\_\_prioceiling);

extern int pthread\_mutexattr\_getprotocol(const pthread\_mutexattr\_t \*

\_\_attr, int \*\_\_protocol);

extern int pthread\_mutexattr\_getpshared(const pthread\_mutexattr\_t \* \_\_attr,

int \*\_\_pshared);

extern int pthread\_mutexattr\_getrobust(const pthread\_mutexattr\_t \* attr,

int \*robust);

extern int pthread\_mutexattr\_getrobust\_np(const pthread\_mutexattr\_t \*

\_\_attr, int \*\_\_robustness);

extern int pthread\_mutexattr\_gettype(const pthread\_mutexattr\_t \* \_\_attr,

int \*\_\_kind);

extern int pthread\_mutexattr\_init(pthread\_mutexattr\_t \* \_\_attr);

extern int pthread\_mutexattr\_setprioceiling(pthread\_mutexattr\_t \* \_\_attr,

int \_\_prioceiling);

extern int pthread\_mutexattr\_setprotocol(pthread\_mutexattr\_t \* \_\_attr,

int \_\_protocol);

extern int pthread\_mutexattr\_setpshared(pthread\_mutexattr\_t \* \_\_attr,

int \_\_pshared);

extern int pthread\_mutexattr\_setrobust(pthread\_mutexattr\_t \* attr,

int robust);

extern int pthread\_mutexattr\_setrobust\_np(pthread\_mutexattr\_t \* \_\_attr,

int \_\_robustness);

extern int pthread\_mutexattr\_settype(pthread\_mutexattr\_t \* \_\_attr,

int \_\_kind);

extern int pthread\_once(pthread\_once\_t \* \_\_once\_control,

void (\*\_\_init\_routine) (void));

extern int pthread\_rwlock\_destroy(pthread\_rwlock\_t \* \_\_rwlock);

extern int pthread\_rwlock\_init(pthread\_rwlock\_t \* \_\_rwlock,

const pthread\_rwlockattr\_t \* \_\_attr);

extern int pthread\_rwlock\_rdlock(pthread\_rwlock\_t \* \_\_rwlock);

extern int pthread\_rwlock\_timedrdlock(pthread\_rwlock\_t \* \_\_rwlock,

const struct timespec \*\_\_abstime);

extern int pthread\_rwlock\_timedwrlock(pthread\_rwlock\_t \* \_\_rwlock,

const struct timespec \*\_\_abstime);

extern int pthread\_rwlock\_tryrdlock(pthread\_rwlock\_t \* \_\_rwlock);

extern int pthread\_rwlock\_trywrlock(pthread\_rwlock\_t \* \_\_rwlock);

extern int pthread\_rwlock\_unlock(pthread\_rwlock\_t \* \_\_rwlock);

extern int pthread\_rwlock\_wrlock(pthread\_rwlock\_t \* \_\_rwlock);

extern int pthread\_rwlockattr\_destroy(pthread\_rwlockattr\_t \* \_\_attr);

extern int pthread\_rwlockattr\_getkind\_np(const pthread\_rwlockattr\_t \*

\_\_attr, int \*\_\_pref);

extern int pthread\_rwlockattr\_getpshared(const pthread\_rwlockattr\_t \*

\_\_attr, int \*\_\_pshared);

extern int pthread\_rwlockattr\_init(pthread\_rwlockattr\_t \* \_\_attr);

extern int pthread\_rwlockattr\_setkind\_np(pthread\_rwlockattr\_t \* \_\_attr,

int \_\_pref);

extern int pthread\_rwlockattr\_setpshared(pthread\_rwlockattr\_t \* \_\_attr,

int \_\_pshared);

extern pthread\_t pthread\_self(void);

extern int pthread\_setcancelstate(int \_\_state, int \*\_\_oldstate);

extern int pthread\_setcanceltype(int \_\_type, int \*\_\_oldtype);

extern int pthread\_setconcurrency(int \_\_level);

extern int pthread\_setschedparam(pthread\_t \_\_target\_thread, int \_\_policy,

const struct sched\_param \*\_\_param);

extern int pthread\_setschedprio(pthread\_t \_\_target\_thread, int \_\_prio);

extern int pthread\_setspecific(pthread\_key\_t \_\_key, const void \*\_\_pointer);

extern int pthread\_spin\_destroy(pthread\_spinlock\_t \* \_\_lock);

extern int pthread\_spin\_init(pthread\_spinlock\_t \* \_\_lock, int \_\_pshared);

extern int pthread\_spin\_lock(pthread\_spinlock\_t \* \_\_lock);

extern int pthread\_spin\_trylock(pthread\_spinlock\_t \* \_\_lock);

extern int pthread\_spin\_unlock(pthread\_spinlock\_t \* \_\_lock);

extern void pthread\_testcancel(void);

### **14.10.3 semaphore.h**

typedef union {

char \_\_size[\_\_SIZEOF\_SEM\_T];

long int \_\_align;

} sem\_t;

#define SEM\_FAILED ((sem\_t\*)0)

#define SEM\_VALUE\_MAX ((int)((~0u)>>1))

extern int sem\_close(sem\_t \* \_\_sem);

extern int sem\_destroy(sem\_t \* \_\_sem);

extern int sem\_getvalue(sem\_t \* \_\_sem, int \*\_\_sval);

extern int sem\_init(sem\_t \* \_\_sem, int \_\_pshared, unsigned int \_\_value);

extern sem\_t \*sem\_open(const char \*\_\_name, int \_\_oflag, ...);

extern int sem\_post(sem\_t \* \_\_sem);

extern int sem\_timedwait(sem\_t \* \_\_sem, const struct timespec \*\_\_abstime);

extern int sem\_trywait(sem\_t \* \_\_sem);

extern int sem\_unlink(const char \*\_\_name);

extern int sem\_wait(sem\_t \* \_\_sem);

## **14.11 Interface Definitions for libpthread**

The interfaces defined on the following pages are included in libpthread and are defined by this specification. Unless otherwise noted, these interfaces shall be included in the source standard.

Other interfaces listed in [Section 14.9](#ID_LIBPTHREAD) shall behave as described in the referenced base document.

#### \_pthread\_cleanup\_pop

##### Name

\_pthread\_cleanup\_pop — establish cancellation handlers

##### Synopsis

#include <pthread.h>

void \_pthread\_cleanup\_pop(struct \_pthread\_cleanup\_buffer \*, int);

##### Description

The \_pthread\_cleanup\_pop() function provides an implementation of the pthread\_cleanup\_pop() macro described in [*POSIX 1003.1-2008 (ISO/IEC 9945-2009)*](#ID_STD_46_SUSV4).

The \_pthread\_cleanup\_pop() function is not in the source standard; it is only in the binary standard.

#### \_pthread\_cleanup\_push

##### Name

\_pthread\_cleanup\_push — establish cancellation handlers

##### Synopsis

#include <pthread.h>

void \_pthread\_cleanup\_push(struct \_pthread\_cleanup\_buffer \*, void (\*) (void \*), void \*);

##### Description

The \_pthread\_cleanup\_push() function provides an implementation of the pthread\_cleanup\_push() macro described in [*POSIX 1003.1-2008 (ISO/IEC 9945-2009)*](#ID_STD_46_SUSV4).

The \_pthread\_cleanup\_push() function is not in the source standard; it is only in the binary standard.

#### pthread\_getattr\_np

##### Name

pthread\_getattr\_np — get thread attributes

##### Synopsis

#include <pthread.h>

int pthread\_getattr\_np(pthread\_t *thread*, pthread\_attr\_t \**attr*);

##### Description

pthread\_getattr\_np() fills in the thread attribute object *attr* with attribute values describing the running thread *thread*. This is useful to detect runtime changes from the values specified in the thread attributes object used to create the thread with pthread\_create(). The following differences may be noted:

• The detach state, since a joinable thread may have detached itself after creation. Use pthread\_attr\_getdetachstate() to extract from *attr*.

• The stack size, which the implementation may align to a suitable boundary. Use pthread\_attr\_getstack() to extract from *attr*.

• The guard size, which the implementation may round upwards to a multiple of the page size, or ignore (i.e., treat as 0), if the application is allocating its own stack. Use pthread\_attr\_getguardsize() to extract from *attr*.

If the stack address attribute was not set in the thread attributes object used to create the thread, then the thread attributes object returned by pthread\_getattr\_np() will show the actual stack address the implementation selected for the thread. Use pthread\_attr\_getstack() to extract from *attr*.

The thread attributes object *attr* should be destroyed using pthread\_attr\_destroy() when it is no longer needed.

##### Return Value

On success, pthread\_getattr\_np() returns 0; on error, it returns a non-zero error number.

##### Errors

ENOMEM

  Insufficient memory to complete the operation.

In addition, if *thread* refers to the main thread, then pthread\_getattr\_np() may also fail due to errors from various underlying calls: fopen(), if the pseudo-file containing the memory region map cannot be opened; getrlimit() if the RLIMIT\_STACK resource limit it not supported.

##### Notes

This function is a GNU extension.

##### See Also

pthread\_attr\_destroy(), pthread\_attr\_getdetachstate(), pthread\_attr\_getguardsize(), pthread\_attr\_getstack(), pthread\_create().

#### pthread\_mutex\_consistent\_np

##### Name

pthread\_mutex\_consistent\_np — mark state protected by robust mutex as consistent

##### Synopsis

#include <pthread.h>

int pthread\_mutex\_consistent\_np(pthread\_mutex\_t \* *\_\_mutex*);

##### Description

pthread\_mutex\_consistent\_np() shall behave as described for pthread\_mutex\_consistent() in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

#### pthread\_mutexattr\_getrobust\_np, pthread\_mutexattr\_setrobust\_np

##### Name

pthread\_mutexattr\_getrobust\_np, pthread\_mutexattr\_setrobust\_np — get and set the mutex robust attribute

##### Synopsis

#include <pthread.h>

int pthread\_mutexattr\_getrobust\_np(const pthread\_mutexattr\_t \* *\_\_attr*, int \* *\_\_robustness*);

int pthread\_mutexattr\_setrobust\_np(const pthread\_mutexattr\_t \* *\_\_attr*, int *\_\_robustness*);

##### Description

pthread\_mutexattr\_setrobust\_np() shall behave as described for pthread\_mutexattr\_setrobust() in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

pthread\_mutexattr\_getrobust\_np() shall behave as described for pthread\_mutexattr\_getrobust() in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

Two additional valid values are defined for *\_\_robustness*: PTHREAD\_MUTEX\_STALLED\_NP, which is identical to PTHREAD\_MUTEX\_STALLED and PTHREAD\_MUTEX\_ROBUST\_NP, which is identical to PTHREAD\_MUTEX\_ROBUST.

#### pthread\_rwlockattr\_getkind\_np, pthread\_rwlockattr\_setkind\_np

##### Name

pthread\_rwlockattr\_getkind\_np, pthread\_rwlockattr\_setkind\_np — get/set the read-write lock kind of the thread read-write lock attribute object

##### Synopsis

#include <pthread.h>

int pthread\_rwlockattr\_getkind\_np(const pthread\_rwlockattr\_t \* *attr*, int \* *pref*);

int pthread\_rwlockattr\_setkind\_np(pthread\_rwlockattr\_t \* *attr*, int \* *pref*);

##### Description

The pthread\_rwlockattr\_setkind\_np() function sets the kind of read-write lock of the thread read-write lock attribute object referred to by attr to the value specified with pref. The argument pref may be set to PTHREAD\_RWLOCK\_PREFER\_READER\_NP, PTHREAD\_RWLOCK\_PREFER\_WRITER\_NONRECURSIVE\_NP, or PTHREAD\_RWLOCK\_PREFER\_WRITER\_NP. The default lock setting is PTHREAD\_RWLOCK\_PREFER\_READER\_NP. A thread may hold multiple read locks, i.e. read locks are recursive. According to The Single Unix Specification, the behavior is unspecified when a reader tries to place a lock, and there is no write lock but writers are waiting. Giving preference to the reader, as is set by default with the PTHREAD\_RWLOCK\_PREFER\_READER\_NP value implies that the reader will receive the requested lock, even if a writer is waiting. As long as there are readers the writer will be starved. Setting the kind to PTHREAD\_RWLOCK\_PREFER\_WRITER\_NONRECURSIVE\_NP, avoids writer starvation as long as any read locking is not done in a recursive fashion. The pthread\_rwlockattr\_getkind\_np() function returns the value of the read-write lock attribute of the thread read-write lock attribute object referred to by *attr* in the pointer *pref*.

##### Return Value

pthread\_rwlockattr\_setkind\_np() function returns 0 on success; on error, it returns a non-zero error number. pthread\_rwlockattr\_setkind\_np() function always returns 0.

##### Errors

EINVAL

*pref* is set to an unsupported value.

##### Notes

Setting the value read-write lock kind to PTHREAD\_RWLOCK\_PREFER\_WRITER\_NP, results in the same behavior as setting the value to PTHREAD\_RWLOCK\_PREFER\_READER\_NP. As long as a reader thread holds the lock the thread holding a write lock will be starved. Setting the kind value to PTHREAD\_RWLOCK\_PREFER\_WRITER\_NONRECURSIVE\_NP, allows the writer to run. However, the writer may not be recursive as is implied by the name.

#### waitpid

##### Name

waitpid — wait for child process

##### Description

waitpid() is as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but with differences as listed below.

#### Need not support WCONTINUED or WIFCONTINUED

Implementations need not support the XSI optional functionality of WCONTINUED() or WIFCONTINUED().

## **14.12 Interfaces for libgcc\_s**

[Table 14-55](#ID_LIB_45_LIBGCC_45_S_45_DEF) defines the library name and shared object name for the libgcc\_s library

**Table 14-55 libgcc\_s Definition**

|  |  |
| --- | --- |
| Library: | libgcc\_s |
| SONAME: | libgcc\_s.so.1 |

The behavior of the interfaces in this library is specified by the following specifications:

|  |
| --- |
| [LSB] [This Specification](#ID_STD_46_LSB) |

### **14.12.1 Unwind Library**

#### 14.12.1.1 Interfaces for Unwind Library

An LSB conforming implementation shall provide the generic functions for Unwind Library specified in [Table 14-56](#ID_TBL_45_LIBGCC_45_S_45_UNWIN_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-56 libgcc\_s - Unwind Library Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| \_Unwind\_Backtrace [[LSB]](#ID_REFSTD_46_LIBGCC_46_S_46_1) | \_Unwind\_DeleteException [[LSB]](#ID_REFSTD_46_LIBGCC_46_S_46_1) | \_Unwind\_FindEnclosingFunction [[LSB]](#ID_REFSTD_46_LIBGCC_46_S_46_1) | \_Unwind\_ForcedUnwind [[LSB]](#ID_REFSTD_46_LIBGCC_46_S_46_1) |
| \_Unwind\_GetCFA [[LSB]](#ID_REFSTD_46_LIBGCC_46_S_46_1) | \_Unwind\_GetGR [[LSB]](#ID_REFSTD_46_LIBGCC_46_S_46_1) | \_Unwind\_GetIP [[LSB]](#ID_REFSTD_46_LIBGCC_46_S_46_1) | \_Unwind\_GetIPInfo(GCC\_4.2.0) [[LSB]](#ID_REFSTD_46_LIBGCC_46_S_46_1) |
| \_Unwind\_GetLanguageSpecificData [[LSB]](#ID_REFSTD_46_LIBGCC_46_S_46_1) | \_Unwind\_GetRegionStart [[LSB]](#ID_REFSTD_46_LIBGCC_46_S_46_1) | \_Unwind\_RaiseException [[LSB]](#ID_REFSTD_46_LIBGCC_46_S_46_1) | \_Unwind\_Resume [[LSB]](#ID_REFSTD_46_LIBGCC_46_S_46_1) |
| \_Unwind\_Resume\_or\_Rethrow [[LSB]](#ID_REFSTD_46_LIBGCC_46_S_46_1) | \_Unwind\_SetGR [[LSB]](#ID_REFSTD_46_LIBGCC_46_S_46_1) | \_Unwind\_SetIP [[LSB]](#ID_REFSTD_46_LIBGCC_46_S_46_1) |  |

## **14.13 Data Definitions for libgcc\_s**

This section defines global identifiers and their values that are associated with interfaces contained in libgcc\_s. These definitions are organized into groups that correspond to system headers. This convention is used as a convenience for the reader, and does not imply the existence of these headers, or their content. Where an interface is defined as requiring a particular system header file all of the data definitions for that system header file presented here shall be in effect.

This section gives data definitions to promote binary application portability, not to repeat source interface definitions available elsewhere. System providers and application developers should use this ABI to supplement - not to replace - source interface definition specifications.

This specification uses the [ISO C (1999)](#ID_STD_46_ISOC99) C Language as the reference programming language, and data definitions are specified in ISO C format. The C language is used here as a convenient notation. Using a C language description of these data objects does not preclude their use by other programming languages.

### **14.13.1 unwind.h**

struct \_Unwind\_Context;

struct \_Unwind\_Exception;

typedef unsigned int \_Unwind\_Ptr \_\_attribute\_\_ ((\_\_mode\_\_(\_\_pointer\_\_)));

typedef unsigned int \_Unwind\_Word \_\_attribute\_\_ ((\_\_mode\_\_(\_\_word\_\_)));

typedef unsigned int \_Unwind\_Exception\_Class

\_\_attribute\_\_ ((\_\_mode\_\_(\_\_DI\_\_)));

typedef enum {

\_URC\_NO\_REASON = 0,

\_URC\_FOREIGN\_EXCEPTION\_CAUGHT = 1,

\_URC\_FATAL\_PHASE2\_ERROR = 2,

\_URC\_FATAL\_PHASE1\_ERROR = 3,

\_URC\_NORMAL\_STOP = 4,

\_URC\_END\_OF\_STACK = 5,

\_URC\_HANDLER\_FOUND = 6,

\_URC\_INSTALL\_CONTEXT = 7,

\_URC\_CONTINUE\_UNWIND = 8

} \_Unwind\_Reason\_Code;

typedef void (\*\_Unwind\_Exception\_Cleanup\_Fn) (\_Unwind\_Reason\_Code,

struct \_Unwind\_Exception \*);

struct \_Unwind\_Exception {

\_Unwind\_Exception\_Class exception\_class;

\_Unwind\_Exception\_Cleanup\_Fn exception\_cleanup;

\_Unwind\_Word private\_1;

\_Unwind\_Word private\_2;

} \_\_attribute\_\_ ((\_\_aligned\_\_));

#define \_UA\_SEARCH\_PHASE 1

#define \_UA\_END\_OF\_STACK 16

#define \_UA\_CLEANUP\_PHASE 2

#define \_UA\_HANDLER\_FRAME 4

#define \_UA\_FORCE\_UNWIND 8

typedef int \_Unwind\_Action;

typedef \_Unwind\_Reason\_Code(\*\_Unwind\_Stop\_Fn) (int version,

\_Unwind\_Action actions,

\_Unwind\_Exception\_Class

exceptionClass,

struct \_Unwind\_Exception \*

exceptionObject,

struct \_Unwind\_Context \*

context,

void \*stop\_parameter);

typedef \_Unwind\_Reason\_Code(\*\_Unwind\_Trace\_Fn) (struct \_Unwind\_Context \*,

void \*);

extern \_Unwind\_Reason\_Code \_Unwind\_Backtrace(\_Unwind\_Trace\_Fn, void \*);

extern void \_Unwind\_DeleteException(struct \_Unwind\_Exception \*);

extern void \*\_Unwind\_FindEnclosingFunction(void \*);

extern \_Unwind\_Reason\_Code \_Unwind\_ForcedUnwind(struct \_Unwind\_Exception \*,

\_Unwind\_Stop\_Fn, void \*);

extern \_Unwind\_Word \_Unwind\_GetCFA(struct \_Unwind\_Context \*);

extern \_Unwind\_Word \_Unwind\_GetGR(struct \_Unwind\_Context \*, int);

extern \_Unwind\_Ptr \_Unwind\_GetIP(struct \_Unwind\_Context \*);

extern \_Unwind\_Ptr \_Unwind\_GetIPInfo(struct \_Unwind\_Context \*, int \*);

extern void \*\_Unwind\_GetLanguageSpecificData(struct \_Unwind\_Context \*);

extern \_Unwind\_Ptr \_Unwind\_GetRegionStart(struct \_Unwind\_Context \*);

extern \_Unwind\_Reason\_Code \_Unwind\_RaiseException(struct \_Unwind\_Exception

\*);

extern void \_Unwind\_Resume(struct \_Unwind\_Exception \*);

extern \_Unwind\_Reason\_Code \_Unwind\_Resume\_or\_Rethrow(struct

\_Unwind\_Exception \*);

extern void \_Unwind\_SetGR(struct \_Unwind\_Context \*, int, u\_int64\_t);

extern void \_Unwind\_SetIP(struct \_Unwind\_Context \*, \_Unwind\_Ptr);

## **14.14 Interface Definitions for libgcc\_s**

The interfaces defined on the following pages are included in libgcc\_s and are defined by this specification. Unless otherwise noted, these interfaces shall be included in the source standard.

Other interfaces listed in [Section 14.12](#ID_LIBGCC_45_S) shall behave as described in the referenced base document.

#### \_Unwind\_Backtrace

##### Name

\_Unwind\_Backtrace — private C++ error handling method

##### Synopsis

\_Unwind\_Reason\_Code \_Unwind\_Backtrace(\_Unwind\_Trace\_Fn *trace*, void \* *trace\_argument*);

##### Description

\_Unwind\_Backtrace() performs a stack backtrace using unwind data. The *trace* callback is called for every stack frame in the call chain. No cleanup actions are performed.

#### \_Unwind\_DeleteException

##### Name

\_Unwind\_DeleteException — private C++ error handling method

##### Synopsis

void \_Unwind\_DeleteException(struct \_Unwind\_Exception \* *object*);

##### Description

\_Unwind\_DeleteException() deletes the given exception *object*. If a given runtime resumes normal execution after catching a foreign exception, it will not know how to delete that exception. Such an exception shall be deleted by calling \_Unwind\_DeleteException(). This is a convenience function that calls the function pointed to by the *exception\_cleanup* field of the exception header.

#### \_Unwind\_FindEnclosingFunction

##### Name

\_Unwind\_FindEnclosingFunction — private C++ error handling method

##### Synopsis

void \* \_Unwind\_FindEnclosingFunction(void \* *ip*);

##### Description

\_Unwind\_FindEnclosingFunction() Find the start address of the procedure containing the specified *ip* or NULL if it cannot be found (for example, because the function has no unwind info).

Note that there is not necessarily a one-to-one correspondence between source level functions and procedures. Some functions do not have unwind-info and others are split into multiple procedures.

#### \_Unwind\_ForcedUnwind

##### Name

\_Unwind\_ForcedUnwind — private C++ error handling method

##### Synopsis

#include <unwind.h>

\_Unwind\_Reason\_Code \_Unwind\_ForcedUnwind(struct \_Unwind\_Exception \* *object*, \_Unwind\_Stop\_Fn *stop*, void \* *stop\_parameter*);

##### Description

Forced unwinding is a single-phase process. *stop* and *stop\_parameter* control the termination of the unwind process instead of the usual personality routine query. Stop function *stop* is called for each unwind frame, with the parameteres described for the usual personality routine below, plus an additional *stop\_parameter*.

##### Return Value

When *stop* identifies the destination frame, it transfers control to the user code as appropriate without returning, normally after calling \_Unwind\_DeleteException(). If not, then it should return an \_Unwind\_Reason\_Code value.

If *stop* returns any reason code other than \_URC\_NO\_REASON, then the stack state is indeterminate from the point of view of the caller of \_Unwind\_ForcedUnwind(). Rather than attempt to return, therefore, the unwind library should use the *exception\_cleanup* entry in *object*, and then call abort().

\_URC\_NO\_REASON

  This is not the destination from. The unwind runtime will call frame's personality routine with the \_UA\_FORCE\_UNWIND and \_UA\_CLEANUP\_PHASE flag set in *actions*, and then unwind to the next frame and call the stop() function again.

\_URC\_END\_OF\_STACK

  In order to allow \_Unwind\_ForcedUnwind() to perform special processing when it reaches the end of the stack, the unwind runtime will call it after the last frame is rejected, with a NULL stack pointer in the context, and the STOP() FUNCTION SHALL CATCH THIS CONDITION. iT MAY return this code if it cannot handle end-of-stack.

\_URC\_FATAL\_PHASE2\_ERROR

  The stop() function may return this code for other fatal conditions like stack corruption.

#### \_Unwind\_GetCFA

##### Name

\_Unwind\_GetCFA — private C++ error handling method

##### Synopsis

\_Unwind\_Word \_Unwind\_GetCFA(struct \_Unwind\_Context \* *context*);

##### Description

\_Unwind\_GetCFA() shall retrieve the value of the Canonical Frame Address (CFA) of the given *context*.

#### \_Unwind\_GetGR

##### Name

\_Unwind\_GetGR — private C++ error handling method

##### Synopsis

\_Unwind\_Word \_Unwind\_GetGR(struct \_Unwind\_Context \* *context*, int *index*);

##### Description

\_Unwind\_GetGR() returns data at *index* found in *context*. The register is identified by its index: 0 to 31 are for the fixed registers, and 32 to 127 are for the stacked registers.

During the two phases of unwinding, only GR1 has a guaranteed value, which is the global pointer of the frame referenced by the unwind *context*. If the register has its NAT bit set, the behavior is unspecified.

#### \_Unwind\_GetIP

##### Name

\_Unwind\_GetIP — private C++ error handling method

##### Synopsis

\_Unwind\_Ptr \_Unwind\_GetIP(struct \_Unwind\_Context \* *context*);

##### Description

\_Unwind\_GetIP() returns the instruction pointer value for the routine identified by the unwind *context*.

#### \_Unwind\_GetIPInfo

##### Name

\_Unwind\_GetIPInfo — private C++ error handling method

##### Synopsis

\_Unwind\_Ptr \_Unwind\_GetIPInfo(struct \_Unwind\_Context \* *context*, int \* *ip\_before\_insn*);

##### Description

\_Unwind\_GetIPInfo() returns the instruction pointer value for the routine identified by the unwind *context* and sets *ip\_before\_insn* flag indicating whether that IP is before or after first not yet fully executed instruction.

#### \_Unwind\_GetLanguageSpecificData

##### Name

\_Unwind\_GetLanguageSpecificData — private C++ error handling method

##### Synopsis

#include <unwind.h>

\_Unwind\_Ptr \_Unwind\_GetLanguageSpecificData(struct \_Unwind\_Context \* *context*);

##### Description

\_Unwind\_GetLanguageSpecificData() returns the address of the language specific data area for the current stack frame described by *context*.

#### \_Unwind\_GetRegionStart

##### Name

\_Unwind\_GetRegionStart — private C++ error handling method

##### Synopsis

\_Unwind\_Ptr \_Unwind\_GetRegionStart(struct \_Unwind\_Context \* *context*);

##### Description

\_Unwind\_GetRegionStart() routine returns the address (i.e., 0) of the beginning of the procedure or code fragment described by the current unwind descriptor block.

#### \_Unwind\_RaiseException

##### Name

\_Unwind\_RaiseException — private C++ error handling method

##### Synopsis

\_Unwind\_Reason\_Code \_Unwind\_RaiseException(struct \_Unwind\_Exception \* *object*);

##### Description

\_Unwind\_RaiseException() raises an exception, passing along the given exception *object*, which should have its *exception\_class* and *exception\_cleanup* fields set. The exception object has been allocated by the language-specific runtime, and has a language-specific format, exception that it shall contain an \_Unwind\_Exception.

##### Return Value

\_Unwind\_RaiseException() does not return unless an error condition is found. If an error condition occurs, an \_Unwind\_Reason\_Code is returnd:

\_URC\_END\_OF\_STACK

  The unwinder encountered the end of the stack during phase one without finding a handler. The unwind runtime will not have modified the stack. The C++ runtime will normally call uncaught\_exception() in this case.

\_URC\_FATAL\_PHASE1\_ERROR

  The unwinder encountered an unexpected error during phase one, because of something like stack corruption. The unwind runtime will not have modified the stack. The C++ runtime will normally call terminate() in this case.

\_URC\_FATAL\_PHASE2\_ERROR

  The unwinder encountered an unexpected error during phase two. This is usually a *throw*, which will call terminate().

#### \_Unwind\_Resume

##### Name

\_Unwind\_Resume — private C++ error handling method

##### Synopsis

void \_Unwind\_Resume(struct \_Unwind\_Exception \* *object*);

##### Description

\_Unwind\_Resume() resumes propagation of an existing exception *object*. A call to this routine is inserted as the end of a landing pad that performs cleanup, but does not resume normal execution. It causes unwinding to proceed further.

#### \_Unwind\_Resume\_or\_Rethrow

##### Name

\_Unwind\_Resume\_or\_Rethrow — private C++ error handling method

##### Synopsis

\_Unwind\_Reason\_Code \_Unwind\_Resume\_or\_Rethrow(struct \_Unwind\_Exception \* *exception\_object*);

##### Description

If the unwind was initiated due to a forced unwind, \_Unwind\_Resume\_or\_Rethrow() shall resume that operation, else it shall re-raise the exception.

#### \_Unwind\_SetGR

##### Name

\_Unwind\_SetGR — private C++ error handling method

##### Synopsis

void \_Unwind\_SetGR(struct \_Unwind\_Context \* *context*, int *index*, uint *value*);

##### Description

\_Unwind\_SetGR() sets the *value* of the register *index*ed for the routine identified by the unwind *context*.

#### \_Unwind\_SetIP

##### Name

\_Unwind\_SetIP — private C++ error handling method

##### Synopsis

#include <unwind.h>

void \_Unwind\_SetIP(struct \_Unwind\_Context \* *context*, \_Unwind\_Ptr *value*);

##### Description

\_Unwind\_SetIP() sets the instruction pointer for the routine identified by the unwind *context* to *value*.

## **14.15 Interfaces for libdl**

[Table 14-57](#ID_LIB_45_LIBDL_45_DEF) defines the library name and shared object name for the libdl library

**Table 14-57 libdl Definition**

|  |  |
| --- | --- |
| Library: | libdl |
| SONAME: | libdl.so.2 |

The behavior of the interfaces in this library is specified by the following specifications:

|  |
| --- |
| [LSB] [This Specification](#ID_STD_46_LSB) |
| [SUSv4] [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) |

### **14.15.1 Dynamic Loader**

#### 14.15.1.1 Interfaces for Dynamic Loader

An LSB conforming implementation shall provide the generic functions for Dynamic Loader specified in [Table 14-58](#ID_TBL_45_LIBDL_45_DYNAM_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-58 libdl - Dynamic Loader Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| dladdr [[LSB]](#ID_REFSTD_46_LIBDL_46_1) | dlclose [[SUSv4]](#ID_REFSTD_46_LIBDL_46_2) | dlerror [[SUSv4]](#ID_REFSTD_46_LIBDL_46_2) | dlopen [[LSB]](#ID_REFSTD_46_LIBDL_46_1) |
| dlsym [[LSB]](#ID_REFSTD_46_LIBDL_46_1) | dlvsym [[LSB]](#ID_REFSTD_46_LIBDL_46_1) |  |  |

## **14.16 Data Definitions for libdl**

This section defines global identifiers and their values that are associated with interfaces contained in libdl. These definitions are organized into groups that correspond to system headers. This convention is used as a convenience for the reader, and does not imply the existence of these headers, or their content. Where an interface is defined as requiring a particular system header file all of the data definitions for that system header file presented here shall be in effect.

This section gives data definitions to promote binary application portability, not to repeat source interface definitions available elsewhere. System providers and application developers should use this ABI to supplement - not to replace - source interface definition specifications.

This specification uses the [ISO C (1999)](#ID_STD_46_ISOC99) C Language as the reference programming language, and data definitions are specified in ISO C format. The C language is used here as a convenient notation. Using a C language description of these data objects does not preclude their use by other programming languages.

### **14.16.1 dlfcn.h**

#define RTLD\_NEXT ((void \*) -1l)

#define RTLD\_DEFAULT ((void \*) 0)

#define RTLD\_LOCAL 0

#define RTLD\_LAZY 0x00001

#define RTLD\_NOW 0x00002

#define RTLD\_NOLOAD 0x00004

#define RTLD\_DEEPBIND 0x00008

#define RTLD\_GLOBAL 0x00100

#define RTLD\_NODELETE 0x01000

typedef struct {

char \*dli\_fname;

void \*dli\_fbase;

char \*dli\_sname;

void \*dli\_saddr;

} Dl\_info;

extern int dladdr(const void \*\_\_address, Dl\_info \* \_\_info);

extern int dlclose(void \*\_\_handle);

extern char \*dlerror(void);

extern void \*dlopen(const char \*\_\_file, int \_\_mode);

extern void \*dlsym(void \*\_\_handle, const char \*\_\_name);

extern void \*dlvsym(void \*handle, const char \*name, const char \*version);

## **14.17 Interface Definitions for libdl**

The interfaces defined on the following pages are included in libdl and are defined by this specification. Unless otherwise noted, these interfaces shall be included in the source standard.

Other interfaces listed in [Section 14.15](#ID_LIBDL) shall behave as described in the referenced base document.

#### dladdr

##### Name

dladdr — find the shared object containing a given address

##### Synopsis

#include <dlfcn.h>

typedef struct {

const char *\*dli\_fname*;

void *\*dli\_fbase*;

const char *\*dli\_sname*;

void *\*dli\_saddr*;

} Dl\_info;

int dladdr(const void \* *addr*, Dl\_info \* *dlip*);

##### Description

The dladdr() function shall query the dynamic linker for information about the shared object containing the address *addr*. The information shall be returned in the user supplied data structure referenced by *dlip*.

The structure shall contain at least the following members:

*dli\_fname*

  The pathname of the shared object containing the address

*dli\_fbase*

  The base address at which the shared object is mapped into the address space of the calling process.

*dli\_sname*

  The name of the nearest runtime symbol with value less than or equal to *addr*. Where possible, the symbol name shall be returned as it would appear in C source code.

If no symbol with a suitable value is found, both this field and *dli\_saddr* shall be set to NULL.

*dli\_saddr*

  The address of the symbol returned in *dli\_sname*. This address has type "pointer to *type*", where *type* is the type of the symbol *dli\_sname*.

**Example:** If the symbol in *dli\_sname* is a function, then the type of *dli\_saddr* is of type "pointer to function".

The behavior of dladdr() is only specified in dynamically linked programs.

##### Return Value

On success, dladdr() shall return non-zero, and the structure referenced by *dlip* shall be filled in as described. Otherwise, dladdr() shall return zero, and the cause of the error can be fetched with dlerror().

##### Errors

See dlerror().

##### Environment

LD\_LIBRARY\_PATH

  directory search-path for object files

#### dlopen

##### Name

dlopen — open dynamic object

##### Synopsis

#include <dlfcn.h>

void \* dlopen(const char \* *filename*, int *flag*);

##### Description

The dlopen() function shall behave as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but with additional behaviors listed below.

If the file argument does not contain a <slash> character, then the system shall look for a library of that name in at least the following directories, and use the first one which is found:

• The directories specified by the DT\_RPATH dynamic entry.

• The directories specified in the LD\_LIBRARY\_PATH environment variable (which is a colon separated list of pathnames). This step shall be skipped for setuid and setgid executables.

• A set of directories sufficient to contain the libraries specified in this standard.

**Note:** Traditionally, /lib and /usr/lib. This case would also cover cases in which the system used the mechanism of /etc/ld.so.conf and /etc/ld.so.cache to provide access.

Example: An application which is not linked against libm may choose to dlopen libm.

##### Additional flags

In addition to the available values for *flag* as documented in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), the following values may also be ORed into *flag*:

RTLD\_NODELETE

  Do not unload the library during dlclose(). Consequently, the library's static variables are not reinitialized if the library is reloaded with dlopen() at a later time.

RTLD\_NOLOAD

  Do not load the library. This can be used to test if the library is already resident. dlopen() returns a NULL pointer if it is not resident; it returns the library's handle if it is resident. This flag can also be used to promote the flags on a library that is already loaded. For example, a library that was previously loaded with RTLD\_LOCAL can be reopened using RTLD\_NOLOAD|RTLD\_GLOBAL.

RTLD\_DEEPBIND

  Place the lookup scope of the symbols in this library ahead of the global scope. This means that a self-contained library will use its own symbols in preference to global symbols with the same name contained in libraries that have already been loaded.

#### dlsym

##### Name

dlsym — obtain the address of a symbol from a dlopen object

##### Description

dlsym() is as specified in the [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but with differences as listed below.

RTLD\_NEXT, RTLD\_DEFAULT Required

The values RTLD\_NEXT and RTLD\_DEFAULT, described as reserved for future use in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), are required, with behavior as described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

#### dlvsym

##### Name

dlvsym — obtain the address of a symbol from a dlopen object

##### Synopsis

#include <dlfcn.h>

void \* dlvsym(void \* *handle*, char \* *name*, char \* *version*);

##### Description

dlvsym() does the same as dlsym() but takes a version string as an additional argument.

## **14.18 Interfaces for librt**

[Table 14-59](#ID_LIB_45_LIBRT_45_DEF) defines the library name and shared object name for the librt library

**Table 14-59 librt Definition**

|  |  |
| --- | --- |
| Library: | librt |
| SONAME: | librt.so.1 |

The behavior of the interfaces in this library is specified by the following specifications:

|  |
| --- |
| [LFS] [Large File Support](#ID_STD_46_LFS) |
| [SUSv4] [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) |

### **14.18.1 Shared Memory Objects**

#### 14.18.1.1 Interfaces for Shared Memory Objects

An LSB conforming implementation shall provide the generic functions for Shared Memory Objects specified in [Table 14-60](#ID_TBL_45_LIBRT_45_SHARE_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-60 librt - Shared Memory Objects Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| shm\_open [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) | shm\_unlink [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) |  |  |

### **14.18.2 Asynchronous I/O**

#### 14.18.2.1 Interfaces for Asynchronous I/O

An LSB conforming implementation shall provide the generic functions for Asynchronous I/O specified in [Table 14-61](#ID_TBL_45_LIBRT_45_ASYNC_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-61 librt - Asynchronous I/O Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| aio\_cancel [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) | aio\_cancel64 [[LFS]](#ID_REFSTD_46_LIBRT_46_1) | aio\_error [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) | aio\_error64 [[LFS]](#ID_REFSTD_46_LIBRT_46_1) |
| aio\_fsync [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) | aio\_fsync64 [[LFS]](#ID_REFSTD_46_LIBRT_46_1) | aio\_read [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) | aio\_read64 [[LFS]](#ID_REFSTD_46_LIBRT_46_1) |
| aio\_return [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) | aio\_return64 [[LFS]](#ID_REFSTD_46_LIBRT_46_1) | aio\_suspend [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) | aio\_suspend64 [[LFS]](#ID_REFSTD_46_LIBRT_46_1) |
| aio\_write [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) | aio\_write64 [[LFS]](#ID_REFSTD_46_LIBRT_46_1) | lio\_listio(GLIBC\_2.4) [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) | lio\_listio64(GLIBC\_2.4) [[LFS]](#ID_REFSTD_46_LIBRT_46_1) |

### **14.18.3 Clock**

#### 14.18.3.1 Interfaces for Clock

An LSB conforming implementation shall provide the generic functions for Clock specified in [Table 14-62](#ID_TBL_45_LIBRT_45_CLOCK_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-62 librt - Clock Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| clock\_getcpuclockid [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) | clock\_getres [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) | clock\_gettime [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) | clock\_nanosleep [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) |
| clock\_settime [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) |  |  |  |

### **14.18.4 Timers**

#### 14.18.4.1 Interfaces for Timers

An LSB conforming implementation shall provide the generic functions for Timers specified in [Table 14-63](#ID_TBL_45_LIBRT_45_TIMER_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-63 librt - Timers Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| timer\_create [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) | timer\_delete [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) | timer\_getoverrun [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) | timer\_gettime [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) |
| timer\_settime [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) |  |  |  |

### **14.18.5 Message Queues**

#### 14.18.5.1 Interfaces for Message Queues

An LSB conforming implementation shall provide the generic functions for Message Queues specified in [Table 14-64](#ID_TBL_45_LIBRT_45_MESSA_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-64 librt - Message Queues Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| mq\_close(GLIBC\_2.3.4) [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) | mq\_getattr(GLIBC\_2.3.4) [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) | mq\_notify(GLIBC\_2.3.4) [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) | mq\_open(GLIBC\_2.3.4) [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) |
| mq\_receive(GLIBC\_2.3.4) [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) | mq\_send(GLIBC\_2.3.4) [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) | mq\_setattr(GLIBC\_2.3.4) [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) | mq\_timedreceive(GLIBC\_2.3.4) [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) |
| mq\_timedsend(GLIBC\_2.3.4) [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) | mq\_unlink(GLIBC\_2.3.4) [[SUSv4]](#ID_REFSTD_46_LIBRT_46_2) |  |  |

## **14.19 Data Definitions for librt**

This section defines global identifiers and their values that are associated with interfaces contained in librt. These definitions are organized into groups that correspond to system headers. This convention is used as a convenience for the reader, and does not imply the existence of these headers, or their content. Where an interface is defined as requiring a particular system header file all of the data definitions for that system header file presented here shall be in effect.

This section gives data definitions to promote binary application portability, not to repeat source interface definitions available elsewhere. System providers and application developers should use this ABI to supplement - not to replace - source interface definition specifications.

This specification uses the [ISO C (1999)](#ID_STD_46_ISOC99) C Language as the reference programming language, and data definitions are specified in ISO C format. The C language is used here as a convenient notation. Using a C language description of these data objects does not preclude their use by other programming languages.

### **14.19.1 aio.h**

#define AIO\_CANCELED 0

#define AIO\_NOTCANCELED 1

#define AIO\_ALLDONE 2

#define LIO\_READ 0

#define LIO\_WRITE 1

#define LIO\_NOP 2

#define LIO\_WAIT 0

#define LIO\_NOWAIT 1

struct aiocb {

int aio\_fildes; /\* File desriptor \*/

int aio\_lio\_opcode; /\* Operation to be performed \*/

int aio\_reqprio; /\* Request priority offset \*/

void \*aio\_buf; /\* Location of buffer \*/

size\_t aio\_nbytes; /\* Length of transfer \*/

struct sigevent aio\_sigevent; /\* Signal number and value \*/

struct aiocb \*\_\_next\_prio; /\* internal, do not use \*/

int \_\_abs\_prio; /\* internal, do not use \*/

int \_\_policy; /\* internal, do not use \*/

int \_\_error\_code; /\* internal, do not use \*/

ssize\_t \_\_return\_value; /\* internal, do not use \*/

off\_t aio\_offset; /\* File offset \*/

char \_\_pad[sizeof(off64\_t) - sizeof(off\_t)];

char \_\_unused[32];

};

struct aiocb64 {

int aio\_fildes; /\* File desriptor \*/

int aio\_lio\_opcode; /\* Operation to be performed \*/

int aio\_reqprio; /\* Request priority offset \*/

void \*aio\_buf; /\* Location of buffer \*/

size\_t aio\_nbytes; /\* Length of transfer \*/

struct sigevent aio\_sigevent; /\* Signal number and value \*/

struct aiocb \*\_\_next\_prio; /\* internal, do not use \*/

int \_\_abs\_prio; /\* internal, do not use \*/

int \_\_policy; /\* internal, do not use \*/

int \_\_error\_code; /\* internal, do not use \*/

ssize\_t \_\_return\_value; /\* internal, do not use \*/

off64\_t aio\_offset; /\* File offset \*/

char \_\_unused[32];

};

extern int aio\_cancel(int fildes, struct aiocb \*aiocbp);

extern int aio\_cancel64(int fildes, struct aiocb64 \*aiocbp);

extern int aio\_error(struct aiocb \*aiocbp);

extern int aio\_error64(struct aiocb64 \*aiocbp);

extern int aio\_fsync(int operation, struct aiocb \*aiocbp);

extern int aio\_fsync64(int operation, struct aiocb64 \*aiocbp);

extern int aio\_read(struct aiocb \*aiocbp);

extern int aio\_read64(struct aiocb64 \*aiocbp);

extern int aio\_return(struct aiocb \*aiocbp);

extern int aio\_return64(struct aiocb64 \*aiocbp);

extern int aio\_suspend(struct aiocb \*list[], int nent,

struct timespec \*timeout);

extern int aio\_suspend64(struct aiocb64 \*list[], int nent,

struct timespec \*timeout);

extern int aio\_write(struct aiocb \*aiocbp);

extern int aio\_write64(struct aiocb64 \*aiocbp);

extern int lio\_listio(int mode, struct aiocb \*list[], int nent,

struct sigevent \*sig);

extern int lio\_listio64(int mode, struct aiocb64 \*list[], int nent,

struct sigevent \*sig);

### **14.19.2 mqueue.h**

typedef int mqd\_t;

struct mq\_attr {

long int mq\_flags;

long int mq\_maxmsg;

long int mq\_msgsize;

long int mq\_curmsgs;

long int \_\_pad[4];

};

extern int mq\_close(mqd\_t \_\_mqdes);

extern int mq\_getattr(mqd\_t \_\_mqdes, struct mq\_attr \*\_\_mqstat);

extern int mq\_notify(mqd\_t \_\_mqdes, const struct sigevent \*\_\_notification);

extern mqd\_t mq\_open(const char \*\_\_name, int \_\_oflag, ...);

extern ssize\_t mq\_receive(mqd\_t \_\_mqdes, char \*\_\_msg\_ptr, size\_t \_\_msg\_len,

unsigned int \*\_\_msg\_prio);

extern int mq\_send(mqd\_t \_\_mqdes, const char \*\_\_msg\_ptr, size\_t \_\_msg\_len,

unsigned int \_\_msg\_prio);

extern int mq\_setattr(mqd\_t \_\_mqdes, const struct mq\_attr \*\_\_mqstat,

struct mq\_attr \*\_\_omqstat);

extern ssize\_t mq\_timedreceive(mqd\_t \_\_mqdes, char \*\_\_msg\_ptr,

size\_t \_\_msg\_len, unsigned int \*\_\_msg\_prio,

const struct timespec \*\_\_abs\_timeout);

extern int mq\_timedsend(mqd\_t \_\_mqdes, const char \*\_\_msg\_ptr,

size\_t \_\_msg\_len, unsigned int \_\_msg\_prio,

const struct timespec \*\_\_abs\_timeout);

extern int mq\_unlink(const char \*\_\_name);

## **14.20 Interfaces for libcrypt**

[Table 14-65](#ID_LIB_45_LIBCRYPT_45_DEF) defines the library name and shared object name for the libcrypt library

**Table 14-65 libcrypt Definition**

|  |  |
| --- | --- |
| Library: | libcrypt |
| SONAME: | libcrypt.so.1 |

The behavior of the interfaces in this library is specified by the following specifications:

|  |
| --- |
| [LSB] [This Specification](#ID_STD_46_LSB) |
| [SUSv4] [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) |

### **14.20.1 Encryption**

#### 14.20.1.1 Interfaces for Encryption

An LSB conforming implementation shall provide the generic functions for Encryption specified in [Table 14-66](#ID_TBL_45_LIBCRYPT_45_ENCRY_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-66 libcrypt - Encryption Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| crypt [[SUSv4]](#ID_REFSTD_46_LIBCRYPT_46_2) | crypt\_r [[LSB]](#ID_REFSTD_46_LIBCRYPT_46_1) | encrypt [[SUSv4]](#ID_REFSTD_46_LIBCRYPT_46_2) | encrypt\_r [[LSB]](#ID_REFSTD_46_LIBCRYPT_46_1) |
| setkey [[SUSv4]](#ID_REFSTD_46_LIBCRYPT_46_2) | setkey\_r [[LSB]](#ID_REFSTD_46_LIBCRYPT_46_1) |  |  |

## **14.21 Data Definitions for libcrypt**

This section defines global identifiers and their values that are associated with interfaces contained in libcrypt. These definitions are organized into groups that correspond to system headers. This convention is used as a convenience for the reader, and does not imply the existence of these headers, or their content. Where an interface is defined as requiring a particular system header file all of the data definitions for that system header file presented here shall be in effect.

This section gives data definitions to promote binary application portability, not to repeat source interface definitions available elsewhere. System providers and application developers should use this ABI to supplement - not to replace - source interface definition specifications.

This specification uses the [ISO C (1999)](#ID_STD_46_ISOC99) C Language as the reference programming language, and data definitions are specified in ISO C format. The C language is used here as a convenient notation. Using a C language description of these data objects does not preclude their use by other programming languages.

### **14.21.1 crypt.h**

struct crypt\_data {

char keysched[128];

char sb0[32768];

char sb1[32768];

char sb2[32768];

char sb3[32768];

char crypt\_3\_buf[14];

char current\_salt[2];

long int current\_saltbits;

int direction;

int initialized;

};

extern char \*crypt\_r(const char \*key, const char \*salt,

struct crypt\_data \*data);

extern void encrypt\_r(const char \*block, int edflag,

struct crypt\_data \*data);

extern void setkey\_r(const char \*key, struct crypt\_data \*data);

## **14.22 Interface Definitions for libcrypt**

The interfaces defined on the following pages are included in libcrypt and are defined by this specification. Unless otherwise noted, these interfaces shall be included in the source standard.

Other interfaces listed in [Section 14.20](#ID_LIBCRYPT) shall behave as described in the referenced base document.

#### crypt\_r

##### Name

crypt\_r — Cryptographic string encoding function

##### Synopsis

#include <crypt.h>

char \* crypt\_r(const char \* *key*, const char \* *salt*, struct crypt\_data \* *data*);

##### Description

The crypt\_r() function is a re-entrant version of the crypt() function. crypt\_r() shall behave as specified for crypt() in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but with an additional parameter, a pointer to a structure which is used to store result data and bookkeeping information.

The caller should set the *initialized* field of the crypt\_data structure to zero before the first call to crypt\_r().

##### Notes

INSERT TEXT HERE

##### See Also

crypt(), setkey\_r(), encrypt\_r().

#### encrypt\_r

##### Name

encrypt\_r — Cryptographic encoding function

##### Synopsis

#include <crypt.h>

void encrypt\_r(const char \* *block*, int *edflag*, struct crypt\_data \* *data*);

##### Description

The encrypt\_r() function is a re-entrant version of the encrypt() function. encrypt\_r() shall behave as specified for encrypt() in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but with an additional parameter, a pointer to a structure which is used to store result data and bookkeeping information.

##### Notes

INSERT TEXT HERE

##### See Also

encrypt(), crypt\_r(), setkey\_r().

#### setkey\_r

##### Name

setkey\_r — Set cryptographic encoding key

##### Synopsis

#include <crypt.h>

void setkey\_r(const char \* *key*, struct crypt\_data \* *data*);

##### Description

The setkey\_r() function is a re-entrant version of the setkey() function. setkey\_r() shall behave as specified for setkey() in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but with an additional parameter, a pointer to a structure which is used to store result data and bookkeeping information.

The caller should set the *initialized* field of the crypt\_data structure to zero before the first call to setkey\_r().

##### Notes

INSERT TEXT HERE

##### See Also

setkey(), crypt\_r(), encrypt\_r().

## **14.23 Interfaces for libpam**

[Table 14-67](#ID_LIB_45_LIBPAM_45_DEF) defines the library name and shared object name for the libpam library

**Table 14-67 libpam Definition**

|  |  |
| --- | --- |
| Library: | libpam |
| SONAME: | libpam.so.0 |

The Pluggable Authentication Module (PAM) interfaces allow applications to request authentication via a system administrator defined mechanism, known as a *service*.

A single service name, other, shall always be present. The behavior of this service shall be determined by the system administrator. Additional service names may also exist.

**Note:** Future versions of this specification might define additional service names.

The behavior of the interfaces in this library is specified by the following specifications:

|  |
| --- |
| [LSB] [This Specification](#ID_STD_46_LSB) |
| [PAM] [PAM](#ID_STD_46_PAM) |

### **14.23.1 Pluggable Authentication API**

#### 14.23.1.1 Interfaces for Pluggable Authentication API

An LSB conforming implementation shall provide the generic functions for Pluggable Authentication API specified in [Table 14-68](#ID_TBL_45_LIBPAM_45_PLUGG_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 14-68 libpam - Pluggable Authentication API Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| pam\_acct\_mgmt(LIBPAM\_1.0) [[LSB]](#ID_REFSTD_46_LIBPAM_46_1) | pam\_authenticate(LIBPAM\_1.0) [[LSB]](#ID_REFSTD_46_LIBPAM_46_1) | pam\_chauthtok(LIBPAM\_1.0) [[LSB]](#ID_REFSTD_46_LIBPAM_46_1) | pam\_close\_session(LIBPAM\_1.0) [[LSB]](#ID_REFSTD_46_LIBPAM_46_1) |
| pam\_end(LIBPAM\_1.0) [[LSB]](#ID_REFSTD_46_LIBPAM_46_1) | pam\_fail\_delay(LIBPAM\_1.0) [[LSB]](#ID_REFSTD_46_LIBPAM_46_1) | pam\_get\_data(LIBPAM\_1.0) [[PAM]](#ID_REFSTD_46_LIBPAM_46_2) | pam\_get\_item(LIBPAM\_1.0) [[LSB]](#ID_REFSTD_46_LIBPAM_46_1) |
| pam\_get\_user(LIBPAM\_1.0) [[PAM]](#ID_REFSTD_46_LIBPAM_46_2) | pam\_getenv(LIBPAM\_1.0) [[LSB]](#ID_REFSTD_46_LIBPAM_46_1) | pam\_getenvlist(LIBPAM\_1.0) [[LSB]](#ID_REFSTD_46_LIBPAM_46_1) | pam\_open\_session(LIBPAM\_1.0) [[LSB]](#ID_REFSTD_46_LIBPAM_46_1) |
| pam\_putenv(LIBPAM\_1.0) [[LSB]](#ID_REFSTD_46_LIBPAM_46_1) | pam\_set\_data(LIBPAM\_1.0) [[PAM]](#ID_REFSTD_46_LIBPAM_46_2) | pam\_set\_item(LIBPAM\_1.0) [[LSB]](#ID_REFSTD_46_LIBPAM_46_1) | pam\_setcred(LIBPAM\_1.0) [[LSB]](#ID_REFSTD_46_LIBPAM_46_1) |
| pam\_start(LIBPAM\_1.0) [[LSB]](#ID_REFSTD_46_LIBPAM_46_1) | pam\_strerror(LIBPAM\_1.0) [[LSB]](#ID_REFSTD_46_LIBPAM_46_1) |  |  |

## **14.24 Data Definitions for libpam**

This section defines global identifiers and their values that are associated with interfaces contained in libpam. These definitions are organized into groups that correspond to system headers. This convention is used as a convenience for the reader, and does not imply the existence of these headers, or their content. Where an interface is defined as requiring a particular system header file all of the data definitions for that system header file presented here shall be in effect.

This section gives data definitions to promote binary application portability, not to repeat source interface definitions available elsewhere. System providers and application developers should use this ABI to supplement - not to replace - source interface definition specifications.

This specification uses the [ISO C (1999)](#ID_STD_46_ISOC99) C Language as the reference programming language, and data definitions are specified in ISO C format. The C language is used here as a convenient notation. Using a C language description of these data objects does not preclude their use by other programming languages.

### **14.24.1 security/\_pam\_types.h**

typedef struct pam\_handle pam\_handle\_t;

struct pam\_message {

int msg\_style;

const char \*msg;

};

struct pam\_response {

char \*resp;

int resp\_retcode; /\* currently un-used, zero expected \*/

};

struct pam\_conv {

int (\*conv) (int num\_msg, const struct pam\_message \* \*msg,

struct pam\_response \* \*resp, void \*appdata\_ptr);

void \*appdata\_ptr;

};

#define PAM\_PROMPT\_ECHO\_OFF 1

#define PAM\_PROMPT\_ECHO\_ON 2

#define PAM\_ERROR\_MSG 3

#define PAM\_TEXT\_INFO 4

#define PAM\_SERVICE 1 /\* The service name \*/

#define PAM\_USER 2 /\* The user name \*/

#define PAM\_TTY 3 /\* The tty name \*/

#define PAM\_RHOST 4 /\* The remote host name \*/

#define PAM\_CONV 5 /\* The pam\_conv structure \*/

#define PAM\_RUSER 8 /\* The remote user name \*/

#define PAM\_USER\_PROMPT 9 /\* the prompt for getting a username \*/

#define PAM\_SUCCESS 0 /\* Successful function return \*/

#define PAM\_OPEN\_ERR 1 /\* dlopen() failure \*/

#define PAM\_USER\_UNKNOWN 10 /\* User not known to the underlying authenticaiton module \*/

#define PAM\_MAXTRIES 11 /\* An authentication service has maintained a retry count which \*/

#define PAM\_NEW\_AUTHTOK\_REQD 12 /\* New authentication token required \*/

#define PAM\_ACCT\_EXPIRED 13 /\* User account has expired \*/

#define PAM\_SESSION\_ERR 14 /\* Can not make/remove an entry for the specified session \*/

#define PAM\_CRED\_UNAVAIL 15 /\* Underlying authentication service can not retrieve user cred \*/

#define PAM\_CRED\_EXPIRED 16 /\* User credentials expired \*/

#define PAM\_CRED\_ERR 17 /\* Failure setting user credentials \*/

#define PAM\_CONV\_ERR 19 /\* Conversation error \*/

#define PAM\_SYMBOL\_ERR 2 /\* Symbol not found \*/

#define PAM\_AUTHTOK\_ERR 20 /\* Authentication token manipulation error \*/

#define PAM\_AUTHTOK\_RECOVER\_ERR 21 /\* Authentication information cannot be recovered \*/

#define PAM\_AUTHTOK\_LOCK\_BUSY 22 /\* Authentication token lock busy \*/

#define PAM\_AUTHTOK\_DISABLE\_AGING 23 /\* Authentication token aging disabled \*/

#define PAM\_TRY\_AGAIN 24 /\* Preliminary check by password service \*/

#define PAM\_ABORT 26 /\* Critical error (?module fail now request) \*/

#define PAM\_AUTHTOK\_EXPIRED 27 /\* user's authentication token has expired \*/

#define PAM\_BAD\_ITEM 29 /\* Bad item passed to pam\_\*\_item() \*/

#define PAM\_SERVICE\_ERR 3 /\* Error in service module \*/

#define PAM\_SYSTEM\_ERR 4 /\* System error \*/

#define PAM\_BUF\_ERR 5 /\* Memory buffer error \*/

#define PAM\_PERM\_DENIED 6 /\* Permission denied \*/

#define PAM\_AUTH\_ERR 7 /\* Authentication failure \*/

#define PAM\_CRED\_INSUFFICIENT 8 /\* Can not access authentication data due to insufficient crede \*/

#define PAM\_AUTHINFO\_UNAVAIL 9 /\* Underlying authentication service can not retrieve authentic \*/

#define PAM\_DISALLOW\_NULL\_AUTHTOK 0x0001U

#define PAM\_ESTABLISH\_CRED 0x0002U /\* Set user credentials for an authentication service \*/

#define PAM\_DELETE\_CRED 0x0004U /\* Delete user credentials associated with an authentication se \*/

#define PAM\_REINITIALIZE\_CRED 0x0008U /\* Reinitialize user credentials \*/

#define PAM\_REFRESH\_CRED 0x0010U /\* Extend lifetime of user credentials \*/

#define PAM\_CHANGE\_EXPIRED\_AUTHTOK 0x0020U /\* Extend lifetime of user credentials \*/

#define PAM\_SILENT 0x8000U /\* Authentication service should not generate any messages \*/

extern int pam\_fail\_delay(pam\_handle\_t \*, unsigned int);

extern int pam\_get\_item(const pam\_handle\_t \*, int, const void \*\*);

extern const char \*pam\_getenv(pam\_handle\_t \*, const char \*);

extern char \*\*pam\_getenvlist(pam\_handle\_t \*);

extern int pam\_putenv(pam\_handle\_t \*, const char \*);

extern int pam\_set\_item(pam\_handle\_t \*, int, const void \*);

extern const char \*pam\_strerror(pam\_handle\_t \*, int);

### **14.24.2 security/pam\_appl.h**

extern int pam\_acct\_mgmt(pam\_handle\_t \*, int);

extern int pam\_authenticate(pam\_handle\_t \*, int);

extern int pam\_chauthtok(pam\_handle\_t \*, int);

extern int pam\_close\_session(pam\_handle\_t \*, int);

extern int pam\_end(pam\_handle\_t \*, int);

extern int pam\_open\_session(pam\_handle\_t \*, int);

extern int pam\_setcred(pam\_handle\_t \*, int);

extern int pam\_start(const char \*, const char \*, const struct pam\_conv \*,

pam\_handle\_t \* \*);

### **14.24.3 security/pam\_modules.h**

extern int pam\_get\_data(const pam\_handle\_t \*, const char \*, const void \*\*);

extern int pam\_get\_user(pam\_handle\_t \*, const char \*\*, const char \*);

extern int pam\_set\_data(pam\_handle\_t \*, const char \*, void \*,

void (\*)(pam\_handle\_t \*, void \*, int));

## **14.25 Interface Definitions for libpam**

The interfaces defined on the following pages are included in libpam and are defined by this specification. Unless otherwise noted, these interfaces shall be included in the source standard.

Other interfaces listed in [Section 14.23](#ID_LIBPAM) shall behave as described in the referenced base document.

#### pam\_acct\_mgmt

##### Name

pam\_acct\_mgmt — establish the status of a user's account

##### Synopsis

#include <security/pam\_appl.h>

int pam\_acct\_mgmt(pam\_handle\_t \* *pamh*, int *flags*);

##### Description

pam\_acct\_mgmt() establishes the account's usability and the user's accessibility to the system. It is typically called after the user has been authenticated.

*flags* may be specified as any valid flag (namely, one of those applicable to the *flags* argument of pam\_authenticate()). Additionally, the value of *flags* may be logically or'd with PAM\_SILENT.

##### Return Value

PAM\_SUCCESS

  Success.

PAM\_NEW\_AUTHTOK\_REQD

  User is valid, but user's authentication token has expired. The correct response to this return-value is to require that the user satisfy the pam\_chauthtok() function before obtaining service. It may not be possible for an application to do this. In such a case, the user should be denied access until the account password is updated.

PAM\_ACCT\_EXPIRED

  User is no longer permitted access to the system.

PAM\_AUTH\_ERR

  Authentication error.

PAM\_PERM\_DENIED

  User is not permitted to gain access at this time.

PAM\_USER\_UNKNOWN

  User is not known to a module's account management component.

**Note:** Errors may be translated to text with pam\_strerror().

#### pam\_authenticate

##### Name

pam\_authenticate — authenticate the user

##### Synopsis

#include <security/pam\_appl.h>

int pam\_authenticate(pam\_handle\_t \* *pamh*, int *flags*);

##### Description

pam\_authenticate() serves as an interface to the authentication mechanisms of the loaded modules.

*flags* is an optional parameter that may be specified by the following value:

PAM\_DISALLOW\_NULL\_AUTHTOK

  Instruct the authentication modules to return PAM\_AUTH\_ERR if the user does not have a registered authorization token.

Additionally, the value of *flags* may be logically or'd with PAM\_SILENT.

The process may need to be privileged in order to successfully call this function.

##### Return Value

PAM\_SUCCESS

  Success.

PAM\_AUTH\_ERR

  User was not authenticated or process did not have sufficient privileges to perform authentication.

PAM\_CRED\_INSUFFICIENT

  Application does not have sufficient credentials to authenticate the user.

PAM\_AUTHINFO\_UNAVAIL

  Modules were not able to access the authentication information. This might be due to a network or hardware failure, etc.

PAM\_USER\_UNKNOWN

  Supplied username is not known to the authentication service.

PAM\_MAXTRIES

  One or more authentication modules has reached its limit of tries authenticating the user. Do not try again.

PAM\_ABORT

  One or more authentication modules failed to load.

**Note:** Errors may be translated to text with pam\_strerror().

#### pam\_chauthtok

##### Name

pam\_chauthtok — change the authentication token for a given user

##### Synopsis

#include <security/pam\_appl.h>

int pam\_chauthtok(pam\_handle\_t \* *pamh*, const int *flags*);

##### Description

pam\_chauthtok() is used to change the authentication token for a given user as indicated by the state associated with the handle *pamh*.

*flags* is an optional parameter that may be specified by the following value:

PAM\_CHANGE\_EXPIRED\_AUTHTOK

  User's authentication token should only be changed if it has expired.

Additionally, the value of *flags* may be logically or'd with PAM\_SILENT.

##### Return Value

PAM\_SUCCESS

  Success.

PAM\_AUTHTOK\_ERR

  A module was unable to obtain the new authentication token.

PAM\_AUTHTOK\_RECOVER\_ERR

  A module was unable to obtain the old authentication token.

PAM\_AUTHTOK\_LOCK\_BUSY

  One or more modules were unable to change the authentication token since it is currently locked.

PAM\_AUTHTOK\_DISABLE\_AGING

  Authentication token aging has been disabled for at least one of the modules.

PAM\_PERM\_DENIED

  Permission denied.

PAM\_TRY\_AGAIN

  Not all modules were in a position to update the authentication token(s). In such a case, none of the user's authentication tokens are updated.

PAM\_USER\_UNKNOWN

  User is not known to the authentication token changing service.

**Note:** Errors may be translated to text with pam\_strerror().

#### pam\_close\_session

##### Name

pam\_close\_session — indicate that an authenticated session has ended

##### Synopsis

#include <security/pam\_appl.h>

int pam\_close\_session(pam\_handle\_t \* *pamh*, int *flags*);

##### Description

pam\_close\_session() is used to indicate that an authenticated session has ended. It is used to inform the module that the user is exiting a session. It should be possible for the PAM library to open a session and close the same session from different applications.

*flags* may have the value PAM\_SILENT to indicate that no output should be generated as a result of this function call.

##### Return Value

PAM\_SUCCESS

  Success.

PAM\_SESSION\_ERR

  One of the required loaded modules was unable to close a session for the user.

**Note:** Errors may be translated to text with pam\_strerror().

#### pam\_end

##### Name

pam\_end — terminate the use of the PAM library

##### Synopsis

#include <security/pam\_appl.h>

int pam\_end(pam\_handle\_t \* *pamh*, int *pam\_status*);

##### Description

pam\_end() terminates use of the PAM library. On success, the contents of *\*pamh* are no longer valid, and all memory associated with it is invalid.

Normally, *pam\_status* is passed the value PAM\_SUCCESS, but in the event of an unsuccessful service application, the appropriate PAM error return value should be used.

##### Return Value

PAM\_SUCCESS

  Success.

**Note:** Errors may be translated to text with pam\_strerror().

#### pam\_fail\_delay

##### Name

pam\_fail\_delay — specify delay time to use on authentication error

##### Synopsis

#include <security/pam\_appl.h>

int pam\_fail\_delay(pam\_handle\_t \* *pamh*, unsigned int *micro\_sec*);

##### Description

pam\_fail\_delay() specifies the minimum delay for the PAM library to use when an authentication error occurs. The actual delay can vary by as much at 25%. If this function is called multiple times, the longest time specified by any of the call will be used.

The delay is invoked if an authentication error occurs during the pam\_authenticate() or pam\_chauthtok() function calls.

Independent of the success of pam\_authenticate() or pam\_chauthtok(), the delay time is reset to its default value of 0 when the PAM library returns control to the application from these two functions.

##### Return Value

PAM\_SUCCESS

  Success.

**Note:** Errors may be translated to text with pam\_strerror().

#### pam\_get\_item

##### Name

pam\_get\_item — obtain the value of the indicated item.

##### Synopsis

#include <security/pam\_appl.h>

int pam\_get\_item(const pam\_handle\_t \* *pamh*, int *item\_type*, const void \* \* *item*);

##### Description

pam\_get\_item() obtains the value of the indicated *item\_type*. The possible values of *item\_type* are the same as listed for pam\_set\_item().

On success, *item* contains a pointer to the value of the corresponding item. Note that this is a pointer to the actual data and should not be free()'d or over-written.

##### Return Value

PAM\_SUCCESS

  Success.

PAM\_PERM\_DENIED

  Application passed a NULL pointer for item.

PAM\_BAD\_ITEM

  Application attempted to get an undefined item.

**Note:** Errors may be translated to text with pam\_strerror().

#### pam\_getenv

##### Name

pam\_getenv — get a PAM environment variable

##### Synopsis

#include <security/pam\_appl.h>

const char \* pam\_getenv(const pam\_handle\_t \* *pamh*, const char \* *name*);

##### Description

The pam\_getenv() function shall search the environment associated with the PAM handle *pamh* for the environment variable *name*. If the specified environment variable cannot be found, a null pointer shall be returned. The application shall ensure that it does not modify the string pointed to by the pam\_getenv() function.

##### Return Value

On success, pam\_getenv() returns a pointer to a string of the form name=value.

#### pam\_getenvlist

##### Name

pam\_getenvlist — returns a pointer to the complete PAM environment.

##### Synopsis

#include <security/pam\_appl.h>

char \* const \* pam\_getenvlist(pam\_handle\_t \* *pamh*);

##### Description

pam\_getenvlist() returns a pointer to the complete PAM environment. This pointer points to an array of pointers to NUL-terminated strings and must be terminated by a NULL pointer. Each string has the form "name=value".

The PAM library module allocates memory for the returned value and the associated strings. The calling application is responsible for freeing this memory.

##### Return Value

pam\_getenvlist() returns an array of string pointers containing the PAM environment. On error, NULL is returned.

#### pam\_open\_session

##### Name

pam\_open\_session — indicate session has started

##### Synopsis

#include <security/pam\_appl.h>

int pam\_open\_session(pam\_handle\_t \* *pamh*, int *flags*);

##### Description

The pam\_open\_session() function is used to indicate that an authenticated session has begun, after the user has been identified (see pam\_authenticate()) and, if necessary, granted credentials (see pam\_setcred()). It is used to inform the module that the user is currently in a session. It should be possible for the PAM library to open a session and close the same session from different applications.

*flags* may have the value PAM\_SILENT to indicate that no output be generated as a result of this function call.

##### Return Value

PAM\_SUCCESS

  Success.

PAM\_SESSION\_ERR

  One of the loaded modules was unable to open a session for the user.

**Note:** Errors may be translated to text with pam\_strerror().

#### pam\_putenv

##### Name

pam\_putenv — Add, replace or delete a PAM environment variable

##### Synopsis

#include <security/pam\_appl.h>

int pam\_putenv(const pam\_handle\_t \* *pamh*, const char \* *name\_value*);

##### Description

The pam\_putenv() function shall modify the environment list associated with *pamh*. If *name\_value* contains an '=' character, the characters to the left of the first '=' character represent the *name*, and the remaining characters after the '=' represent the *value*.

If the *name* environment variable exists in the environment associated with *pamh*, it shall be modified to have the value *value*. Otherwise, the *name* shall be added to the environment associated with *pamh* with the value *value*.

If there is no '=' character in *name\_value*, the variable in the environment associated with *pamh* named *name\_value* shall be deleted.

##### Return Value

On success, the pam\_putenv() function shall return PAM\_SUCCESS. Otherwise the return value indicates the error:

PAM\_PERM\_DENIED

  The *name\_value* argument is a null pointer.

PAM\_BAD\_ITEM

  The PAM environment varable named *name\_value* does not exist and therefore cannot be deleted.

PAM\_ABORT

  The PAM handle identifed by *pamh* is corrupt.

PAM\_BUF\_ERR

  Memory buffer error.

#### pam\_set\_item

##### Name

pam\_set\_item — (re)set the value of an item.

##### Synopsis

#include <security/pam\_appl.h>

int pam\_set\_item(pam\_handle\_t \* *pamh*, int *item\_type*, const void \* *item*);

##### Description

pam\_set\_item() (re)sets the value of one of the following item\_types:

PAM\_SERVICE

  service name

PAM\_USER

  user name

PAM\_TTY

  terminal name

The value for a device file should include the /dev/ prefix. The value for graphical, X-based, applications should be the $DISPLAY variable.

PAM\_RHOST

  remote host name

PAM\_CONV

  conversation structure

PAM\_RUSER

  remote user name

PAM\_USER\_PROMPT

  string to be used when prompting for a user's name

The default value for this string is Please enter username: .

For all *item\_types* other than PAM\_CONV, *item* is a pointer to a NULL-terminated character string. In the case of PAM\_CONV, *item* points to an initialized pam\_conv structure.

##### Return Value

PAM\_SUCCESS

  Success.

PAM\_PERM\_DENIED

  An attempt was made to replace the conversation structure with a NULL value.

PAM\_BUF\_ERR

  Function ran out of memory making a copy of the item.

PAM\_BAD\_ITEM

  Application attempted to set an undefined item.

**Note:** Errors may be translated to text with pam\_strerror().

#### pam\_setcred

##### Name

pam\_setcred — set the module-specific credentials of the user

##### Synopsis

#include <security/pam\_appl.h>

extern int pam\_setcred(pam\_handle\_t \* *pamh*, int *flags*);

##### Description

pam\_setcred() sets the module-specific credentials of the user. It is usually called after the user has been authenticated, after the account management function has been called and after a session has been opened for the user.

*flags* maybe specified from among the following values:

PAM\_ESTABLISH\_CRED

  set credentials for the authentication service

PAM\_DELETE\_CRED

  delete credentials associated with the authentication service

PAM\_REINITIALIZE\_CRED

  reinitialize the user credentials

PAM\_REFRESH\_CRED

  extend lifetime of the user credentials

Additionally, the value of *flags* may be logically or'd with PAM\_SILENT.

##### Return Value

PAM\_SUCCESS

  Success.

PAM\_CRED\_UNAVAIL

  Module cannot retrieve the user's credentials.

PAM\_CRED\_EXPIRED

  User's credentials have expired.

PAM\_USER\_UNKNOWN

  User is not known to an authentication module.

PAM\_CRED\_ERR

  Module was unable to set the credentials of the user.

**Note:** Errors may be translated to text with pam\_strerror().

#### pam\_start

##### Name

pam\_start — initialize the PAM library

##### Synopsis

#include <security/pam\_appl.h>

int pam\_start(const char \* *service\_name*, const char \* *user*, const struct pam\_conv \* *pam\_conversation*, pam\_handle\_t \* \* *pamh*);

##### Description

pam\_start() is used to initialize the PAM library. It must be called prior to any other usage of the PAM library. On success, *\*pamh* becomes a handle that provides continuity for successive calls to the PAM library. pam\_start() expects arguments as follows: the *service\_name* of the program, the *username* of the individual to be authenticated, a pointer to an application-supplied pam\_conv structure, and a pointer to a *pam\_handle\_t* pointer.

An application must provide the *conversation function* used for direct communication between a loaded module and the application. The application also typically provides a means for the module to prompt the user for a password, etc.

The structure, pam\_conv, is defined to be,

struct pam\_conv {

int (\*conv) (int num\_msg,

const struct pam\_message \* \*msg,

struct pam\_response \* \*resp,

void \*appdata\_ptr);

void \*appdata\_ptr;

};

It is initialized by the application before it is passed to the library. The contents of this structure are attached to the *\*pamh* handle. The point of this argument is to provide a mechanism for any loaded module to interact directly with the application program; this is why it is called a conversation structure.

When a module calls the referenced conv() function, *appdata\_ptr* is set to the second element of this structure.

The other arguments of a call to conv() concern the information exchanged by module and application. *num\_msg* holds the length of the array of pointers passed via *msg*. On success, the pointer *resp* points to an array of *num\_msg* pam\_response structures, holding the application-supplied text. Note that *resp* is a struct pam\_response array and not an array of pointers.

##### Return Value

PAM\_SUCCESS

  Success.

PAM\_BUF\_ERR

  Memory allocation error.

PAM\_ABORT

  Internal failure.

##### ERRORS

May be translated to text with pam\_strerror().

#### pam\_strerror

##### Name

pam\_strerror — returns a string describing the PAM error

##### Synopsis

#include <security/pam\_appl.h>

const char \* pam\_strerror(pam\_handle\_t \* *pamh*, int *errnum*);

##### Description

pam\_strerror() returns a string describing the PAM error associated with *errnum*.

##### Return Value

On success, this function returns a description of the indicated error. The application should not free or modify this string. Otherwise, a string indicating that the error is unknown shall be returned. It is unspecified whether or not the string returned is translated according to the setting of LC\_MESSAGES.

# **IV Utility Libraries**

# **15 Utility Libraries**

## **15.1 Introduction**

An LSB-conforming implementation shall also support the following utility libraries which are built on top of the interfaces provided by the base libraries. These libraries implement common functionality, and hide additional system dependent information such as file formats and device names.

• libz

• libncurses

• libncursesw

• libutil

The structure of the definitions for these libraries follows the same model as used for [Base Libraries](#ID_BASELIB).

## **15.2 Interfaces for libz**

[Table 15-1](#ID_LIB_45_LIBZ_45_DEF) defines the library name and shared object name for the libz library

**Table 15-1 libz Definition**

|  |  |
| --- | --- |
| Library: | libz |
| SONAME: | libz.so.1 |

The behavior of the interfaces in this library is specified by the following specifications:

|  |
| --- |
| [LSB] [This Specification](#ID_STD_46_LSB) |

### **15.2.1 Compression Library**

#### 15.2.1.1 Interfaces for Compression Library

An LSB conforming implementation shall provide the generic functions for Compression Library specified in [Table 15-2](#ID_TBL_45_LIBZ_45_COMPR_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 15-2 libz - Compression Library Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| adler32 [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | compress [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | compress2 [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | compressBound(ZLIB\_1.2.0) [[LSB]](#ID_REFSTD_46_LIBZ_46_1) |
| crc32 [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | deflate [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | deflateBound(ZLIB\_1.2.0) [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | deflateCopy [[LSB]](#ID_REFSTD_46_LIBZ_46_1) |
| deflateEnd [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | deflateInit2\_ [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | deflateInit\_ [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | deflateParams [[LSB]](#ID_REFSTD_46_LIBZ_46_1) |
| deflatePrime(ZLIB\_1.2.0.8) [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | deflateReset [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | deflateSetDictionary [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | get\_crc\_table [[LSB]](#ID_REFSTD_46_LIBZ_46_1) |
| gzclearerr(ZLIB\_1.2.0.2) [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | gzclose [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | gzdopen [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | gzeof [[LSB]](#ID_REFSTD_46_LIBZ_46_1) |
| gzerror [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | gzflush [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | gzgetc [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | gzgets [[LSB]](#ID_REFSTD_46_LIBZ_46_1) |
| gzopen [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | gzprintf [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | gzputc [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | gzputs [[LSB]](#ID_REFSTD_46_LIBZ_46_1) |
| gzread [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | gzrewind [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | gzseek [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | gzsetparams [[LSB]](#ID_REFSTD_46_LIBZ_46_1) |
| gztell [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | gzwrite [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | inflate [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | inflateBack(ZLIB\_1.2.0) [[LSB]](#ID_REFSTD_46_LIBZ_46_1) |
| inflateBackEnd(ZLIB\_1.2.0) [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | inflateBackInit\_(ZLIB\_1.2.0) [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | inflateCopy(ZLIB\_1.2.0) [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | inflateEnd [[LSB]](#ID_REFSTD_46_LIBZ_46_1) |
| inflateInit2\_ [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | inflateInit\_ [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | inflateReset [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | inflateSetDictionary [[LSB]](#ID_REFSTD_46_LIBZ_46_1) |
| inflateSync [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | inflateSyncPoint [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | uncompress [[LSB]](#ID_REFSTD_46_LIBZ_46_1) | zError [[LSB]](#ID_REFSTD_46_LIBZ_46_1) |
| zlibVersion [[LSB]](#ID_REFSTD_46_LIBZ_46_1) |  |  |  |

## **15.3 Data Definitions for libz**

This section defines global identifiers and their values that are associated with interfaces contained in libz. These definitions are organized into groups that correspond to system headers. This convention is used as a convenience for the reader, and does not imply the existence of these headers, or their content. Where an interface is defined as requiring a particular system header file all of the data definitions for that system header file presented here shall be in effect.

This section gives data definitions to promote binary application portability, not to repeat source interface definitions available elsewhere. System providers and application developers should use this ABI to supplement - not to replace - source interface definition specifications.

This specification uses the [ISO C (1999)](#ID_STD_46_ISOC99) C Language as the reference programming language, and data definitions are specified in ISO C format. The C language is used here as a convenient notation. Using a C language description of these data objects does not preclude their use by other programming languages.

### **15.3.1 zconf.h**

#define ZEXPORT

#define ZEXPORTVA

#define OF(args) args

#define ZEXTERN extern

### **15.3.2 zlib.h**

#define ZLIB\_VERSION "1.2.2"

#define Z\_NULL 0

#define MAX\_WBITS 15 /\* 32K LZ77 window \*/

#define MAX\_MEM\_LEVEL 9 /\* Maximum value for memLevel in deflateInit2 \*/

#define deflateInit2(strm,level,method,windowBits,memLevel,strategy) \

deflateInit2\_((strm),(level),(method),(windowBits),(memLevel),(strategy),ZLIB\_VERSION,sizeof(z\_stream))

#define deflateInit(strm,level) \

deflateInit\_((strm), (level), ZLIB\_VERSION, sizeof(z\_stream))

#define inflateInit2(strm,windowBits) \

inflateInit2\_((strm), (windowBits), ZLIB\_VERSION, sizeof(z\_stream))

#define inflateInit(strm) \

inflateInit\_((strm), ZLIB\_VERSION, sizeof(z\_stream))

#define inflateBackInit(strm, windowBits, window) \

inflateBackInit\_((strm), (windowBits), (window), \

ZLIB\_VERSION, sizeof(z\_stream))

typedef char charf;

typedef int intf;

typedef void \*voidpf;

typedef unsigned int uInt;

typedef unsigned long int uLong;

typedef uLong uLongf;

typedef void \*voidp;

typedef unsigned char Byte;

typedef off\_t z\_off\_t;

typedef void \*const voidpc;

typedef voidpf(\*alloc\_func) (voidpf opaque, uInt items, uInt size);

typedef void (\*free\_func) (voidpf opaque, voidpf address);

struct internal\_state {

int dummy;

};

typedef Byte Bytef;

typedef uInt uIntf;

typedef unsigned int (\*in\_func) (void \*, unsigned char \*\*);

typedef int (\*out\_func) (void \*, unsigned char \*, unsigned int);

typedef struct z\_stream\_s {

Bytef \*next\_in; /\* next input byte \*/

uInt avail\_in; /\* number of bytes available at next\_in \*/

uLong total\_in; /\* total nb of input bytes read so far \*/

Bytef \*next\_out; /\* next output byte should be put there \*/

uInt avail\_out; /\* remaining free space at next\_out \*/

uLong total\_out; /\* total nb of bytes output so far \*/

char \*msg; /\* last error message, NULL if no error \*/

struct internal\_state \*state; /\* not visible by applications \*/

alloc\_func zalloc; /\* used to allocate the internal state \*/

free\_func zfree; /\* used to free the internal state \*/

voidpf opaque; /\* private data object passed to zalloc and zfree \*/

int data\_type; /\* best guess about the data type: ascii or binary \*/

uLong adler; /\* adler32 value of the uncompressed data \*/

uLong reserved; /\* reserved for future use \*/

} z\_stream;

typedef z\_stream \*z\_streamp;

typedef voidp gzFile;

#define Z\_NO\_FLUSH 0

#define Z\_PARTIAL\_FLUSH 1

#define Z\_SYNC\_FLUSH 2

#define Z\_FULL\_FLUSH 3

#define Z\_FINISH 4

#define Z\_BLOCK 5

#define Z\_ERRNO (-1)

#define Z\_STREAM\_ERROR (-2)

#define Z\_DATA\_ERROR (-3)

#define Z\_MEM\_ERROR (-4)

#define Z\_BUF\_ERROR (-5)

#define Z\_VERSION\_ERROR (-6)

#define Z\_OK 0

#define Z\_STREAM\_END 1

#define Z\_NEED\_DICT 2

#define Z\_DEFAULT\_COMPRESSION (-1)

#define Z\_NO\_COMPRESSION 0

#define Z\_BEST\_SPEED 1

#define Z\_BEST\_COMPRESSION 9

#define Z\_DEFAULT\_STRATEGY 0

#define Z\_FILTERED 1

#define Z\_HUFFMAN\_ONLY 2

#define Z\_BINARY 0

#define Z\_ASCII 1

#define Z\_UNKNOWN 2

#define Z\_DEFLATED 8

extern uLong adler32(uLong adler, const Bytef \* buf, uInt len);

extern int compress(Bytef \* dest, uLongf \* destLen, const Bytef \* source,

uLong sourceLen);

extern int compress2(Bytef \* dest, uLongf \* destLen, const Bytef \* source,

uLong sourceLen, int level);

extern uLong compressBound(uLong sourceLen);

extern uLong crc32(uLong crc, const Bytef \* buf, uInt len);

extern int deflate(z\_streamp strm, int flush);

extern uLong deflateBound(z\_streamp strm, uLong sourceLen);

extern int deflateCopy(z\_streamp dest, z\_streamp source);

extern int deflateEnd(z\_streamp strm);

extern int deflateInit2\_(z\_streamp strm, int level, int method,

int windowBits, int memLevel, int strategy,

const char \*version, int stream\_size);

extern int deflateInit\_(z\_streamp strm, int level, const char \*version,

int stream\_size);

extern int deflateParams(z\_streamp strm, int level, int strategy);

extern int deflatePrime(z\_streamp strm, int bits, int value);

extern int deflateReset(z\_streamp strm);

extern int deflateSetDictionary(z\_streamp strm, const Bytef \* dictionary,

uInt dictLength);

extern const uLongf \*get\_crc\_table(void);

extern void gzclearerr(gzFile file);

extern int gzclose(gzFile file);

extern gzFile gzdopen(int fd, const char \*mode);

extern int gzeof(gzFile file);

extern const char \*gzerror(gzFile file, int \*errnum);

extern int gzflush(gzFile file, int flush);

extern int gzgetc(gzFile file);

extern char \*gzgets(gzFile file, char \*buf, int len);

extern gzFile gzopen(const char \*path, const char \*mode);

extern int gzprintf(gzFile file, const char \*format, ...);

extern int gzputc(gzFile file, int c);

extern int gzputs(gzFile file, const char \*s);

extern int gzread(gzFile file, voidp buf, unsigned int len);

extern int gzrewind(gzFile file);

extern z\_off\_t gzseek(gzFile file, z\_off\_t offset, int whence);

extern int gzsetparams(gzFile file, int level, int strategy);

extern z\_off\_t gztell(gzFile file);

extern int gzwrite(gzFile file, voidpc buf, unsigned int len);

extern int inflate(z\_streamp strm, int flush);

extern int inflateBack(z\_streamp strm, in\_func in, void \*in\_desc,

out\_func out, void \*out\_desc);

extern int inflateBackEnd(z\_streamp strm);

extern int inflateBackInit\_(z\_streamp strm, int windowBits,

unsigned char \*window, const char \*version,

int stream\_size);

extern int inflateCopy(z\_streamp dest, z\_streamp source);

extern int inflateEnd(z\_streamp strm);

extern int inflateInit2\_(z\_streamp strm, int windowBits,

const char \*version, int stream\_size);

extern int inflateInit\_(z\_streamp strm, const char \*version,

int stream\_size);

extern int inflateReset(z\_streamp strm);

extern int inflateSetDictionary(z\_streamp strm, const Bytef \* dictionary,

uInt dictLength);

extern int inflateSync(z\_streamp strm);

extern int inflateSyncPoint(z\_streamp z);

extern int uncompress(Bytef \* dest, uLongf \* destLen, const Bytef \* source,

uLong sourceLen);

extern const char \*zError(int);

extern const char \*zlibVersion(void);

## **15.4 Interface Definitions for libz**

The interfaces defined on the following pages are included in libz and are defined by this specification. Unless otherwise noted, these interfaces shall be included in the source standard.

Other interfaces listed in [Section 15.2](#ID_LIBZ) shall behave as described in the referenced base document.

#### adler32

##### Name

adler32 — compute Adler 32 Checksum

##### Synopsis

#include <zlib.h>

uLong adler32(uLong *adler*, const Bytef \* *buf*, uInt *len*);

##### Description

The adler32() function shall compute a running Adler-32 checksum (as described in [RFC 1950: ZLIB Compressed Data Format Specication](#ID_STD_46_RFC1950)). On entry, *adler* is the previous value for the checksum, and *buf* shall point to an array of *len* bytes of data to be added to this checksum. The adler32() function shall return the new checksum.

If *buf* is NULL (or Z\_NULL), adler32() shall return the initial checksum.

##### Return Value

The adler32() function shall return the new checksum value.

##### Errors

None defined.

##### Application Usage (informative)

The following code fragment demonstrates typical usage of the adler32() function:

uLong adler = adler32(0L, Z\_NULL, 0);

while (read\_buffer(buffer, length) != EOF) {

adler = adler32(adler, buffer, length);

}

if (adler != original\_adler) error();

#### compress

##### Name

compress — compress data

##### Synopsis

#include <zlib.h>

int compress(Bytef \* *dest*, uLongf \* *destLen*, const Bytef \* *source*, uLong *sourceLen*);

##### Description

The compress() function shall attempt to compress *sourceLen* bytes of data in the buffer *source*, placing the result in the buffer *dest*.

On entry, *destLen* should point to a value describing the size of the *dest* buffer. The application should ensure that this value be at least (sourceLen × 1.001) + 12. On successful exit, the variable referenced by *destLen* shall be updated to hold the length of compressed data in *dest*.

The compress() function is equivalent to compress2() with a *level* of Z\_DEFAULT\_COMPRESSION.

##### Return Value

On success, compress() shall return Z\_OK. Otherwise, compress() shall return a value to indicate the error.

##### Errors

On error, compress() shall return a value as described below:

Z\_BUF\_ERROR

  The buffer *dest* was not large enough to hold the compressed data.

Z\_MEM\_ERROR

  Insufficient memory.

#### compress2

##### Name

compress2 — compress data at a specified level

##### Synopsis

#include <zlib.h>

int compress2(Bytef \* *dest*, uLongf \* *destLen*, const Bytef \* *source*, uLong *sourceLen*, int *level*);

##### Description

The compress2() function shall attempt to compress *sourceLen* bytes of data in the buffer *source*, placing the result in the buffer *dest*, at the level described by *level*. The *level* supplied shall be a value between 0 and 9, or the value Z\_DEFAULT\_COMPRESSION. A *level* of 1 requests the highest speed, while a *level* of 9 requests the highest compression. A *level* of 0 indicates that no compression should be used, and the output shall be the same as the input.

On entry, *destLen* should point to a value describing the size of the *dest* buffer. The application should ensure that this value be at least (sourceLen × 1.001) + 12. On successful exit, the variable referenced by *destLen* shall be updated to hold the length of compressed data in *dest*.

The compress() function is equivalent to compress2() with a *level* of Z\_DEFAULT\_COMPRESSION.

##### Return Value

On success, compress2() shall return Z\_OK. Otherwise, compress2() shall return a value to indicate the error.

##### Errors

On error, compress2() shall return a value as described below:

Z\_BUF\_ERROR

  The buffer *dest* was not large enough to hold the compressed data.

Z\_MEM\_ERROR

  Insufficient memory.

Z\_STREAM\_ERROR

  The *level* was not Z\_DEFAULT\_COMPRESSION, or was not between 0 and 9.

#### compressBound

##### Name

compressBound — compute compressed data size

##### Synopsis

#include <zlib.h>

int compressBound(uLong *sourceLen*);

##### Description

The compressBound() function shall estimate the size of buffer required to compress *sourceLen* bytes of data using the compress() or compress2() functions. If successful, the value returned shall be an upper bound for the size of buffer required to compress *sourceLen* bytes of data, using the parameters stored in *stream*, in a single call to compress() or compress2().

##### Return Value

The compressBound() shall return a value representing the upper bound of an array to allocate to hold the compressed data in a single call to compress() or compress2(). This function may return a conservative value that may be larger than *sourceLen*.

##### Errors

None defined.

#### crc32

##### Name

crc32 — compute CRC-32 Checksum

##### Synopsis

#include <zlib.h>

uLong crc32(uLong *crc*, const Bytef \* *buf*, uInt *len*);

##### Description

The crc32() function shall compute a running Cyclic Redundancy Check checksum, as defined in [ITU-T V.42](#ID_STD_46_ITUV42). On entry, *crc* is the previous value for the checksum, and *buf* shall point to an array of *len* bytes of data to be added to this checksum. The crc32() function shall return the new checksum.

If *buf* is NULL (or Z\_NULL), crc32() shall return the initial checksum.

##### Return Value

The crc32() function shall return the new checksum value.

##### Errors

None defined.

##### Application Usage (informative)

The following code fragment demonstrates typical usage of the crc32() function:

uLong crc = crc32(0L, Z\_NULL, 0);

while (read\_buffer(buffer, length) != EOF) {

crc = crc32(crc, buffer, length);

}

if (crc != original\_crc) error();

#### deflate

##### Name

deflate — compress data

##### Synopsis

#include <zlib.h>

int deflate(z\_streamp *stream*, int *flush*);

##### Description

The deflate() function shall attempt to compress data until either the input buffer is empty or the output buffer is full. The *stream* references a z\_stream structure. Before the first call to deflate(), this structure should have been initialized by a call to deflateInit2\_().

**Note:** deflateInit2\_() is only in the binary standard; source level applications should initialize *stream* via a call to deflateInit() or deflateInit2().

In addition, the *stream* input and output buffers should have been initialized as follows:

*next\_in*

  should point to the data to be compressed.

*avail\_in*

  should contain the number of bytes of data in the buffer referenced by *next\_in*.

*next\_out*

  should point to a buffer where compressed data may be placed.

*avail\_out*

  should contain the size in bytes of the buffer referenced by *next\_out*

The deflate() function shall perform one or both of the following actions:

1. Compress input data from *next\_in* and update *next\_in*, *avail\_in* and *total\_in* to reflect the data that has been compressed.

2. Fill the output buffer referenced by *next\_out*, and update *next\_out*, *avail\_out* and *total\_out* to reflect the compressed data that has been placed there. If *flush* is not Z\_NO\_FLUSH, and *avail\_out* indicates that there is still space in output buffer, this action shall always occur (see below for further details).

The deflate() function shall return when either *avail\_in* reaches zero (indicating that all the input data has been compressed), or *avail\_out* reaches zero (indicating that the output buffer is full).

On success, the deflate() function shall set the *adler* field of the *stream* to the adler32() checksum of all the input data compressed so far (represented by *total\_in*).

If the deflate() function shall attempt to determine the type of input data, and set field *data\_type* in *stream* to Z\_ASCII if the majority of the data bytes fall within the ASCII (ISO 646) printable character range. Otherwise, it shall set *data\_type* to Z\_BINARY. This data type is informational only, and does not affect the compression algorithm.

**Note:** Future versions of the LSB may remove this requirement, since it is based on an outdated character set that does not support Internationalization, and does not affect the algorithm. It is included for information only at this release. Applications should not depend on this field.

##### Flush Operation

The parameter *flush* determines when compressed bits are added to the output buffer in *next\_out*. If *flush* is Z\_NO\_FLUSH, deflate() may return with some data pending output, and not yet added to the output buffer.

If *flush* is Z\_SYNC\_FLUSH, deflate() shall flush all pending output to *next\_out* and align the output to a byte boundary. A synchronization point is generated in the output.

If *flush* is Z\_FULL\_FLUSH, all output shall be flushed, as for Z\_SYNC\_FLUSH, and the compression state shall be reset. A synchronization point is generated in the output.

**Rationale:** Z\_SYNC\_FLUSH is intended to ensure that the compressed data contains all the data compressed so far, and allows a decompressor to reconstruct all of the input data. Z\_FULL\_FLUSH allows decompression to restart from this point if the previous compressed data has been lost or damaged. Flushing is likely to degrade the performance of the compression system, and should only be used where necessary.

If *flush* is set to Z\_FINISH, all pending input shall be processed and deflate() shall return with Z\_STREAM\_END if there is sufficient space in the output buffer at *next\_out*, as indicated by *avail\_out*. If deflate() is called with *flush* set to Z\_FINISH and there is insufficient space to store the compressed data, and no other error has occurred during compression, deflate() shall return Z\_OK, and the application should call deflate() again with *flush* unchanged, and having updated *next\_out* and *avail\_out*.

If all the compression is to be done in a single step, deflate() may be called with *flush* set to Z\_FINISH immediately after the stream has been initialized if *avail\_out* is set to at least the value returned by deflateBound().

##### Return Value

On success, deflate() shall return Z\_OK, unless *flush* was set to Z\_FINISH and there was sufficient space in the output buffer to compress all of the input data. In this case, deflate() shall return Z\_STREAM\_END. On error, deflate() shall return a value to indicate the error.

**Note:** If deflate() returns Z\_OK and has set *avail\_out* to zero, the function should be called again with the same value for *flush*, and with updated *next\_out* and *avail\_out* until deflate() returns with Z\_OK (or Z\_STREAM\_END if *flush* is set to Z\_FINISH) and a non-zero *avail\_out*.

##### Errors

On error, deflate() shall return a value as described below, and set the *msg* field of *stream* to point to a string describing the error:

Z\_BUF\_ERROR

  No progress is possible; either *avail\_in* or *avail\_out* was zero.

Z\_MEM\_ERROR

  Insufficient memory.

Z\_STREAM\_ERROR

  The state (as represented in *stream*) is inconsistent, or *stream* was NULL.

#### deflateBound

##### Name

deflateBound — compute compressed data size

##### Synopsis

#include <zlib.h>

int deflateBound(z\_streamp *stream*, uLong *sourceLen*);

##### Description

The deflateBound() function shall estimate the size of buffer required to compress *sourceLen* bytes of data. If successful, the value returned shall be an upper bound for the size of buffer required to compress *sourceLen* bytes of data, using the parameters stored in *stream*, in a single call to deflate() with flush set to Z\_FINISH.

On entry, *stream* should have been initialized via a call to deflateInit\_() or deflateInit2\_().

##### Return Value

The deflateBound() shall return a value representing the upper bound of an array to allocate to hold the compressed data in a single call to deflate(). If the *stream* is not correctly initialized, or is NULL, then deflateBound() may return a conservative value that may be larger than *sourceLen*.

##### Errors

None defined.

#### deflateCopy

##### Name

deflateCopy — copy compression stream

##### Synopsis

#include <zlib.h>

int deflateCopy(z\_streamp *dest*, z\_streamp *source*);

##### Description

The deflateCopy() function shall copy the compression state information in *source* to the uninitialized z\_stream structure referenced by *dest*.

On successful return, *dest* will be an exact copy of the stream referenced by *source*. The input and output buffer pointers in *next\_in* and *next\_out* will reference the same data.

##### Return Value

On success, deflateCopy() shall return Z\_OK. Otherwise it shall return a value less than zero to indicate the error.

##### Errors

On error, deflateCopy() shall return a value as described below:

Z\_STREAM\_ERROR

  The state in *source* is inconsistent, or either *source* or *dest* was NULL.

Z\_MEM\_ERROR

  Insufficient memory available.

##### Application Usage (informative)

This function can be useful when several compression strategies will be tried, for example when there are several ways of pre-processing the input data with a filter. The streams that will be discarded should then be freed by calling deflateEnd(). Note that deflateCopy() duplicates the internal compression state which can be quite large, so this strategy may be slow and can consume lots of memory.

#### deflateEnd

##### Name

deflateEnd — free compression stream state

##### Synopsis

#include <zlib.h>

int deflateEnd(z\_streamp *stream*);

##### Description

The deflateEnd() function shall free all allocated state information referenced by *stream*. All pending output is discarded, and unprocessed input is ignored.

##### Return Value

On success, deflateEnd() shall return Z\_OK, or Z\_DATA\_ERROR if there was pending output discarded or input unprocessed. Otherwise it shall return Z\_STREAM\_ERROR to indicate the error.

##### Errors

On error, deflateEnd() shall return Z\_STREAM\_ERROR. The following conditions shall be treated as an error:

• The state in *stream* is inconsistent or inappropriate.

• *stream* is NULL.

#### deflateInit2\_

##### Name

deflateInit2\_ — initialize compression system

##### Synopsis

#include <zlib.h>

int deflateInit2\_ (z\_streamp *strm*, int *level*, int *method*, int *windowBits*, int *memLevel*, int *strategy*, char \* *version*, int *stream\_size*);

##### Description

The deflateInit2\_() function shall initialize the compression system. On entry, *strm* shall refer to a user supplied z\_stream object (a z\_stream\_s structure). The following fields shall be set on entry:

*zalloc*

  a pointer to an alloc\_func function, used to allocate state information. If this is NULL, a default allocation function will be used.

*zfree*

  a pointer to a free\_func function, used to free memory allocated by the *zalloc* function. If this is NULL a default free function will be used.

*opaque*

  If *alloc\_func* is not NULL, *opaque* is a user supplied pointer to data that will be passed to the *alloc\_func* and *free\_func* functions.

If the *version* requested is not compatible with the version implemented, or if the size of the z\_stream\_s structure provided in *stream\_size* does not match the size in the library implementation, deflateInit2\_() shall fail, and return Z\_VERSION\_ERROR.

The *level* supplied shall be a value between 0 and 9, or the value Z\_DEFAULT\_COMPRESSION. A *level* of 1 requests the highest speed, while a *level* of 9 requests the highest compression. A *level* of 0 indicates that no compression should be used, and the output shall be the same as the input.

The *method* selects the compression algorithm to use. LSB conforming implementation shall support the Z\_DEFLATED method, and may support other implementation defined methods.

The *windowBits* parameter shall be a base 2 logarithm of the window size to use, and shall be a value between 8 and 15. A smaller value will use less memory, but will result in a poorer compression ratio, while a higher value will give better compression but utilize more memory.

The *memLevel* parameter specifies how much memory to use for the internal state. The value of *memLevel* shall be between 1 and MAX\_MEM\_LEVEL. Smaller values use less memory but are slower, while higher values use more memory to gain compression speed.

The *strategy* parameter selects the compression strategy to use:

Z\_DEFAULT\_STRATEGY

  use the system default compression strategy. Z\_DEFAULT\_STRATEGY is particularly appropriate for text data.

Z\_FILTERED

  use a compression strategy tuned for data consisting largely of small values with a fairly random distribution. Z\_FILTERED uses more Huffman encoding and less string matching than Z\_DEFAULT\_STRATEGY.

Z\_HUFFMAN\_ONLY

  force Huffman encoding only, with no string match.

The deflateInit2\_() function is not in the source standard; it is only in the binary standard. Source applications should use the deflateInit2() macro.

##### Return Value

On success, the deflateInit2\_() function shall return Z\_OK. Otherwise, deflateInit2\_() shall return a value as described below to indicate the error.

##### Errors

On error, deflateInit2\_() shall return one of the following error indicators:

Z\_STREAM\_ERROR

  Invalid parameter.

Z\_MEM\_ERROR

  Insufficient memory available.

Z\_VERSION\_ERROR

  The version requested is not compatible with the library version, or the z\_stream size differs from that used by the library.

In addition, the *msg* field of the *strm* may be set to an error message.

#### deflateInit\_

##### Name

deflateInit\_ — initialize compression system

##### Synopsis

#include <zlib.h>

int deflateInit\_(z\_streamp *stream*, int *level*, const char \* *version*, int *stream\_size*);

##### Description

The deflateInit\_() function shall initialize the compression system. On entry, *stream* shall refer to a user supplied z\_stream object (a z\_stream\_s structure). The following fields shall be set on entry:

*zalloc*

  a pointer to an alloc\_func function, used to allocate state information. If this is NULL, a default allocation function will be used.

*zfree*

  a pointer to a free\_func function, used to free memory allocated by the *zalloc* function. If this is NULL a default free function will be used.

*opaque*

  If *alloc\_func* is not NULL, *opaque* is a user supplied pointer to data that will be passed to the *alloc\_func* and *free\_func* functions.

If the *version* requested is not compatible with the version implemented, or if the size of the z\_stream\_s structure provided in *stream\_size* does not match the size in the library implementation, deflateInit\_() shall fail, and return Z\_VERSION\_ERROR.

The *level* supplied shall be a value between 0 and 9, or the value Z\_DEFAULT\_COMPRESSION. A *level* of 1 requests the highest speed, while a *level* of 9 requests the highest compression. A *level* of 0 indicates that no compression should be used, and the output shall be the same as the input.

The deflateInit\_() function is not in the source standard; it is only in the binary standard. Source applications should use the deflateInit() macro.

The deflateInit\_() function is equivalent to

deflateInit2\_(stream, level, Z\_DEFLATED, MAX\_WBITS, MAX\_MEM\_LEVEL,

Z\_DEFAULT\_STRATEGY, version, stream\_size);

##### Return Value

On success, the deflateInit\_() function shall return Z\_OK. Otherwise, deflateInit\_() shall return a value as described below to indicate the error.

##### Errors

On error, deflateInit\_() shall return one of the following error indicators:

Z\_STREAM\_ERROR

  Invalid parameter.

Z\_MEM\_ERROR

  Insufficient memory available.

Z\_VERSION\_ERROR

  The version requested is not compatible with the library version, or the z\_stream size differs from that used by the library.

In addition, the *msg* field of the *stream* may be set to an error message.

#### deflateParams

##### Name

deflateParams — set compression parameters

##### Synopsis

#include <zlib.h>

int deflateParams(z\_streamp *stream*, int *level*, int *strategy*);

##### Description

The deflateParams() function shall dynamically alter the compression parameters for the compression stream object *stream*. On entry, *stream* shall refer to a user supplied z\_stream object (a z\_stream\_s structure), already initialized via a call to deflateInit\_() or deflateInit2\_().

The *level* supplied shall be a value between 0 and 9, or the value Z\_DEFAULT\_COMPRESSION. A *level* of 1 requests the highest speed, while a *level* of 9 requests the highest compression. A *level* of 0 indicates that no compression should be used, and the output shall be the same as the input. If the compression level is altered by deflateParams(), and some data has already been compressed with this *stream* (i.e. *total\_in* is not zero), and the new *level* requires a different underlying compression method, then *stream* shall be flushed by a call to deflate().

The *strategy* parameter selects the compression strategy to use:

Z\_DEFAULT\_STRATEGY

  use the system default compression strategy. Z\_DEFAULT\_STRATEGY is particularly appropriate for text data.

Z\_FILTERED

  use a compression strategy tuned for data consisting largely of small values with a fairly random distribution. Z\_FILTERED uses more Huffman encoding and less string matching than Z\_DEFAULT\_STRATEGY.

Z\_HUFFMAN\_ONLY

  force Huffman encoding only, with no string match.

##### Return Value

On success, the deflateParams() function shall return Z\_OK. Otherwise, deflateParams() shall return a value as described below to indicate the error.

##### Errors

On error, deflateParams() shall return one of the following error indicators:

Z\_STREAM\_ERROR

  Invalid parameter.

Z\_MEM\_ERROR

  Insufficient memory available.

Z\_BUF\_ERROR

  Insufficient space in *stream* to flush the current output.

In addition, the *msg* field of the *strm* may be set to an error message.

##### Application Usage (Informative)

Applications should ensure that the *stream* is flushed, e.g. by a call to **deflate(stream, Z\_SYNC\_FLUSH)** before calling deflateParams(), or ensure that there is sufficient space in *next\_out* (as identified by *avail\_out*) to ensure that all pending output and all uncompressed input can be flushed in a single call to deflate().

**Rationale:** Although the deflateParams() function should flush pending output and compress all pending input, the result is unspecified if there is insufficient space in the output buffer. Applications should only call deflateParams() when the *stream* is effectively empty (flushed).

The deflateParams() can be used to switch between compression and straight copy of the input data, or to switch to a different kind of input data requiring a different strategy.

#### deflateReset

##### Name

deflateReset — reset compression stream state

##### Synopsis

#include <zlib.h>

int deflateReset(z\_streamp *stream*);

##### Description

The deflateReset() function shall reset all state associated with *stream*. All pending output shall be discarded, and the counts of processed bytes (*total\_in* and *total\_out*) shall be reset to zero.

##### Return Value

On success, deflateReset() shall return Z\_OK. Otherwise it shall return Z\_STREAM\_ERROR to indicate the error.

##### Errors

On error, deflateReset() shall return Z\_STREAM\_ERROR. The following conditions shall be treated as an error:

• The state in *stream* is inconsistent or inappropriate.

• *stream* is NULL.

#### deflateSetDictionary

##### Name

deflateSetDictionary — initialize compression dictionary

##### Synopsis

#include <zlib.h>

int deflateSetDictionary(z\_streamp *stream*, const Bytef \* *dictionary*, uInt *dictlen*);

##### Description

The deflateSetDictionary() function shall initialize the compression dictionary associated with *stream* using the *dictlen* bytes referenced by *dictionary*.

The implementation may silently use a subset of the provided dictionary if the dictionary cannot fit in the current window associated with *stream* (see deflateInit2\_()). The application should ensure that the dictionary is sorted such that the most commonly used strings occur at the end of the dictionary.

If the dictionary is successfully set, the Adler32 checksum of the entire provided dictionary shall be stored in the *adler* member of *stream*. This value may be used by the decompression system to select the correct dictionary. The compression and decompression systems must use the same dictionary.

*stream* shall reference an initialized compression stream, with *total\_in* zero (i.e. no data has been compressed since the stream was initialized).

##### Return Value

On success, deflateSetDictionary() shall return Z\_OK. Otherwise it shall return Z\_STREAM\_ERROR to indicate an error.

##### Errors

On error, deflateSetDictionary() shall return a value as described below:

Z\_STREAM\_ERROR

  The state in *stream* is inconsistent, or *stream* was NULL.

##### Application Usage (informative)

The application should provide a dictionary consisting of strings {{{ed note: do we really mean "strings"? Null terminated?}}} that are likely to be encountered in the data to be compressed. The application should ensure that the dictionary is sorted such that the most commonly used strings occur at the end of the dictionary.

The use of a dictionary is optional; however if the data to be compressed is relatively short and has a predictable structure, the use of a dictionary can substantially improve the compression ratio.

#### get\_crc\_table

##### Name

get\_crc\_table — generate a table for crc calculations

##### Synopsis

#include <zlib.h>

const uLongf \* get\_crc\_table(void);

##### Description

Generate tables for a byte-wise 32-bit CRC calculation based on the polynomial: x32+x26+x23+x22+x16+x12+x11+x10+x8+x7+x5+x4+x2+x+1

In a multi-threaded application, get\_crc\_table() should be called by one thread to initialize the tables before any other thread calls any libz function.

##### Return Value

The get\_crc\_table() function shall return a pointer to the first of a set of tables used internally to calculate CRC-32 values (see crc32()).

##### Errors

None defined.

#### gzclose

##### Name

gzclose — close a compressed file stream

##### Synopsis

#include <zlib.h>

int gzclose (gzFile *file* );

##### Description

The gzclose() function shall close the compressed file stream *file*. If *file* was open for writing, gzclose() shall first flush any pending output. Any state information allocated shall be freed.

##### Return Value

On success, gzclose() shall return Z\_OK. Otherwise, gzclose() shall return an error value as described below.

##### Errors

On error, gzclose() may set the global variable errno to indicate the error. The gzclose() shall return a value other than Z\_OK on error.

Z\_STREAM\_ERROR

*file* was NULL (or Z\_NULL), or did not refer to an open compressed file stream.

Z\_ERRNO

  An error occurred in the underlying base libraries, and the application should check errno for further information.

Z\_BUF\_ERROR

  no compression progress is possible during buffer flush (see deflate()).

#### gzdopen

##### Name

gzdopen — open a compressed file

##### Synopsis

#include <zlib.h>

gzFile gzdopen ( int *fd*, const char \**mode* );

##### Description

The gzdopen() function shall attempt to associate the open file referenced by *fd* with a gzFile object. The *mode* argument is based on that of fopen(), but the *mode* parameter may also contain the following characters:

*digit*

  set the compression level to *digit*. A low value (e.g. 1) means high speed, while a high value (e.g. 9) means high compression. A compression level of 0 (zero) means no compression. See deflateInit2\_() for further details.

*[fhR]*

  set the compression strategy to *[fhR]*. The letter f corresponds to filtered data, the letter h corresponds to Huffman only compression, and the letter R corresponds to Run Length Encoding. See deflateInit2\_() for further details.

If *fd* refers to an uncompressed file, and *mode* refers to a read mode, gzdopen() shall attempt to open the file and return a gzFile object suitable for reading directly from the file without any decompression.

If *mode* is NULL, or if *mode* does not contain one of r, w, or a, gzdopen() shall return Z\_NULL, and need not set any other error condition.

##### Example

gzdopen(fileno(stdin), "r");

Attempt to associate the standard input with a gzFile object.

##### Return Value

On success, gzdopen() shall return a gzFile object. On failure, gzdopen() shall return Z\_NULL and may set errno accordingly.

**Note:** At version 1.2.2, zlib does not set errno for several error conditions. Applications may not be able to determine the cause of an error.

##### Errors

On error, gzdopen() may set the global variable errno to indicate the error.

#### gzeof

##### Name

gzeof — check for end-of-file on a compressed file stream

##### Synopsis

#include <zlib.h>

int gzeof (gzFile *file* );

##### Description

The gzeof() function shall test the compressed file stream *file* for end of file.

##### Return Value

If *file* was open for reading and end of file has been reached, gzeof() shall return 1. Otherwise, gzeof() shall return 0.

##### Errors

None defined.

#### gzerror

##### Name

gzerror — decode an error on a compressed file stream

##### Synopsis

#include <zlib.h>

const char \* gzerror (gzFile *file*, int \* *errnum*);

##### Description

The gzerror() function shall return a string describing the last error to have occurred associated with the open compressed file stream referred to by *file*. It shall also set the location referenced by *errnum* to an integer value that further identifies the error.

##### Return Value

The gzerror() function shall return a string that describes the last error associated with the given *file* compressed file stream. This string shall have the format "%s: %s", with the name of the file, followed by a colon, a space, and the description of the error. If the compressed file stream was opened by a call to gzdopen(), the format of the filename is unspecified.

**Rationale:** Although in all current implementations of libz file descriptors are named "<fd:%d>", the code suggests that this is for debugging purposes only, and may change in a future release.

It is unspecified if the string returned is determined by the setting of the LC\_MESSAGES category in the current locale.

##### Errors

None defined.

#### gzflush

##### Name

gzflush — flush a compressed file stream

##### Synopsis

#include <zlib.h>

int gzflush(gzFile *file*, int *flush*);

##### Description

The gzflush() function shall flush pending output to the compressed file stream identified by *file*, which must be open for writing.

##### Flush Operation

The parameter *flush* determines which compressed bits are added to the output file. If *flush* is Z\_NO\_FLUSH, gzflush() may return with some data pending output, and not yet written to the file.

If *flush* is Z\_SYNC\_FLUSH, gzflush() shall flush all pending output to *file* and align the output to a byte boundary. There may still be data pending compression that is not flushed.

If *flush* is Z\_FULL\_FLUSH, all output shall be flushed, as for Z\_SYNC\_FLUSH, and the compression state shall be reset. There may still be data pending compression that is not flushed.

**Rationale:** Z\_SYNC\_FLUSH is intended to ensure that the compressed data contains all the data compressed so far, and allows a decompressor to reconstruct all of the input data. Z\_FULL\_FLUSH allows decompression to restart from this point if the previous compressed data has been lost or damaged. Flushing is likely to degrade the performance of the compression system, and should only be used where necessary.

If *flush* is set to Z\_FINISH, all pending uncompressed data shall be compressed and all output shall be flushed.

##### Return Value

On success, gzflush() shall return the value Z\_OK. Otherwise gzflush() shall return a value to indicate the error, and may set the error number associated with the compressed file stream *file*.

**Note:** If *flush* is set to Z\_FINISH and the flush operation is successful, gzflush() will return Z\_OK, but the compressed file stream error value may be set to Z\_STREAM\_END.

##### Errors

On error, gzflush() shall return an error value, and may set the error number associated with the stream identified by *file* to indicate the error. Applications may use gzerror() to access this error value.

Z\_ERRNO

  An underlying base library function has indicated an error. The global variable errno may be examined for further information.

Z\_STREAM\_ERROR

  The stream is invalid, is not open for writing, or is in an invalid state.

Z\_BUF\_ERROR

  no compression progress is possible (see deflate()).

Z\_MEM\_ERROR

  Insufficient memory available to compress.

#### gzgetc

##### Name

gzgetc — read a character from a compressed file

##### Synopsis

#include <zlib.h>

int gzgetc (gzFile *file*);

##### Description

The gzgetc() function shall read the next single character from the compressed file stream referenced by *file*, which shall have been opened in a read mode (see gzopen() and gzdopen()).

##### Return Value

On success, gzgetc() shall return the uncompressed character read, otherwise, on end of file or error, gzgetc() shall return -1.

##### Errors

On end of file or error, gzgetc() shall return -1. Further information can be found by calling gzerror() with a pointer to the compressed file stream.

#### gzgets

##### Name

gzgets — read a string from a compressed file

##### Synopsis

#include <zlib.h>

char \* gzgets (gzFile *file*, char \* *buf*, int *len*);

##### Description

The gzgets() function shall attempt to read data from the compressed file stream *file*, uncompressing it into *buf* until either *len*-1 bytes have been inserted into *buf*, or until a newline character has been uncompressed into *buf*. A null byte shall be appended to the uncompressed data. The *file* shall have been opened in for reading (see gzopen() and gzdopen()).

##### Return Value

On success, gzgets() shall return a pointer to *buf*. Otherwise, gzgets() shall return Z\_NULL. Applications may examine the cause using gzerror().

##### Errors

On error, gzgets() shall return Z\_NULL. The following conditions shall always be treated as an error:

|  |
| --- |
| *file* is NULL, or does not refer to a file open for reading; |
| *buf* is NULL; |
| *len* is less than or equal to zero. |

#### gzopen

##### Name

gzopen — open a compressed file

##### Synopsis

#include <zlib.h>

gzFile gzopen (const char \**path* , const char \**mode* );

##### Description

The gzopen() function shall open the compressed file named by *path*. The *mode* argument is based on that of fopen(), but the *mode* parameter may also contain the following characters:

*digit*

  set the compression level to *digit*. A low value (e.g. 1) means high speed, while a high value (e.g. 9) means high compression. A compression level of 0 (zero) means no compression. See deflateInit2\_() for further details.

*[fhR]*

  set the compression strategy to *[fhR]*. The letter f corresponds to filtered data, the letter h corresponds to Huffman only compression, and the letter R corresponds to Run Length Encoding. See deflateInit2\_() for further details.

If *path* refers to an uncompressed file, and *mode* refers to a read mode, gzopen() shall attempt to open the file and return a gzFile object suitable for reading directly from the file without any decompression.

If *path* or *mode* is NULL, or if *mode* does not contain one of r, w, or a, gzopen() shall return Z\_NULL, and need not set any other error condition.

The gzFile object is also referred to as a compressed file stream.

##### Example

gzopen("file.gz", "w6h");

Attempt to create a new compressed file, file.gz, at compression level 6 using Huffman only compression.

##### Return Value

On success, gzopen() shall return a gzFile object (also known as a *compressed file stream*). On failure, gzopen() shall return Z\_NULL and may set errno accordingly.

**Note:** At version 1.2.2, zlib does not set errno for several error conditions. Applications may not be able to determine the cause of an error.

##### Errors

On error, gzopen() may set the global variable errno to indicate the error.

#### gzprintf

##### Name

gzprintf — format data and compress

##### Synopsis

#include <zlib.h>

int gzprintf (gzFile *file*, const char \* *fmt*, ...);

##### Description

The gzprintf() function shall format data as for fprintf(), and write the resulting string to the compressed file stream *file*.

##### Return Value

The gzprintf() function shall return the number of uncompressed bytes actually written, or a value less than or equal to 0 in the event of an error.

##### Errors

If *file* is NULL, or refers to a compressed file stream that has not been opened for writing, gzprintf() shall return Z\_STREAM\_ERROR. Otherwise, errors are as for gzwrite().

#### gzputc

##### Name

gzputc — write character to a compressed file

##### Synopsis

#include <zlib.h>

int gzputc (gzFile *file*, int *c*);

##### Description

The gzputc() function shall write the single character *c*, converted from integer to unsigned character, to the compressed file referenced by *file*, which shall have been opened in a write mode (see gzopen() and gzdopen()).

##### Return Value

On success, gzputc() shall return the value written, otherwise gzputc() shall return -1.

##### Errors

On error, gzputc() shall return -1.

#### gzputs

##### Name

gzputs — string write to a compressed file

##### Synopsis

#include <zlib.h>

int gzputs (gzFile *file*, const char \* *s*);

##### Description

The gzputs() function shall write the null terminated string *s* to the compressed file referenced by *file*, which shall have been opened in a write mode (see gzopen() and gzdopen()). The terminating null character shall not be written. The gzputs() function shall return the number of uncompressed bytes actually written.

##### Return Value

On success, gzputs() shall return the number of uncompressed bytes actually written to *file*. On error gzputs() shall return a value less than or equal to 0. Applications may examine the cause using gzerror().

##### Errors

On error, gzputs() shall set the error number associated with the stream identified by *file* to indicate the error. Applications should use gzerror() to access this error value. If *file* is NULL, gzputs() shall return Z\_STREAM\_ERR.

Z\_ERRNO

  An underlying base library function has indicated an error. The global variable errno may be examined for further information.

Z\_STREAM\_ERROR

  The stream is invalid, is not open for writing, or is in an invalid state.

Z\_BUF\_ERROR

  no compression progress is possible (see deflate()).

Z\_MEM\_ERROR

  Insufficient memory available to compress.

#### gzread

##### Name

gzread — read from a compressed file

##### Synopsis

#include <zlib.h>

int gzread (gzFile *file*, voidp *buf*, unsigned int *len*);

##### Description

The gzread() function shall read data from the compressed file referenced by *file*, which shall have been opened in a read mode (see gzopen() and gzdopen()). The gzread() function shall read data from *file*, and uncompress it into *buf*. At most, *len* bytes of uncompressed data shall be copied to *buf*. If the file is not compressed, gzread() shall simply copy data from *file* to *buf* without alteration.

##### Return Value

On success, gzread() shall return the number of bytes decompressed into *buf*. If gzread() returns 0, either the end-of-file has been reached or an underlying read error has occurred. Applications should use gzerror() or gzeof() to determine which occurred. On other errors, gzread() shall return a value less than 0 and applications may examine the cause using gzerror().

##### Errors

On error, gzread() shall set the error number associated with the stream identified by *file* to indicate the error. Applications should use gzerror() to access this error value.

Z\_ERRNO

  An underlying base library function has indicated an error. The global variable errno may be examined for further information.

Z\_STREAM\_END

  End of file has been reached on input.

Z\_DATA\_ERROR

  A CRC error occurred when reading data; the file is corrupt.

Z\_STREAM\_ERROR

  The stream is invalid, or is in an invalid state.

Z\_NEED\_DICT

  A dictionary is needed (see inflateSetDictionary()).

Z\_MEM\_ERROR

  Insufficient memory available to decompress.

#### gzrewind

##### Name

gzrewind — reset the file-position indicator on a compressed file stream

##### Synopsis

#include <zlib.h>

int gzrewind(gzFile *file*);

##### Description

The gzrewind() function shall set the starting position for the next read on compressed file stream *file* to the beginning of file. *file* must be open for reading.

gzrewind() is equivalent to

(int)gzseek(file, 0L, SEEK\_SET)

.

##### Return Value

On success, gzrewind() shall return 0. On error, gzrewind() shall return -1, and may set the error value for *file* accordingly.

##### Errors

On error, gzrewind() shall return -1, indicating that *file* is NULL, or does not represent an open compressed file stream, or represents a compressed file stream that is open for writing and is not currently at the beginning of file.

#### gzseek

##### Name

gzseek — reposition a file-position indicator in a compressed file stream

##### Synopsis

#include <zlib.h>

z\_off\_t gzseek(gzFile *file*, z\_off\_t *offset*, int *whence*);

##### Description

The gzseek() function shall set the file-position indicator for the compressed file stream *file*. The file-position indicator controls where the next read or write operation on the compressed file stream shall take place. The *offset* indicates a byte offset in the uncompressed data. The *whence* parameter may be one of:

SEEK\_SET

  the offset is relative to the start of the uncompressed data.

SEEK\_CUR

  the offset is relative to the current positition in the uncompressed data.

**Note:** The value SEEK\_END need not be supported.

If the *file* is open for writing, the new offset must be greater than or equal to the current offset. In this case, gzseek() shall compress a sequence of null bytes to fill the gap from the previous offset to the new offset.

##### Return Value

On success, gzseek() shall return the resulting offset in the file expressed as a byte position in the *uncompressed* data stream. On error, gzseek() shall return -1, and may set the error value for *file* accordingly.

##### Errors

On error, gzseek() shall return -1. The following conditions shall always result in an error:

• *file* is NULL

• *file* does not represent an open compressed file stream.

• *file* refers to a compressed file stream that is open for writing, and the newly computed offset is less than the current offset.

• The newly computed offset is less than zero.

• *whence* is not one of the supported values.

##### Application Usage (informative)

If *file* is open for reading, the implementation may still need to uncompress all of the data up to the new offset. As a result, gzseek() may be extremely slow in some circumstances.

#### gzsetparams

##### Name

gzsetparams — dynamically set compression parameters

##### Synopsis

#include <zlib.h>

int gzsetparams (gzFile *file*, int *level*, int *strategy*);

##### Description

The gzsetparams() function shall set the compression level and compression strategy on the compressed file stream referenced by *file*. The compressed file stream shall have been opened in a write mode. The *level* and *strategy* are as defined in [deflateInit2.](#ID_ZLIB_45_DEFLATEINIT2). If there is any data pending writing, it shall be flushed before the parameters are updated.

##### Return Value

On success, the gzsetparams() function shall return Z\_OK.

##### Errors

On error, gzsetparams() shall return one of the following error indications:

Z\_STREAM\_ERROR

  Invalid parameter, or *file* not open for writing.

Z\_BUF\_ERROR

  An internal inconsistency was detected while flushing the previous buffer.

#### gztell

##### Name

gztell — find position on a compressed file stream

##### Synopsis

#include <zlib.h>

z\_off\_t gztell (gzFile *file* );

##### Description

The gztell() function shall return the starting position for the next read or write operation on compressed file stream *file*. This position represents the number of bytes from the beginning of file in the uncompressed data.

gztell() is equivalent to

gzseek(file, 0L, SEEK\_CUR)

.

##### Return Value

gztell() shall return the current offset in the file expressed as a byte position in the *uncompressed* data stream. On error, gztell() shall return -1, and may set the error value for *file* accordingly.

##### Errors

On error, gztell() shall return -1, indicating that *file* is NULL, or does not represent an open compressed file stream.

#### gzwrite

##### Name

gzwrite — write to a compressed file

##### Synopsis

#include <zlib.h>

int gzwrite (gzFile *file*, voidpc *buf*, unsigned int *len*);

##### Description

The gzwrite() function shall write data to the compressed file referenced by *file*, which shall have been opened in a write mode (see gzopen() and gzdopen()). On entry, *buf* shall point to a buffer containing *len* bytes of uncompressed data. The gzwrite() function shall compress this data and write it to *file*. The gzwrite() function shall return the number of uncompressed bytes actually written.

##### Return Value

On success, gzwrite() shall return the number of uncompressed bytes actually written to *file*. On error gzwrite() shall return a value less than or equal to 0. Applications may examine the cause using gzerror().

##### Errors

On error, gzwrite() shall set the error number associated with the stream identified by *file* to indicate the error. Applications should use gzerror() to access this error value.

Z\_ERRNO

  An underlying base library function has indicated an error. The global variable errno may be examined for further information.

Z\_STREAM\_ERROR

  The stream is invalid, is not open for writing, or is in an invalid state.

Z\_BUF\_ERROR

  no compression progress is possible (see deflate()).

Z\_MEM\_ERROR

  Insufficient memory available to compress.

#### inflate

##### Name

inflate — decompress data

##### Synopsis

#include <zlib.h>

int inflate(z\_streamp *stream*, int *flush*);

##### Description

The inflate() function shall attempt to decompress data until either the input buffer is empty or the output buffer is full. The *stream* references a z\_stream structure. Before the first call to inflate(), this structure should have been initialized by a call to inflateInit2\_().

**Note:** inflateInit2\_() is only in the binary standard; source level applications should initialize *stream* via a call to inflateInit() or inflateInit2().

In addition, the *stream* input and output buffers should have been initialized as follows:

*next\_in*

  should point to the data to be decompressed.

*avail\_in*

  should contain the number of bytes of data in the buffer referenced by *next\_in*.

*next\_out*

  should point to a buffer where decompressed data may be placed.

*avail\_out*

  should contain the size in bytes of the buffer referenced by *next\_out*

The inflate() function shall perform one or both of the following actions:

1. Decompress input data from *next\_in* and update *next\_in*, *avail\_in* and *total\_in* to reflect the data that has been decompressed.

2. Fill the output buffer referenced by *next\_out*, and update *next\_out*, *avail\_out*, and *total\_out* to reflect the decompressed data that has been placed there. If *flush* is not Z\_NO\_FLUSH, and *avail\_out* indicates that there is still space in output buffer, this action shall always occur (see below for further details).

The inflate() function shall return when either *avail\_in* reaches zero (indicating that all the input data has been compressed), or *avail\_out* reaches zero (indicating that the output buffer is full).

Flush Operation

The parameter *flush* determines when uncompressed bytes are added to the output buffer in *next\_out*. If *flush* is Z\_NO\_FLUSH, inflate() may return with some data pending output, and not yet added to the output buffer.

If *flush* is Z\_SYNC\_FLUSH, inflate() shall flush all pending output to *next\_out*, and update *next\_out* and *avail\_out* accordingly.

If *flush* is set to Z\_BLOCK, inflate() shall stop adding data to the output buffer if and when the next compressed block boundary is reached (see [RFC 1951: DEFLATE Compressed Data Format Specification](#ID_STD_46_RFC1951)).

If *flush* is set to Z\_FINISH, all of the compressed input shall be decompressed and added to the output. If there is insufficient output space (i.e. the compressed input data uncompresses to more than *avail\_out* bytes), then inflate() shall fail and return Z\_BUF\_ERROR.

##### Return Value

On success, inflate() shall return Z\_OK if decompression progress has been made, or Z\_STREAM\_END if all of the input data has been decompressed and there was sufficient space in the output buffer to store the uncompressed result. On error, inflate() shall return a value to indicate the error.

**Note:** If inflate() returns Z\_OK and has set *avail\_out* to zero, the function should be called again with the same value for *flush*, and with updated *next\_out* and *avail\_out* until inflate() returns with either Z\_OK or Z\_STREAM\_END and a non-zero *avail\_out*.

On success, inflate() shall set the *adler* to the Adler-32 checksum of the output produced so far (i.e. *total\_out* bytes).

##### Errors

On error, inflate() shall return a value as described below, and may set the *msg* field of *stream* to point to a string describing the error:

Z\_BUF\_ERROR

  No progress is possible; either *avail\_in* or *avail\_out* was zero.

Z\_MEM\_ERROR

  Insufficient memory.

Z\_STREAM\_ERROR

  The state (as represented in *stream*) is inconsistent, or *stream* was NULL.

Z\_NEED\_DICT

  A preset dictionary is required. The *adler* field shall be set to the Adler-32 checksum of the dictionary chosen by the compressor.

#### inflateEnd

##### Name

inflateEnd — free decompression stream state

##### Synopsis

#include <zlib.h>

int inflateEnd(z\_streamp *stream*);

##### Description

The inflateEnd() function shall free all allocated state information referenced by *stream*. All pending output is discarded, and unprocessed input is ignored.

##### Return Value

On success, inflateEnd() shall return Z\_OK. Otherwise it shall return Z\_STREAM\_ERROR to indicate the error.

##### Errors

On error, inflateEnd() shall return Z\_STREAM\_ERROR. The following conditions shall be treated as an error:

• The state in *stream* is inconsistent.

• *stream* is NULL.

• The *zfree* function pointer is NULL.

#### inflateInit2\_

##### Name

inflateInit2\_ — initialize decompression system

##### Synopsis

#include <zlib.h>

int inflateInit2\_ (z\_streamp *strm*, int *windowBits*, char \* *version*, int *stream\_size*);

##### Description

The inflateInit2\_() function shall initialize the decompression system. On entry, *strm* shall refer to a user supplied z\_stream object (a z\_stream\_s structure). The following fields shall be set on entry:

*zalloc*

  a pointer to an alloc\_func function, used to allocate state information. If this is NULL, a default allocation function will be used.

*zfree*

  a pointer to a free\_func function, used to free memory allocated by the *zalloc* function. If this is NULL a default free function will be used.

*opaque*

  If *alloc\_func* is not NULL, *opaque* is a user supplied pointer to data that will be passed to the *alloc\_func* and *free\_func* functions.

If the *version* requested is not compatible with the version implemented, or if the size of the z\_stream\_s structure provided in *stream\_size* does not match the size in the library implementation, inflateInit2\_() shall fail, and return Z\_VERSION\_ERROR.

The *windowBits* parameter shall be a base 2 logarithm of the maximum window size to use, and shall be a value between 8 and 15. If the input data was compressed with a larger window size, subsequent attempts to decompress this data will fail with Z\_DATA\_ERROR, rather than try to allocate a larger window.

The inflateInit2\_() function is not in the source standard; it is only in the binary standard. Source applications should use the inflateInit2() macro.

##### Return Value

On success, the inflateInit2\_() function shall return Z\_OK. Otherwise, inflateInit2\_() shall return a value as described below to indicate the error.

##### Errors

On error, inflateInit2\_() shall return one of the following error indicators:

Z\_STREAM\_ERROR

  Invalid parameter.

Z\_MEM\_ERROR

  Insufficient memory available.

Z\_VERSION\_ERROR

  The version requested is not compatible with the library version, or the z\_stream size differs from that used by the library.

In addition, the *msg* field of the *strm* may be set to an error message.

#### inflateInit\_

##### Name

inflateInit\_ — initialize decompression system

##### Synopsis

#include <zlib.h>

int inflateInit\_(z\_streamp *stream*, const char \* *version*, int *stream\_size*);

##### Description

The inflateInit\_() function shall initialize the decompression system. On entry, *stream* shall refer to a user supplied z\_stream object (a z\_stream\_s structure). The following fields shall be set on entry:

*zalloc*

  a pointer to an alloc\_func function, used to allocate state information. If this is NULL, a default allocation function will be used.

*zfree*

  a pointer to a free\_func function, used to free memory allocated by the *zalloc* function. If this is NULL a default free function will be used.

*opaque*

  If *alloc\_func* is not NULL, *opaque* is a user supplied pointer to data that will be passed to the *alloc\_func* and *free\_func* functions.

If the *version* requested is not compatible with the version implemented, or if the size of the z\_stream\_s structure provided in *stream\_size* does not match the size in the library implementation, inflateInit\_() shall fail, and return Z\_VERSION\_ERROR.

The inflateInit\_() function is not in the source standard; it is only in the binary standard. Source applications should use the inflateInit() macro.

The inflateInit\_() shall be equivalent to

inflateInit2\_(strm, MAX\_WBITS, version, stream\_size);

##### Return Value

On success, the inflateInit\_() function shall return Z\_OK. Otherwise, inflateInit\_() shall return a value as described below to indicate the error.

##### Errors

On error, inflateInit\_() shall return one of the following error indicators:

Z\_STREAM\_ERROR

  Invalid parameter.

Z\_MEM\_ERROR

  Insufficient memory available.

Z\_VERSION\_ERROR

  The version requested is not compatible with the library version, or the z\_stream size differs from that used by the library.

In addition, the *msg* field of the *strm* may be set to an error message.

#### inflateReset

##### Name

inflateReset — reset decompression stream state

##### Synopsis

#include <zlib.h>

int inflateReset(z\_streamp *stream*);

##### Description

The inflateReset() function shall reset all state associated with *stream*. All pending output shall be discarded, and the counts of processed bytes (*total\_in* and *total\_out*) shall be reset to zero.

##### Return Value

On success, inflateReset() shall return Z\_OK. Otherwise it shall return Z\_STREAM\_ERROR to indicate the error.

##### Errors

On error, inflateReset() shall return Z\_STREAM\_ERROR. The following conditions shall be treated as an error:

• The state in *stream* is inconsistent or inappropriate.

• *stream* is NULL.

#### inflateSetDictionary

##### Name

inflateSetDictionary — initialize decompression dictionary

##### Synopsis

#include <zlib.h>

int inflateSetDictionary(z\_streamp *stream*, const Bytef \* *dictionary*, uInt *dictlen*);

##### Description

The inflateSetDictionary() function shall initialize the decompression dictionary associated with *stream* using the *dictlen* bytes referenced by *dictionary*.

The inflateSetDictionary() function should be called immediately after a call to inflate() has failed with return value Z\_NEED\_DICT. The *dictionary* must have the same Adler-32 checksum as the dictionary used for the compression (see deflateSetDictionary()).

*stream* shall reference an initialized decompression stream, with *total\_in* zero (i.e. no data has been decompressed since the stream was initialized).

##### Return Value

On success, inflateSetDictionary() shall return Z\_OK. Otherwise it shall return a value as indicated below.

##### Errors

On error, inflateSetDictionary() shall return a value as described below:

Z\_STREAM\_ERROR

  The state in *stream* is inconsistent, or *stream* was NULL.

Z\_DATA\_ERROR

  The Adler-32 checksum of the supplied dictionary does not match that used for the compression.

##### Application Usage (informative)

The application should provide a dictionary consisting of strings {{{ed note: do we really mean "strings"? Null terminated?}}} that are likely to be encountered in the data to be compressed. The application should ensure that the dictionary is sorted such that the most commonly used strings occur at the end of the dictionary.

The use of a dictionary is optional; however if the data to be compressed is relatively short and has a predictable structure, the use of a dictionary can substantially improve the compression ratio.

#### inflateSync

##### Name

inflateSync — advance compression stream to next sync point

##### Synopsis

#include <zlib.h>

int inflateSync(z\_streamp *stream*);

##### Description

The inflateSync() function shall advance through the compressed data in *stream*, skipping any invalid compressed data, until the next full flush point is reached, or all input is exhausted. See the description for deflate() with flush level Z\_FULL\_FLUSH. No output is placed in *next\_out*.

##### Return Value

On success, inflateSync() shall return Z\_OK, and update the *next\_in*, *avail\_in*, and *total\_in* fields of *stream* to reflect the number of bytes of compressed data that have been skipped. Otherwise, inflateSync() shall return a value as described below to indicate the error.

##### Errors

On error, inflateSync() shall return a value as described below:

Z\_STREAM\_ERROR

  The state (as represented in *stream*) is inconsistent, or *stream* was NULL.

Z\_BUF\_ERROR

  There is no data available to skip over.

Z\_DATA\_ERROR

  No sync point was found.

#### inflateSyncPoint

##### Name

inflateSyncPoint — test for synchronization point

##### Synopsis

#include <zlib.h>

int inflateSyncPoint(z\_streamp *stream*);

##### Description

The inflateSyncPoint() function shall return a non-zero value if the compressed data stream referenced by *stream* is at a synchronization point.

##### Return Value

If the compressed data in *stream* is at a synchronization point (see deflate() with a flush level of Z\_SYNC\_FLUSH or Z\_FULL\_FLUSH), inflateSyncPoint() shall return a non-zero value, other than Z\_STREAM\_ERROR. Otherwise, if the *stream* is valid, inflateSyncPoint() shall return 0. If *stream* is invalid, or in an invalid state, inflateSyncPoint() shall return Z\_STREAM\_ERROR to indicate the error.

##### Errors

On error, inflateSyncPoint() shall return a value as described below:

Z\_STREAM\_ERROR

  The state (as represented in *stream*) is inconsistent, or *stream* was NULL.

#### uncompress

##### Name

uncompress — uncompress data

##### Synopsis

#include <zlib.h>

int uncompress(Bytef \* *dest*, uLongf \* *destLen*, const Bytef \* *source*, uLong *sourceLen*);

##### Description

The uncompress() function shall attempt to uncompress *sourceLen* bytes of data in the buffer *source*, placing the result in the buffer *dest*.

On entry, *destLen* should point to a value describing the size of the *dest* buffer. The application should ensure that this value is large enough to hold the entire uncompressed data.

**Note:** The LSB does not describe any mechanism by which a compressor can communicate the size required to the uncompressor.

On successful exit, the variable referenced by *destLen* shall be updated to hold the length of uncompressed data in *dest*.

##### Return Value

On success, uncompress() shall return Z\_OK. Otherwise, uncompress() shall return a value to indicate the error.

##### Errors

On error, uncompress() shall return a value as described below:

Z\_BUF\_ERROR

  The buffer *dest* was not large enough to hold the uncompressed data.

Z\_MEM\_ERROR

  Insufficient memory.

Z\_DATA\_ERROR

  The compressed data (referenced by *source*) was corrupted.

#### zError

##### Name

zError — translate error number to string

##### Synopsis

#include <zlib.h>

const char \* zError(int *err*);

##### Description

The zError() function shall return the string identifying the error associated with *err*. This allows for conversion from error code to string for functions such as compress() and uncompress(), that do not always set the string version of an error.

##### Return Value

The zError() function shall return a the string identifying the error associated with *err*, or NULL if *err* is not a valid error code.

It is unspecified if the string returned is determined by the setting of the LC\_MESSAGES category in the current locale.

##### Errors

None defined.

#### zlibVersion

##### Name

zlibVersion — discover library version at run time

##### Synopsis

#include <zlib.h>

const char \* zlibVersion (void);

##### Description

The zlibVersion() function shall return the string identifying the interface version at the time the library was built.

Applications should compare the value returned from zlibVersion() with the macro constant ZLIB\_VERSION for compatibility.

##### Return Value

The zlibVersion() function shall return a the string identifying the version of the library currently implemented.

##### Errors

None defined.

## **15.5 Interfaces for libncurses**

[Table 15-3](#ID_LIB_45_LIBNCURSES_45_DEF) defines the library name and shared object name for the libncurses library

**Table 15-3 libncurses Definition**

|  |  |
| --- | --- |
| Library: | libncurses |
| SONAME: | libncurses.so.5 |

The parameters or return types of the following interfaces have had the const qualifier added as shown here, as compared to the specification in [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES).

extern const char \*keyname (int);

extern SCREEN \*newterm (const char \*, FILE \*, FILE \*);

extern const char \*unctrl (chtype);

extern int mvprintw (int, int, const char \*, ...);

extern int mvwprintw (WINDOW \*, int, int, const char \*, ...);

extern int printw (const char \*, ...);

extern int vwprintw (WINDOW \*, const char \*, va\_list);

extern int vw\_printw (WINDOW \*, const char \*, va\_list);

extern int wprintw (WINDOW \*, const char \*, ...);

extern int mvscanw (int, int, const char \*, ...);

extern int mvwscanw (WINDOW \*, int, int, const char \*, ...);

extern int scanw (const char \*, ...);

extern int vwscanw (WINDOW \*, const char \*, va\_list);

extern int vw\_scanw (WINDOW \*, const char \*, va\_list);

extern int wscanw (WINDOW \*, const char \*, ...);

The behavior of the interfaces in this library is specified by the following specifications:

|  |
| --- |
| [LSB] [This Specification](#ID_STD_46_LSB) |
| [X-CURSES] [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES) |

### **15.5.1 Curses**

#### 15.5.1.1 Interfaces for Curses

An LSB conforming implementation shall provide the generic functions for Curses specified in [Table 15-4](#ID_TBL_45_LIBNCURSES_45_CURSE_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 15-4 libncurses - Curses Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| addch [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | addchnstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | addchstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | addnstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| addstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | attr\_get [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | attr\_off [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | attr\_on [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| attr\_set [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | attroff [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | attron [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | attrset [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| baudrate [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | beep [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | bkgd [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | bkgdset [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| border [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | box [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | can\_change\_color [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | cbreak [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| chgat [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | clear [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | clearok [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | clrtobot [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| clrtoeol [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | color\_content [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | color\_set [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | copywin [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| curs\_set [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | def\_prog\_mode [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | def\_shell\_mode [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | del\_curterm [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| delay\_output [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | delch [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | deleteln [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | delscreen [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| delwin [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | derwin [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | doupdate [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | dupwin [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| echo [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | echochar [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | endwin [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | erase [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| erasechar [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | filter [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | flash [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | flushinp [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| getbkgd [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | getch [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | getnstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | getstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| getwin [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | halfdelay [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | has\_colors [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | has\_ic [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| has\_il [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | hline [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | idcok [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | idlok [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| immedok [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | inch [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | inchnstr [[LSB]](#ID_REFSTD_46_LIBNCURSES_46_1) | inchstr [[LSB]](#ID_REFSTD_46_LIBNCURSES_46_1) |
| init\_color [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | init\_pair [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | initscr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | innstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| insch [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | insdelln [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | insertln [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | insnstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| insstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | instr [[LSB]](#ID_REFSTD_46_LIBNCURSES_46_1) | intrflush [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | is\_linetouched [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| is\_wintouched [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | isendwin [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | keyname [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | keypad [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| killchar [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | leaveok [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | longname [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | meta [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| move [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvaddch [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvaddchnstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvaddchstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| mvaddnstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvaddstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvchgat [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvcur [[LSB]](#ID_REFSTD_46_LIBNCURSES_46_1) |
| mvdelch [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvderwin [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvgetch [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvgetnstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| mvgetstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvhline [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvinch [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvinchnstr [[LSB]](#ID_REFSTD_46_LIBNCURSES_46_1) |
| mvinchstr [[LSB]](#ID_REFSTD_46_LIBNCURSES_46_1) | mvinnstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvinsch [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvinsnstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| mvinsstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvinstr [[LSB]](#ID_REFSTD_46_LIBNCURSES_46_1) | mvprintw [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvscanw [[LSB]](#ID_REFSTD_46_LIBNCURSES_46_1) |
| mvvline [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvwaddch [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvwaddchnstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvwaddchstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| mvwaddnstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvwaddstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvwchgat [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvwdelch [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| mvwgetch [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvwgetnstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvwgetstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvwhline [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| mvwin [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvwinch [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvwinchnstr [[LSB]](#ID_REFSTD_46_LIBNCURSES_46_1) | mvwinchstr [[LSB]](#ID_REFSTD_46_LIBNCURSES_46_1) |
| mvwinnstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvwinsch [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvwinsnstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvwinsstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| mvwinstr [[LSB]](#ID_REFSTD_46_LIBNCURSES_46_1) | mvwprintw [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | mvwscanw [[LSB]](#ID_REFSTD_46_LIBNCURSES_46_1) | mvwvline [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| napms [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | newpad [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | newterm [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | newwin [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| nl [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | nocbreak [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | nodelay [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | noecho [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| nonl [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | noqiflush [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | noraw [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | notimeout [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| overlay [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | overwrite [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | pair\_content [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | pechochar [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| pnoutrefresh [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | prefresh [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | printw [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | putp [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| putwin [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | qiflush [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | raw [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | redrawwin [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| refresh [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | reset\_prog\_mode [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | reset\_shell\_mode [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | resetty [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| restartterm [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | ripoffline [[LSB]](#ID_REFSTD_46_LIBNCURSES_46_1) | savetty [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | scanw [[LSB]](#ID_REFSTD_46_LIBNCURSES_46_1) |
| scr\_dump [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | scr\_init [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | scr\_restore [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | scr\_set [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| scrl [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | scroll [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | scrollok [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | set\_curterm [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| set\_term [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | setscrreg [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | setupterm [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | slk\_attr\_set [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| slk\_attroff [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | slk\_attron [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | slk\_attrset [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | slk\_clear [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| slk\_color [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | slk\_init [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | slk\_label [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | slk\_noutrefresh [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| slk\_refresh [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | slk\_restore [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | slk\_set [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | slk\_touch [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| standend [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | standout [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | start\_color [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | subpad [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| subwin [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | syncok [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | termattrs [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | termname [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| tgetent [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | tgetflag [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | tgetnum [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | tgetstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| tgoto [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | tigetflag [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | tigetnum [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | tigetstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| timeout [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | touchline [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | touchwin [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | tparm [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| tputs [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | typeahead [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | unctrl [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | ungetch [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| untouchwin [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | use\_env [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | vidattr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | vidputs [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| vline [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | vw\_printw [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | vw\_scanw [[LSB]](#ID_REFSTD_46_LIBNCURSES_46_1) | vwprintw [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| vwscanw [[LSB]](#ID_REFSTD_46_LIBNCURSES_46_1) | waddch [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | waddchnstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | waddchstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| waddnstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | waddstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | wattr\_get [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | wattr\_off [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| wattr\_on [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | wattr\_set [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | wattroff [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | wattron [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| wattrset [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | wbkgd [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | wbkgdset [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | wborder [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| wchgat [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | wclear [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | wclrtobot [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | wclrtoeol [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| wcolor\_set [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | wcursyncup [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | wdelch [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | wdeleteln [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| wechochar [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | werase [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | wgetch [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | wgetnstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| wgetstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | whline [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | winch [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | winchnstr [[LSB]](#ID_REFSTD_46_LIBNCURSES_46_1) |
| winchstr [[LSB]](#ID_REFSTD_46_LIBNCURSES_46_1) | winnstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | winsch [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | winsdelln [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| winsertln [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | winsnstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | winsstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | winstr [[LSB]](#ID_REFSTD_46_LIBNCURSES_46_1) |
| wmove [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | wnoutrefresh [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | wprintw [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | wredrawln [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| wrefresh [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | wscanw [[LSB]](#ID_REFSTD_46_LIBNCURSES_46_1) | wscrl [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | wsetscrreg [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| wstandend [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | wstandout [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | wsyncdown [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | wsyncup [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| wtimeout [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | wtouchln [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | wvline [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |  |

An LSB conforming implementation shall provide the generic deprecated functions for Curses specified in [Table 15-5](#ID_TBL_45_LIBNCURSES_45_CURSE_45_DEPINTS), with the full mandatory functionality as described in the referenced underlying specification.

**Note:** These interfaces are deprecated, and applications should avoid using them. These interfaces may be withdrawn in future releases of this specification.

**Table 15-5 libncurses - Curses Deprecated Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| tgetent [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | tgetflag [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | tgetnum [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | tgetstr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| tgoto [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |  |  |  |

An LSB conforming implementation shall provide the generic data interfaces for Curses specified in [Table 15-6](#ID_TBL_45_LIBNCURSES_45_CURSE_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 15-6 libncurses - Curses Data Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| COLORS [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | COLOR\_PAIRS [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | COLS [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | LINES [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |
| acs\_map [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | cur\_term [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | curscr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | newscr [[LSB]](#ID_REFSTD_46_LIBNCURSES_46_1) |
| stdscr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) | ttytype [[X-CURSES]](#ID_REFSTD_46_LIBNCURSES_46_2) |  |  |

## **15.6 Data Definitions for libncurses**

This section defines global identifiers and their values that are associated with interfaces contained in libncurses. These definitions are organized into groups that correspond to system headers. This convention is used as a convenience for the reader, and does not imply the existence of these headers, or their content. Where an interface is defined as requiring a particular system header file all of the data definitions for that system header file presented here shall be in effect.

This section gives data definitions to promote binary application portability, not to repeat source interface definitions available elsewhere. System providers and application developers should use this ABI to supplement - not to replace - source interface definition specifications.

This specification uses the [ISO C (1999)](#ID_STD_46_ISOC99) C Language as the reference programming language, and data definitions are specified in ISO C format. The C language is used here as a convenient notation. Using a C language description of these data objects does not preclude their use by other programming languages.

### **15.6.1 curses.h**

#define getattrs(win) ((win)?(win)->\_attrs:A\_NORMAL)

#define ERR (-1)

#define OK (0)

#define ACS\_RARROW (acs\_map['+'])

#define ACS\_LARROW (acs\_map[','])

#define ACS\_UARROW (acs\_map['-'])

#define ACS\_DARROW (acs\_map['.'])

#define ACS\_BLOCK (acs\_map['0'])

#define ACS\_CKBOARD (acs\_map['a'])

#define ACS\_DEGREE (acs\_map['f'])

#define ACS\_PLMINUS (acs\_map['g'])

#define ACS\_BOARD (acs\_map['h'])

#define ACS\_LANTERN (acs\_map['i'])

#define ACS\_LRCORNER (acs\_map['j'])

#define ACS\_URCORNER (acs\_map['k'])

#define ACS\_ULCORNER (acs\_map['l'])

#define ACS\_LLCORNER (acs\_map['m'])

#define ACS\_PLUS (acs\_map['n'])

#define ACS\_S1 (acs\_map['o'])

#define ACS\_HLINE (acs\_map['q'])

#define ACS\_S9 (acs\_map['s'])

#define ACS\_LTEE (acs\_map['t'])

#define ACS\_RTEE (acs\_map['u'])

#define ACS\_BTEE (acs\_map['v'])

#define ACS\_TTEE (acs\_map['w'])

#define ACS\_VLINE (acs\_map['x'])

#define ACS\_DIAMOND (acs\_map['`'])

#define ACS\_BULLET (acs\_map['~'])

#define setsyx(y,x) do{if((y)==-1&&(x)==-1)newscr->\_leaveok=TRUE;else{newscr->\_leaveok=FALSE;wmove(newscr,(y),(x));}}while(0)

#define getsyx(y,x) do{if(newscr->\_leaveok)(y)=(x)=-1;elsegetyx(newscr,(y),(x));}while(0)

#define vid\_attr(a,pair,opts) vidattr(a)

#define getmaxyx(win,y,x) \

(y=(win)?((win)->\_maxy+1):ERR,x=(win)?((win)->\_maxx+1):ERR)

#define getbegyx(win,y,x) \

(y=(win)?(win)->\_begy:ERR,x=(win)?(win)->\_begx:ERR)

#define getyx(win,y,x) \

(y=(win)?(win)->\_cury:ERR,x=(win)?(win)->\_curx:ERR)

#define getparyx(win,y,x) \

(y=(win)?(win)->\_pary:ERR,x=(win)?(win)->\_parx:ERR)

#define \_\_NCURSES\_H 1

#define NCURSES\_EXPORT(type) type

#define NCURSES\_EXPORT\_VAR(type) type

#define WA\_ALTCHARSET A\_ALTCHARSET

#define WA\_ATTRIBUTES A\_ATTRIBUTES

#define WA\_BLINK A\_BLINK

#define WA\_BOLD A\_BOLD

#define WA\_DIM A\_DIM

#define WA\_HORIZONTAL A\_HORIZONTAL

#define WA\_INVIS A\_INVIS

#define WA\_LEFT A\_LEFT

#define WA\_LOW A\_LOW

#define WA\_NORMAL A\_NORMAL

#define WA\_PROTECT A\_PROTECT

#define WA\_REVERSE A\_REVERSE

#define WA\_RIGHT A\_RIGHT

#define WA\_STANDOUT A\_STANDOUT

#define WA\_TOP A\_TOP

#define WA\_UNDERLINE A\_UNDERLINE

#define WA\_VERTICAL A\_VERTICAL

#define A\_REVERSE NCURSES\_BITS(1UL,10)

#define COLOR\_BLACK 0

#define COLOR\_RED 1

#define COLOR\_GREEN 2

#define COLOR\_YELLOW 3

#define COLOR\_BLUE 4

#define COLOR\_MAGENTA 5

#define COLOR\_CYAN 6

#define COLOR\_WHITE 7

#define \_SUBWIN 0x01

#define \_ENDLINE 0x02

#define \_FULLWIN 0x04

#define \_SCROLLWIN 0x08

#define \_ISPAD 0x10

#define \_HASMOVED 0x20

typedef unsigned char bool;

typedef unsigned long int chtype;

typedef struct screen SCREEN;

typedef struct \_win\_st WINDOW;

typedef chtype attr\_t;

typedef struct {

attr\_t attr;

wchar\_t chars[5];

} cchar\_t;

struct pdat {

short \_pad\_y;

short \_pad\_x;

short \_pad\_top;

short \_pad\_left;

short \_pad\_bottom;

short \_pad\_right;

};

struct \_win\_st {

short \_cury; /\* current cursor position \*/

short \_curx;

short \_maxy; /\* maximums of x and y, NOT window size \*/

short \_maxx;

short \_begy; /\* screen coords of upper-left-hand corner \*/

short \_begx;

short \_flags; /\* window state flags \*/

attr\_t \_attrs; /\* current attribute for non-space character \*/

chtype \_bkgd; /\* current background char/attribute pair \*/

bool \_notimeout; /\* no time out on function-key entry? \*/

bool \_clear; /\* consider all data in the window invalid? \*/

bool \_leaveok; /\* OK to not reset cursor on exit? \*/

bool \_scroll; /\* OK to scroll this window? \*/

bool \_idlok; /\* OK to use insert/delete line? \*/

bool \_idcok; /\* OK to use insert/delete char? \*/

bool \_immed; /\* window in immed mode? (not yet used) \*/

bool \_sync; /\* window in sync mode? \*/

bool \_use\_keypad; /\* process function keys into KEY\_ symbols? \*/

int \_delay; /\* 0 = nodelay, <0 = blocking, >0 = delay \*/

struct ldat \*\_line; /\* the actual line data \*/

short \_regtop; /\* top line of scrolling region \*/

short \_regbottom; /\* bottom line of scrolling region \*/

int \_parx; /\* x coordinate of this window in parent \*/

int \_pary; /\* y coordinate of this window in parent \*/

WINDOW \*\_parent; /\* pointer to parent if a sub-window \*/

struct pdat \_pad;

short \_yoffset; /\* real begy is \_begy + \_yoffset \*/

cchar\_t \_bkgrnd; /\* current background char/attribute pair \*/

};

#define KEY\_F(n) (KEY\_F0+(n))

#define KEY\_CODE\_YES 0400

#define KEY\_BREAK 0401

#define KEY\_MIN 0401

#define KEY\_DOWN 0402

#define KEY\_UP 0403

#define KEY\_LEFT 0404

#define KEY\_RIGHT 0405

#define KEY\_HOME 0406

#define KEY\_BACKSPACE 0407

#define KEY\_F0 0410

#define KEY\_DL 0510

#define KEY\_IL 0511

#define KEY\_DC 0512

#define KEY\_IC 0513

#define KEY\_EIC 0514

#define KEY\_CLEAR 0515

#define KEY\_EOS 0516

#define KEY\_EOL 0517

#define KEY\_SF 0520

#define KEY\_SR 0521

#define KEY\_NPAGE 0522

#define KEY\_PPAGE 0523

#define KEY\_STAB 0524

#define KEY\_CTAB 0525

#define KEY\_CATAB 0526

#define KEY\_ENTER 0527

#define KEY\_SRESET 0530

#define KEY\_RESET 0531

#define KEY\_PRINT 0532

#define KEY\_LL 0533

#define KEY\_A1 0534

#define KEY\_A3 0535

#define KEY\_B2 0536

#define KEY\_C1 0537

#define KEY\_C3 0540

#define KEY\_BTAB 0541

#define KEY\_BEG 0542

#define KEY\_CANCEL 0543

#define KEY\_CLOSE 0544

#define KEY\_COMMAND 0545

#define KEY\_COPY 0546

#define KEY\_CREATE 0547

#define KEY\_END 0550

#define KEY\_EXIT 0551

#define KEY\_FIND 0552

#define KEY\_HELP 0553

#define KEY\_MARK 0554

#define KEY\_MESSAGE 0555

#define KEY\_MOVE 0556

#define KEY\_NEXT 0557

#define KEY\_OPEN 0560

#define KEY\_OPTIONS 0561

#define KEY\_PREVIOUS 0562

#define KEY\_REDO 0563

#define KEY\_REFERENCE 0564

#define KEY\_REFRESH 0565

#define KEY\_REPLACE 0566

#define KEY\_RESTART 0567

#define KEY\_RESUME 0570

#define KEY\_SAVE 0571

#define KEY\_SBEG 0572

#define KEY\_SCANCEL 0573

#define KEY\_SCOMMAND 0574

#define KEY\_SCOPY 0575

#define KEY\_SCREATE 0576

#define KEY\_SDC 0577

#define KEY\_SDL 0600

#define KEY\_SELECT 0601

#define KEY\_SEND 0602

#define KEY\_SEOL 0603

#define KEY\_SEXIT 0604

#define KEY\_SFIND 0605

#define KEY\_SHELP 0606

#define KEY\_SHOME 0607

#define KEY\_SIC 0610

#define KEY\_SLEFT 0611

#define KEY\_SMESSAGE 0612

#define KEY\_SMOVE 0613

#define KEY\_SNEXT 0614

#define KEY\_SOPTIONS 0615

#define KEY\_SPREVIOUS 0616

#define KEY\_SPRINT 0617

#define KEY\_SREDO 0620

#define KEY\_SREPLACE 0621

#define KEY\_SRIGHT 0622

#define KEY\_SRSUME 0623

#define KEY\_SSAVE 0624

#define KEY\_SSUSPEND 0625

#define KEY\_SUNDO 0626

#define KEY\_SUSPEND 0627

#define KEY\_UNDO 0630

#define KEY\_MOUSE 0631

#define KEY\_RESIZE 0632

#define KEY\_MAX 0777

#define PAIR\_NUMBER(a) (((a)&A\_COLOR)>>8)

#define NCURSES\_BITS(mask,shift) ((mask)<<((shift)+8))

#define A\_CHARTEXT (NCURSES\_BITS(1UL,0)-1UL)

#define A\_NORMAL 0L

#define NCURSES\_ATTR\_SHIFT 8

#define A\_COLOR NCURSES\_BITS(((1UL)<<8)-1UL,0)

#define A\_BLINK NCURSES\_BITS(1UL,11)

#define A\_DIM NCURSES\_BITS(1UL,12)

#define A\_BOLD NCURSES\_BITS(1UL,13)

#define A\_ALTCHARSET NCURSES\_BITS(1UL,14)

#define A\_INVIS NCURSES\_BITS(1UL,15)

#define A\_PROTECT NCURSES\_BITS(1UL,16)

#define A\_HORIZONTAL NCURSES\_BITS(1UL,17)

#define A\_LEFT NCURSES\_BITS(1UL,18)

#define A\_LOW NCURSES\_BITS(1UL,19)

#define A\_RIGHT NCURSES\_BITS(1UL,20)

#define A\_TOP NCURSES\_BITS(1UL,21)

#define A\_VERTICAL NCURSES\_BITS(1UL,22)

#define A\_STANDOUT NCURSES\_BITS(1UL,8)

#define A\_UNDERLINE NCURSES\_BITS(1UL,9)

#define COLOR\_PAIR(n) NCURSES\_BITS(n,0)

#define A\_ATTRIBUTES NCURSES\_BITS(~(1UL-1UL),0)

extern int COLORS;

extern int COLOR\_PAIRS;

extern int COLS;

extern int LINES;

extern chtype acs\_map[];

extern int addch(const chtype);

extern int addchnstr(const chtype \*, int);

extern int addchstr(const chtype \*);

extern int addnstr(const char \*, int);

extern int addstr(const char \*);

extern int attr\_get(attr\_t \*, short \*, void \*);

extern int attr\_off(attr\_t, void \*);

extern int attr\_on(attr\_t, void \*);

extern int attr\_set(attr\_t, short, void \*);

extern int attroff(int);

extern int attron(int);

extern int attrset(int);

extern int baudrate(void);

extern int beep(void);

extern int bkgd(chtype);

extern void bkgdset(chtype);

extern int border(chtype, chtype, chtype, chtype, chtype, chtype, chtype,

chtype);

extern int box(WINDOW \*, chtype, chtype);

extern bool can\_change\_color(void);

extern int cbreak(void);

extern int chgat(int, attr\_t, short, const void \*);

extern int clear(void);

extern int clearok(WINDOW \*, bool);

extern int clrtobot(void);

extern int clrtoeol(void);

extern int color\_content(short, short \*, short \*, short \*);

extern int color\_set(short, void \*);

extern int copywin(const WINDOW \*, WINDOW \*, int, int, int, int, int, int,

int);

extern int curs\_set(int);

extern WINDOW \*curscr;

extern int def\_prog\_mode(void);

extern int def\_shell\_mode(void);

extern int delay\_output(int);

extern int delch(void);

extern int deleteln(void);

extern void delscreen(SCREEN \*);

extern int delwin(WINDOW \*);

extern WINDOW \*derwin(WINDOW \*, int, int, int, int);

extern int doupdate(void);

extern WINDOW \*dupwin(WINDOW \*);

extern int echo(void);

extern int echochar(const chtype);

extern int endwin(void);

extern int erase(void);

extern char erasechar(void);

extern void filter(void);

extern int flash(void);

extern int flushinp(void);

extern chtype getbkgd(WINDOW \*);

extern int getch(void);

extern int getnstr(char \*, int);

extern int getstr(char \*);

extern WINDOW \*getwin(FILE \*);

extern int halfdelay(int);

extern bool has\_colors(void);

extern bool has\_ic(void);

extern bool has\_il(void);

extern int hline(chtype, int);

extern void idcok(WINDOW \*, bool);

extern int idlok(WINDOW \*, bool);

extern void immedok(WINDOW \*, bool);

extern chtype inch(void);

extern int inchnstr(chtype \*, int);

extern int inchstr(chtype \*);

extern int init\_color(short, short, short, short);

extern int init\_pair(short, short, short);

extern WINDOW \*initscr(void);

extern int innstr(char \*, int);

extern int insch(chtype);

extern int insdelln(int);

extern int insertln(void);

extern int insnstr(const char \*, int);

extern int insstr(const char \*);

extern int instr(char \*);

extern int intrflush(WINDOW \*, bool);

extern bool is\_linetouched(WINDOW \*, int);

extern bool is\_wintouched(WINDOW \*);

extern bool isendwin(void);

extern const char \*keyname(int);

extern int keypad(WINDOW \*, bool);

extern char killchar(void);

extern int leaveok(WINDOW \*, bool);

extern char \*longname(void);

extern int meta(WINDOW \*, bool);

extern int move(int, int);

extern int mvaddch(int, int, const chtype);

extern int mvaddchnstr(int, int, const chtype \*, int);

extern int mvaddchstr(int, int, const chtype \*);

extern int mvaddnstr(int, int, const char \*, int);

extern int mvaddstr(int, int, const char \*);

extern int mvchgat(int, int, int, attr\_t, short, const void \*);

extern int mvcur(int, int, int, int);

extern int mvdelch(int, int);

extern int mvderwin(WINDOW \*, int, int);

extern int mvgetch(int, int);

extern int mvgetnstr(int, int, char \*, int);

extern int mvgetstr(int, int, char \*);

extern int mvhline(int, int, chtype, int);

extern chtype mvinch(int, int);

extern int mvinchnstr(int, int, chtype \*, int);

extern int mvinchstr(int, int, chtype \*);

extern int mvinnstr(int, int, char \*, int);

extern int mvinsch(int, int, chtype);

extern int mvinsnstr(int, int, const char \*, int);

extern int mvinsstr(int, int, const char \*);

extern int mvinstr(int, int, char \*);

extern int mvprintw(int, int, const char \*, ...);

extern int mvscanw(int, int, const char \*, ...);

extern int mvvline(int, int, chtype, int);

extern int mvwaddch(WINDOW \*, int, int, const chtype);

extern int mvwaddchnstr(WINDOW \*, int, int, const chtype \*, int);

extern int mvwaddchstr(WINDOW \*, int, int, const chtype \*);

extern int mvwaddnstr(WINDOW \*, int, int, const char \*, int);

extern int mvwaddstr(WINDOW \*, int, int, const char \*);

extern int mvwchgat(WINDOW \*, int, int, int, attr\_t, short, const void \*);

extern int mvwdelch(WINDOW \*, int, int);

extern int mvwgetch(WINDOW \*, int, int);

extern int mvwgetnstr(WINDOW \*, int, int, char \*, int);

extern int mvwgetstr(WINDOW \*, int, int, char \*);

extern int mvwhline(WINDOW \*, int, int, chtype, int);

extern int mvwin(WINDOW \*, int, int);

extern chtype mvwinch(WINDOW \*, int, int);

extern int mvwinchnstr(WINDOW \*, int, int, chtype \*, int);

extern int mvwinchstr(WINDOW \*, int, int, chtype \*);

extern int mvwinnstr(WINDOW \*, int, int, char \*, int);

extern int mvwinsch(WINDOW \*, int, int, chtype);

extern int mvwinsnstr(WINDOW \*, int, int, const char \*, int);

extern int mvwinsstr(WINDOW \*, int, int, const char \*);

extern int mvwinstr(WINDOW \*, int, int, char \*);

extern int mvwprintw(WINDOW \*, int, int, const char \*, ...);

extern int mvwscanw(WINDOW \*, int, int, const char \*, ...);

extern int mvwvline(WINDOW \*, int, int, chtype, int);

extern int napms(int);

extern WINDOW \*newpad(int, int);

extern WINDOW \*newscr;

extern SCREEN \*newterm(const char \*, FILE \*, FILE \*);

extern WINDOW \*newwin(int, int, int, int);

extern int nl(void);

extern int nocbreak(void);

extern int nodelay(WINDOW \*, bool);

extern int noecho(void);

extern int nonl(void);

extern void noqiflush(void);

extern int noraw(void);

extern int notimeout(WINDOW \*, bool);

extern int overlay(const WINDOW \*, WINDOW \*);

extern int overwrite(const WINDOW \*, WINDOW \*);

extern int pair\_content(short, short \*, short \*);

extern int pechochar(WINDOW \*, chtype);

extern int pnoutrefresh(WINDOW \*, int, int, int, int, int, int);

extern int prefresh(WINDOW \*, int, int, int, int, int, int);

extern int printw(const char \*, ...);

extern int putwin(WINDOW \*, FILE \*);

extern void qiflush(void);

extern int raw(void);

extern int redrawwin(WINDOW \*);

extern int refresh(void);

extern int reset\_prog\_mode(void);

extern int reset\_shell\_mode(void);

extern int resetty(void);

extern int ripoffline(int, int (\*)(WINDOW \*, int));

extern int savetty(void);

extern int scanw(const char \*, ...);

extern int scr\_dump(const char \*);

extern int scr\_init(const char \*);

extern int scr\_restore(const char \*);

extern int scr\_set(const char \*);

extern int scrl(int);

extern int scroll(WINDOW \*);

extern int scrollok(WINDOW \*, bool);

extern SCREEN \*set\_term(SCREEN \*);

extern int setscrreg(int, int);

extern attr\_t slk\_attr(void);

extern int slk\_attr\_set(const attr\_t, short, void \*);

extern int slk\_attroff(const chtype);

extern int slk\_attron(const chtype);

extern int slk\_attrset(const chtype);

extern int slk\_clear(void);

extern int slk\_color(short);

extern int slk\_init(int);

extern char \*slk\_label(int);

extern int slk\_noutrefresh(void);

extern int slk\_refresh(void);

extern int slk\_restore(void);

extern int slk\_set(int, const char \*, int);

extern int slk\_touch(void);

extern int standend(void);

extern int standout(void);

extern int start\_color(void);

extern WINDOW \*stdscr;

extern WINDOW \*subpad(WINDOW \*, int, int, int, int);

extern WINDOW \*subwin(WINDOW \*, int, int, int, int);

extern int syncok(WINDOW \*, bool);

extern chtype termattrs(void);

extern char \*termname(void);

extern void timeout(int);

extern int touchline(WINDOW \*, int, int);

extern int touchwin(WINDOW \*);

extern int typeahead(int);

extern const char \*unctrl(chtype);

extern int ungetch(int);

extern int untouchwin(WINDOW \*);

extern void use\_env(bool);

extern int vidattr(chtype);

extern int vidputs(chtype, int (\*)(int));

extern int vline(chtype, int);

extern int vw\_printw(WINDOW \*, const char \*, va\_list);

extern int vw\_scanw(WINDOW \*, const char \*, va\_list);

extern int vwprintw(WINDOW \*, const char \*, va\_list);

extern int vwscanw(WINDOW \*, const char \*, va\_list);

extern int waddch(WINDOW \*, const chtype);

extern int waddchnstr(WINDOW \*, const chtype \*, int);

extern int waddchstr(WINDOW \*, const chtype \*);

extern int waddnstr(WINDOW \*, const char \*, int);

extern int waddstr(WINDOW \*, const char \*);

extern int wattr\_get(WINDOW \*, attr\_t \*, short \*, void \*);

extern int wattr\_off(WINDOW \*, attr\_t, void \*);

extern int wattr\_on(WINDOW \*, attr\_t, void \*);

extern int wattr\_set(WINDOW \*, attr\_t, short, void \*);

extern int wattroff(WINDOW \*, int);

extern int wattron(WINDOW \*, int);

extern int wattrset(WINDOW \*, int);

extern int wbkgd(WINDOW \*, chtype);

extern void wbkgdset(WINDOW \*, chtype);

extern int wborder(WINDOW \*, chtype, chtype, chtype, chtype, chtype,

chtype, chtype, chtype);

extern int wchgat(WINDOW \*, int, attr\_t, short, const void \*);

extern int wclear(WINDOW \*);

extern int wclrtobot(WINDOW \*);

extern int wclrtoeol(WINDOW \*);

extern int wcolor\_set(WINDOW \*, short, void \*);

extern void wcursyncup(WINDOW \*);

extern int wdelch(WINDOW \*);

extern int wdeleteln(WINDOW \*);

extern int wechochar(WINDOW \*, const chtype);

extern int werase(WINDOW \*);

extern int wgetch(WINDOW \*);

extern int wgetnstr(WINDOW \*, char \*, int);

extern int wgetstr(WINDOW \*, char \*);

extern int whline(WINDOW \*, chtype, int);

extern chtype winch(WINDOW \*);

extern int winchnstr(WINDOW \*, chtype \*, int);

extern int winchstr(WINDOW \*, chtype \*);

extern int winnstr(WINDOW \*, char \*, int);

extern int winsch(WINDOW \*, chtype);

extern int winsdelln(WINDOW \*, int);

extern int winsertln(WINDOW \*);

extern int winsnstr(WINDOW \*, const char \*, int);

extern int winsstr(WINDOW \*, const char \*);

extern int winstr(WINDOW \*, char \*);

extern int wmove(WINDOW \*, int, int);

extern int wnoutrefresh(WINDOW \*);

extern int wprintw(WINDOW \*, const char \*, ...);

extern int wredrawln(WINDOW \*, int, int);

extern int wrefresh(WINDOW \*);

extern int wscanw(WINDOW \*, const char \*, ...);

extern int wscrl(WINDOW \*, int);

extern int wsetscrreg(WINDOW \*, int, int);

extern int wstandend(WINDOW \*);

extern int wstandout(WINDOW \*);

extern void wsyncdown(WINDOW \*);

extern void wsyncup(WINDOW \*);

extern void wtimeout(WINDOW \*, int);

extern int wtouchln(WINDOW \*, int, int, int);

extern int wvline(WINDOW \*, chtype, int);

### **15.6.2 term.h**

extern TERMINAL \*cur\_term;

extern int del\_curterm(TERMINAL \*);

extern int putp(const char \*);

extern int restartterm(char \*, int, int \*);

extern TERMINAL \*set\_curterm(TERMINAL \*);

extern int setupterm(char \*, int, int \*);

extern int tgetent(char \*, const char \*);

extern int tgetflag(char \*);

extern int tgetnum(char \*);

extern char \*tgetstr(char \*, char \*\*);

extern char \*tgoto(const char \*, int, int);

extern int tigetflag(const char \*);

extern int tigetnum(const char \*);

extern char \*tigetstr(const char \*);

extern char \*tparm(const char \*, ...);

extern int tputs(const char \*, int, int (\*)(int));

extern char ttytype[];

## **15.7 Interface Definitions for libncurses**

The interfaces defined on the following pages are included in libncurses and are defined by this specification. Unless otherwise noted, these interfaces shall be included in the source standard.

Other interfaces listed in [Section 15.5](#ID_LIBNCURSES) shall behave as described in the referenced base document.

#### inchnstr

##### Name

inchnstr — obtain a string of characters and their attributes from a curses window

##### Synopsis

#include <curses.h>

int inchnstr(chtype \* *chstr*, int *n*);

##### Description

The interface inchnstr() shall behave as specified in [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES), except that inchnstr() shall return the number of characters that were read.

#### inchstr

##### Name

inchstr — obtain a string of characters and their attributes from a curses window

##### Synopsis

#include <curses.h>

int inchstr(chtype \* *chstr*);

##### Description

The interface inchstr() shall behave as specified in [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES), except that inchstr() shall return the number of characters that were read.

#### instr

##### Name

instr — obtain a string of characters from a curses window

##### Synopsis

#include <curses.h>

int instr(char \* *str*);

##### Description

The interface instr() shall behave as specified in [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES), except that instr() shall return the number of characters that were read.

#### mvcur

##### Name

mvcur — send cursor movement commands to terminal

##### Synopsis

#include <curses.h>

int mvcur(int *oldrow*, int *oldcol*, int *newrow*, int *newcol*);

##### Description

The interface mvcur() shall behave as described in [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES), except that if (*newrow*, *newcol*) is not a valid address for the terminal in use, the results of the mvcur() function are unspecified.

#### mvinchnstr

##### Name

mvinchnstr — obtain a string of characters and their attributes from a curses window

##### Synopsis

#include <curses.h>

int mvinchnstr(int *y*, int *x*, chtype \* *chstr*, int *n*);

##### Description

The interface mvinchnstr() shall behave as specified in [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES), except that mvinchnstr() shall return the number of characters that were read.

#### mvinchstr

##### Name

mvinchstr — obtain a string of characters and their attributes from a curses window

##### Synopsis

#include <curses.h>

int mvinchstr(int *y*, int *x*, chtype \* *chstr*);

##### Description

The interface mvinchstr() shall behave as specified in [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES), except that mvinchstr() shall return the number of characters that were read.

#### mvinstr

##### Name

mvinstr — obtain a string of characters from a curses window

##### Synopsis

#include <curses.h>

int mvinstr(int *y*, int *x*, char \* *str*);

##### Description

The interface mvinstr() shall behave as specified in [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES), except that mvinstr() shall return the number of characters that were read.

#### mvscanw

##### Name

mvscanw — convert formatted input from a curses window

##### Synopsis

#include <curses.h>

int mvscanw(int *y*, int *x*, const char \**fmt*, *...*);

##### Description

The scanw family of functions shall behave as described in [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES), except as noted below.

##### Differences

This function returns ERR on failure. On success it returns the number of successfully matched and assigned input items. This differs from [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES), which indicates this function returns OK on success.

#### mvwinchnstr

##### Name

mvwinchnstr — obtain a string of characters and their attributes from a curses window

##### Synopsis

#include <curses.h>

int mvwinchnstr(WINDOW \* *win*, int *y*, int *x*, chtype \* *chstr*, int *n*);

##### Description

The interface mvwinchnstr() shall behave as specified in [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES), except that mvwinchnstr() shall return the number of characters that were read.

#### mvwinchstr

##### Name

mvwinchstr — obtain a string of characters and their attributes from a curses window

##### Synopsis

#include <curses.h>

int mvwinchstr(WINDOW \* *win*, int *y*, int *x*, chtype \* *chstr*);

##### Description

The interface mvwinchstr() shall behave as specified in [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES), except that mvwinchstr() shall return the number of characters that were read.

#### mvwinstr

##### Name

mvwinstr — obtain a string of characters from a curses window

##### Synopsis

#include <curses.h>

int mvwinstr(WINDOW \* *win*, int *y*, int *x*, char \* *str*);

##### Description

The interface mvwinstr() shall behave as specified in [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES), except that mvwinstr() shall return the number of characters that were read.

#### mvwscanw

##### Name

mvwscanw — convert formatted input from a curses window

##### Synopsis

#include <curses.h>

int mvwscanw(WINDOW \**win*, int *y*, int *x*, const char \**fmt*, *...*);

##### Description

The scanw family of functions shall behave as described in [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES), except as noted below.

##### Differences

This function returns ERR on failure. On success it returns the number of successfully matched and assigned input items. This differs from [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES), which indicates this function returns OK on success.

#### ripoffline

##### Name

ripoffline — obtain a string of characters and their attributes from a curses window

##### Synopsis

#include <curses.h>

int ripoffline(int *line*, int (\**init*) (WINDOW \*, int));

##### Description

The interface ripoffline() shall behave as specified in [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES), except that ripoffline() shall return -1 if the number of lines that were ripped off exceeds five.

#### scanw

##### Name

scanw — convert formatted input from a curses window

##### Synopsis

#include <curses.h>

int scanw(const char \**fmt*, *...*);

##### Description

The scanw family of functions shall behave as described in [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES), except as noted below.

##### Differences

This function returns ERR on failure. On success it returns the number of successfully matched and assigned input items. This differs from [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES), which indicates this function returns OK on success.

#### vw\_scanw

##### Name

vw\_scanw — convert formatted input from a curses window

##### Synopsis

#include <curses.h>

int vw\_scanw(WINDOW \**win*, const char \**fmt*, va\_list *vararglist*);

##### Description

The scanw family of functions shall behave as described in [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES), except as noted below.

##### Differences

This function returns ERR on failure. On success it returns the number of successfully matched and assigned input items. This differs from [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES), which indicates this function returns OK on success.

#### vwscanw

##### Name

vwscanw — convert formatted input from a curses window

##### Synopsis

#include <curses.h>

int vw\_scanw(WINDOW \**win*, const char \**fmt*, va\_list *vararglist*);

##### Description

The scanw family of functions shall behave as described in [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES), except as noted below.

##### Differences

This function returns ERR on failure. On success it returns the number of successfully matched and assigned input items. This differs from [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES), which indicates this function returns OK on success.

#### winchnstr

##### Name

winchnstr — obtain a string of characters and their attributes from a curses window

##### Synopsis

#include <curses.h>

int winchnstr(WINDOW \* *win*, chtype \* *chstr*, int *n*);

##### Description

The interface winchnstr() shall behave as specified in [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES), except that winchnstr() shall return the number of characters that were read.

#### winchstr

##### Name

winchstr — obtain a string of characters and their attributes from a curses window

##### Synopsis

#include <curses.h>

int winchstr(WINDOW \* *win*, chtype \* *chstr*);

##### Description

The interface winchstr() shall behave as specified in [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES), except that winchstr() shall return the number of characters that were read.

#### winstr

##### Name

winstr — obtain a string of characters from a curses window

##### Synopsis

#include <curses.h>

int winstr(WINDOW \* *win*, char \* *str*);

##### Description

The interface winstr() shall behave as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except that winstr() shall return the number of characters that were read.

#### wscanw

##### Name

wscanw — convert formatted input from a curses window

##### Synopsis

#include <curses.h>

int wscanw(WINDOW \**win*, const char \**fmt*, *...*);

##### Description

The scanw family of functions shall behave as described in [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES), except as noted below.

##### Differences

This function returns ERR on failure. On success it returns the number of successfully matched and assigned input items. This differs from [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES), which indicates this function returns OK on success.

## **15.8 Interfaces for libncursesw**

[Table 15-7](#ID_LIB_45_LIBNCURSESW_45_DEF) defines the library name and shared object name for the libncursesw library

**Table 15-7 libncursesw Definition**

|  |  |
| --- | --- |
| Library: | libncursesw |
| SONAME: | libncursesw.so.5 |

The behavior of the interfaces in this library is specified by the following specifications:

|  |
| --- |
| [Libncursesw] [Libncursesw API](#ID_STD_46_LIBNCURSESW) |
| [LSB] [This Specification](#ID_STD_46_LSB) |
| [ncursesw] [Libncursesw Placeholder](#ID_STD_46_NCURSESW) |
| [X-CURSES] [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES) |

### **15.8.1 Curses Wide**

#### 15.8.1.1 Interfaces for Curses Wide

An LSB conforming implementation shall provide the generic functions for Curses Wide specified in [Table 15-8](#ID_TBL_45_LIBNCURSESW_45_CURSE_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 15-8 libncursesw - Curses Wide Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| add\_wch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | add\_wchnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | add\_wchstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | addch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| addchnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | addchstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | addnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | addnwstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| addstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | addwstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | assume\_default\_colors [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | attr\_get [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| attr\_off [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | attr\_on [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | attr\_set [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | attroff [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| attron [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | attrset [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | baudrate [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | beep [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| bkgd [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | bkgdset [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | bkgrnd [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | bkgrndset [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| border [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | border\_set [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | box [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | box\_set [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| can\_change\_color [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | cbreak [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | chgat [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | clear [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| clearok [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | clrtobot [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | clrtoeol [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | color\_content [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| color\_set [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | copywin [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | curs\_set [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | curses\_version [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| def\_prog\_mode [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | def\_shell\_mode [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | define\_key [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | del\_curterm [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| delay\_output [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | delch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | deleteln [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | delscreen [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| delwin [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | derwin [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | doupdate [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | dupwin [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| echo [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | echo\_wchar [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | echochar [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | endwin [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| erase [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | erasechar [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | erasewchar [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | filter [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| flash [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | flushinp [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | get\_wch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | get\_wstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| getbkgd [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | getbkgrnd [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | getcchar [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | getch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| getmouse [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | getn\_wstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | getnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | getstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| getwin [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | halfdelay [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | has\_colors [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | has\_ic [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| has\_il [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | has\_key [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | has\_mouse [[LSB]](#ID_REFSTD_46_LIBNCURSESW_46_2) | hline [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| hline\_set [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | idcok [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | idlok [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | immedok [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| in\_wch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | in\_wchnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | in\_wchstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | inch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| inchnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | inchstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | init\_color [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | init\_pair [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| initscr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | innstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | innwstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | ins\_nwstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| ins\_wch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | ins\_wstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | insch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | insdelln [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| insertln [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | insnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | insstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | instr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| intrflush [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | inwstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | is\_linetouched [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | is\_wintouched [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| isendwin [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | key\_name [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | keybound [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | keyname [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| keyok [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | keypad [[LSB]](#ID_REFSTD_46_LIBNCURSESW_46_2) | killchar [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | killwchar [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| leaveok [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | longname [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mcprint [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | meta [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| mouse\_trafo [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mouseinterval [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mousemask [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | move [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| mvadd\_wch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvadd\_wchnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvadd\_wchstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvaddch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| mvaddchnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvaddchstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvaddnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvaddnwstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| mvaddstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvaddwstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvchgat [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvcur [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| mvdelch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvderwin [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvget\_wch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvget\_wstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| mvgetch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvgetn\_wstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvgetnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvgetstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| mvhline [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvhline\_set [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvin\_wch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvin\_wchnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| mvin\_wchstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvinch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvinchnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvinchstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| mvinnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvinnwstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvins\_nwstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvins\_wch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| mvins\_wstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvinsch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvinsnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvinsstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| mvinstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvinwstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvprintw [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvscanw [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| mvvline [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvvline\_set [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwadd\_wch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwadd\_wchnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| mvwadd\_wchstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwaddch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwaddchnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwaddchstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| mvwaddnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwaddnwstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwaddstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwaddwstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| mvwchgat [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwdelch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwget\_wch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwget\_wstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| mvwgetch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwgetn\_wstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwgetnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwgetstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| mvwhline [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwhline\_set [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwin [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwin\_wch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| mvwin\_wchnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwin\_wchstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwinch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwinchnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| mvwinchstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwinnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwinnwstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwins\_nwstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| mvwins\_wch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwins\_wstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwinsch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwinsnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| mvwinsstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwinstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwinwstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwprintw [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| mvwscanw [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwvline [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | mvwvline\_set [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | napms [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| newpad [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | newterm [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | newwin [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | nl [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| nocbreak [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | nodelay [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | noecho [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | nonl [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| noqiflush [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | noraw [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | notimeout [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | overlay [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| overwrite [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | pair\_content [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | pecho\_wchar [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | pechochar [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| pnoutrefresh [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | prefresh [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | printw [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | putp [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| putwin [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | qiflush [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | raw [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | redrawwin [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| refresh [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | reset\_prog\_mode [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | reset\_shell\_mode [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | resetty [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| resizeterm [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | restartterm [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | ripoffline [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | savetty [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| scanw [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | scr\_dump [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | scr\_init [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | scr\_restore [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| scr\_set [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | scrl [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | scroll [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | scrollok [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| set\_curterm [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | set\_term [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | setcchar [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | setscrreg [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| setupterm [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | slk\_attr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSESW_46_4) | slk\_attr\_off [[X-CURSES]](#ID_REFSTD_46_LIBNCURSESW_46_4) | slk\_attr\_on [[X-CURSES]](#ID_REFSTD_46_LIBNCURSESW_46_4) |
| slk\_attr\_set [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | slk\_attroff [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | slk\_attron [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | slk\_attrset [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| slk\_clear [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | slk\_color [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | slk\_init [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | slk\_label [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| slk\_noutrefresh [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | slk\_refresh [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | slk\_restore [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | slk\_set [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| slk\_touch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | slk\_wset [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | standend [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | standout [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| start\_color [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | subpad [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | subwin [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | syncok [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| term\_attrs [[X-CURSES]](#ID_REFSTD_46_LIBNCURSESW_46_4) | termattrs [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | termname [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | tgetent [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| tgetflag [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | tgetnum [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | tgetstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | tgoto [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| tigetflag [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | tigetnum [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | tigetstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | timeout [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| touchline [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | touchwin [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | tparm [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | tputs [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| typeahead [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | unctrl [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | unget\_wch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | ungetch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| ungetmouse [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | untouchwin [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | use\_default\_colors [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | use\_env [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| use\_extended\_names [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | vid\_attr [[X-CURSES]](#ID_REFSTD_46_LIBNCURSESW_46_4) | vid\_puts [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | vidattr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| vidputs [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | vline [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | vline\_set [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | vw\_printw [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| vw\_scanw [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | vwprintw [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | vwscanw [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wadd\_wch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| wadd\_wchnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wadd\_wchstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | waddch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | waddchnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| waddchstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | waddnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | waddnwstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | waddstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| waddwstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wattr\_get [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wattr\_off [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wattr\_on [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| wattr\_set [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wattroff [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wattron [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wattrset [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| wbkgd [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wbkgdset [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wbkgrnd [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wbkgrndset [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| wborder [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wborder\_set [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wchgat [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wclear [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| wclrtobot [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wclrtoeol [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wcolor\_set [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wcursyncup [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| wdelch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wdeleteln [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wecho\_wchar [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wechochar [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| werase [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wget\_wch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wget\_wstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wgetbkgrnd [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| wgetch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wgetn\_wstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wgetnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wgetstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| whline [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | whline\_set [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | win\_wch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | win\_wchnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| win\_wchstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | winch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | winchnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | winchstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| winnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | winnwstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wins\_nwstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wins\_wch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| wins\_wstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | winsch [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | winsdelln [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | winsertln [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| winsnstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | winsstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | winstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | winwstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| wmouse\_trafo [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wmove [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wnoutrefresh [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wprintw [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| wredrawln [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wrefresh [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wresize [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wscanw [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| wscrl [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wsetscrreg [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wstandend [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wstandout [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| wsyncdown [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wsyncup [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wtimeout [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wtouchln [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| wunctrl [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wvline [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | wvline\_set [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |  |

An LSB conforming implementation shall provide the generic deprecated functions for Curses Wide specified in [Table 15-9](#ID_TBL_45_LIBNCURSESW_45_CURSE_45_DEPINTS), with the full mandatory functionality as described in the referenced underlying specification.

**Note:** These interfaces are deprecated, and applications should avoid using them. These interfaces may be withdrawn in future releases of this specification.

**Table 15-9 libncursesw - Curses Wide Deprecated Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| tgetent [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | tgetflag [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | tgetnum [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) | tgetstr [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |
| tgoto [[Libncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_1) |  |  |  |

An LSB conforming implementation shall provide the generic data interfaces for Curses Wide specified in [Table 15-10](#ID_TBL_45_LIBNCURSESW_45_CURSE_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 15-10 libncursesw - Curses Wide Data Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| COLORS [[ncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_3) | COLOR\_PAIRS [[ncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_3) | COLS [[ncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_3) | LINES [[ncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_3) |
| acs\_map [[LSB]](#ID_REFSTD_46_LIBNCURSESW_46_2) | cur\_term [[LSB]](#ID_REFSTD_46_LIBNCURSESW_46_2) | curscr [[ncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_3) | newscr [[ncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_3) |
| stdscr [[ncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_3) | ttytype [[ncursesw]](#ID_REFSTD_46_LIBNCURSESW_46_3) |  |  |

## **15.9 Data Definitions for libncursesw**

This section defines global identifiers and their values that are associated with interfaces contained in libncursesw. These definitions are organized into groups that correspond to system headers. This convention is used as a convenience for the reader, and does not imply the existence of these headers, or their content. Where an interface is defined as requiring a particular system header file all of the data definitions for that system header file presented here shall be in effect.

This section gives data definitions to promote binary application portability, not to repeat source interface definitions available elsewhere. System providers and application developers should use this ABI to supplement - not to replace - source interface definition specifications.

This specification uses the [ISO C (1999)](#ID_STD_46_ISOC99) C Language as the reference programming language, and data definitions are specified in ISO C format. The C language is used here as a convenient notation. Using a C language description of these data objects does not preclude their use by other programming languages.

### **15.9.1 ncursesw/curses.h**

#define CURSES 1

#define setsyx(y,x) do { if (newscr) { \

if ((y) == -1 && (x) == -1) \

leaveok(newscr, TRUE); \

else { \

leaveok(newscr, FALSE); \

wmove(newscr, (y), (x)); \

} \

} \

} while(0)

#define getsyx(y,x) do { if (newscr) { \

if (is\_leaveok(newscr)) \

(y) = (x) = -1; \

else \

getyx(newscr,(y), (x)); \

} \

} while(0)

#define CURSES\_H 1

#define NCURSES\_VERSION\_MAJOR 5

#define NCURSES\_VERSION\_MINOR 9

#define NCURSES\_VERSION\_PATCH 20110404

#define NCURSES\_VERSION "5.9"

#define NCURSES\_MOUSE\_VERSION 1

#define NCURSES\_ENABLE\_STDBOOL\_H 1

#define NCURSES\_INLINE inline

#define NCURSES\_TPARM\_VARARGS 1

#ifndef TRUE

#define TRUE 1

#endif

#define NCURSES\_BOOL bool

#ifdef \_\_cplusplus

# define NCURSES\_CAST(type,value) static\_cast<type>(value)

#else

# define NCURSES\_CAST(type,value) (type)(value)

#endif

#define WA\_ATTRIBUTES A\_ATTRIBUTES

#define WA\_NORMAL A\_NORMAL

#define WA\_STANDOUT A\_STANDOUT

#define WA\_UNDERLINE A\_UNDERLINE

#define WA\_REVERSE A\_REVERSE

#define WA\_BLINK A\_BLINK

#define WA\_DIM A\_DIM

#define WA\_BOLD A\_BOLD

#define WA\_ALTCHARSET A\_ALTCHARSET

#define WA\_INVIS A\_INVIS

#define WA\_PROTECT A\_PROTECT

#define WA\_HORIZONTAL A\_HORIZONTAL

#define WA\_LEFT A\_LEFT

#define WA\_LOW A\_LOW

#define WA\_RIGHT A\_RIGHT

#define WA\_TOP A\_TOP

#define WA\_VERTICAL A\_VERTICAL

#define COLOR\_BLACK 0

#define COLOR\_RED 1

#define COLOR\_GREEN 2

#define COLOR\_YELLOW 3

#define COLOR\_BLUE 4

#define COLOR\_MAGENTA 5

#define COLOR\_CYAN 6

#define COLOR\_WHITE 7

#define NCURSES\_ACS(c) (acs\_map[NCURSES\_CAST(unsigned char,c)])

#define ACS\_ULCORNER NCURSES\_ACS('l')

#define ACS\_LLCORNER NCURSES\_ACS('m')

#define ACS\_URCORNER NCURSES\_ACS('k')

#define ACS\_LRCORNER NCURSES\_ACS('j')

#define ACS\_LTEE NCURSES\_ACS('t')

#define ACS\_RTEE NCURSES\_ACS('u')

#define ACS\_BTEE NCURSES\_ACS('v')

#define ACS\_TTEE NCURSES\_ACS('w')

#define ACS\_HLINE NCURSES\_ACS('q')

#define ACS\_VLINE NCURSES\_ACS('x')

#define ACS\_PLUS NCURSES\_ACS('n')

#define ACS\_S1 NCURSES\_ACS('o')

#define ACS\_S9 NCURSES\_ACS('s')

#define ACS\_DIAMOND NCURSES\_ACS('`')

#define ACS\_CKBOARD NCURSES\_ACS('a')

#define ACS\_DEGREE NCURSES\_ACS('f')

#define ACS\_PLMINUS NCURSES\_ACS('g')

#define ACS\_BULLET NCURSES\_ACS('~')

#define ACS\_LARROW NCURSES\_ACS(',')

#define ACS\_RARROW NCURSES\_ACS('+')

#define ACS\_DARROW NCURSES\_ACS('.')

#define ACS\_UARROW NCURSES\_ACS('-')

#define ACS\_BOARD NCURSES\_ACS('h')

#define ACS\_LANTERN NCURSES\_ACS('i')

#define ACS\_BLOCK NCURSES\_ACS('0')

#define ACS\_S3 NCURSES\_ACS('p')

#define ACS\_S7 NCURSES\_ACS('r')

#define ACS\_LEQUAL NCURSES\_ACS('y')

#define ACS\_GEQUAL NCURSES\_ACS('z')

#define ACS\_PI NCURSES\_ACS('{')

#define ACS\_NEQUAL NCURSES\_ACS('|')

#define ACS\_STERLING NCURSES\_ACS('}')

#define ACS\_BSSB ACS\_ULCORNER

#define ACS\_SSBB ACS\_LLCORNER

#define ACS\_BBSS ACS\_URCORNER

#define ACS\_SBBS ACS\_LRCORNER

#define ACS\_SBSS ACS\_RTEE

#define ACS\_SSSB ACS\_LTEE

#define ACS\_SSBS ACS\_BTEE

#define ACS\_BSSS ACS\_TTEE

#define ACS\_BSBS ACS\_HLINE

#define ACS\_SBSB ACS\_VLINE

#define ACS\_SSSS ACS\_PLUS

#define ERR (-1)

#define OK (0)

#define \_SUBWIN 0x01

#define \_ENDLINE 0x02

#define \_FULLWIN 0x04

#define \_SCROLLWIN 0x08

#define \_ISPAD 0x10

#define \_HASMOVED 0x20

#define \_WRAPPED 0x40

#define \_NOCHANGE -1

#define \_NEWINDEX -1

#define CCHARW\_MAX 5

#define NCURSES\_EXT\_COLORS 20110404

#define GCC\_PRINTFLIKE(fmt,var) \_\_attribute\_\_((format(printf,fmt,var)))

#define GCC\_SCANFLIKE(fmt,var) \_\_attribute\_\_((format(scanf,fmt,var)))

#define NCURSES\_EXT\_FUNCS 20110404

#define curses\_version() NCURSES\_VERSION

#define NCURSES\_SP\_FUNCS 20110404

#define NCURSES\_SP\_OUTC NCURSES\_SP\_NAME(NCURSES\_OUTC)

#define NCURSES\_SP\_NAME(name) name

#define NCURSES\_ATTR\_SHIFT 8

#define NCURSES\_BITS(mask,shift) ((mask) << ((shift) + NCURSES\_ATTR\_SHIFT))

#define A\_NORMAL (1UL - 1UL)

#define A\_ATTRIBUTES NCURSES\_BITS(~(1UL - 1UL),0)

#define A\_CHARTEXT (NCURSES\_BITS(1UL,0) - 1UL)

#define A\_COLOR NCURSES\_BITS(((1UL) << 8) - 1UL,0)

#define A\_STANDOUT NCURSES\_BITS(1UL,8)

#define A\_UNDERLINE NCURSES\_BITS(1UL,9)

#define A\_REVERSE NCURSES\_BITS(1UL,10)

#define A\_BLINK NCURSES\_BITS(1UL,11)

#define A\_DIM NCURSES\_BITS(1UL,12)

#define A\_BOLD NCURSES\_BITS(1UL,13)

#define A\_ALTCHARSET NCURSES\_BITS(1UL,14)

#define A\_INVIS NCURSES\_BITS(1UL,15)

#define A\_PROTECT NCURSES\_BITS(1UL,16)

#define A\_HORIZONTAL NCURSES\_BITS(1UL,17)

#define A\_LEFT NCURSES\_BITS(1UL,18)

#define A\_LOW NCURSES\_BITS(1UL,19)

#define A\_RIGHT NCURSES\_BITS(1UL,20)

#define A\_TOP NCURSES\_BITS(1UL,21)

#define A\_VERTICAL NCURSES\_BITS(1UL,22)

#define getyx(win,y,x) (y = getcury(win), x = getcurx(win))

#define getbegyx(win,y,x) (y = getbegy(win), x = getbegx(win))

#define getmaxyx(win,y,x) (y = getmaxy(win), x = getmaxx(win))

#define getparyx(win,y,x) (y = getpary(win), x = getparx(win))

#define wgetstr(w, s) wgetnstr(w, s, -1)

#define getnstr(s, n) wgetnstr(stdscr, s, n)

#define setterm(term) setupterm(term, 1, (int \*)0)

#define fixterm() reset\_prog\_mode()

#define resetterm() reset\_shell\_mode()

#define saveterm() def\_prog\_mode()

#define crmode() cbreak()

#define nocrmode() nocbreak()

#define getattrs(win) NCURSES\_CAST(int, (win) ? (win)->\_attrs : A\_NORMAL)

#define getcurx(win) ((win) ? (win)->\_curx : ERR)

#define getcury(win) ((win) ? (win)->\_cury : ERR)

#define getbegx(win) ((win) ? (win)->\_begx : ERR)

#define getbegy(win) ((win) ? (win)->\_begy : ERR)

#define getmaxx(win) ((win) ? ((win)->\_maxx + 1) : ERR)

#define getmaxy(win) ((win) ? ((win)->\_maxy + 1) : ERR)

#define getparx(win) ((win) ? (win)->\_parx : ERR)

#define getpary(win) ((win) ? (win)->\_pary : ERR)

#define wstandout(win) (wattrset(win,A\_STANDOUT))

#define wstandend(win) (wattrset(win,A\_NORMAL))

#define wattron(win,at) wattr\_on(win, NCURSES\_CAST(attr\_t, at), NULL)

#define wattroff(win,at) wattr\_off(win, NCURSES\_CAST(attr\_t, at), NULL)

#define scroll(win) wscrl(win,1)

#define touchwin(win) wtouchln((win), 0, getmaxy(win), 1)

#define touchline(win, s, c) wtouchln((win), s, c, 1)

#define untouchwin(win) wtouchln((win), 0, getmaxy(win), 0)

#define box(win, v, h) wborder(win, v, v, h, h, 0, 0, 0, 0)

#define border(ls, rs, ts, bs, tl, tr, bl, br) wborder(stdscr, ls, rs, ts, bs, tl, tr, bl, br)

#define hline(ch, n) whline(stdscr, ch, n)

#define vline(ch, n) wvline(stdscr, ch, n)

#define winstr(w, s) winnstr(w, s, -1)

#define winchstr(w, s) winchnstr(w, s, -1)

#define winsstr(w, s) winsnstr(w, s, -1)

#define redrawwin(win) wredrawln(win, 0, (win)->\_maxy+1)

#define waddstr(win,str) waddnstr(win,str,-1)

#define waddchstr(win,str) waddchnstr(win,str,-1)

#define COLOR\_PAIR(n) NCURSES\_BITS(n, 0)

#define PAIR\_NUMBER(a) (NCURSES\_CAST(int,((NCURSES\_CAST(unsigned long,a) & A\_COLOR) >> NCURSES\_ATTR\_SHIFT)))

#define addch(ch) waddch(stdscr,ch)

#define addchnstr(str,n) waddchnstr(stdscr,str,n)

#define addchstr(str) waddchstr(stdscr,str)

#define addnstr(str,n) waddnstr(stdscr,str,n)

#define addstr(str) waddnstr(stdscr,str,-1)

#define attroff(at) wattroff(stdscr,at)

#define attron(at) wattron(stdscr,at)

#define attrset(at) wattrset(stdscr,at)

#define attr\_get(ap,cp,o) wattr\_get(stdscr,ap,cp,o)

#define attr\_off(a,o) wattr\_off(stdscr,a,o)

#define attr\_on(a,o) wattr\_on(stdscr,a,o)

#define attr\_set(a,c,o) wattr\_set(stdscr,a,c,o)

#define bkgd(ch) wbkgd(stdscr,ch)

#define bkgdset(ch) wbkgdset(stdscr,ch)

#define chgat(n,a,c,o) wchgat(stdscr,n,a,c,o)

#define clear() wclear(stdscr)

#define clrtobot() wclrtobot(stdscr)

#define clrtoeol() wclrtoeol(stdscr)

#define color\_set(c,o) wcolor\_set(stdscr,c,o)

#define delch() wdelch(stdscr)

#define deleteln() winsdelln(stdscr,-1)

#define echochar(c) wechochar(stdscr,c)

#define erase() werase(stdscr)

#define getch() wgetch(stdscr)

#define getstr(str) wgetstr(stdscr,str)

#define inch() winch(stdscr)

#define inchnstr(s,n) winchnstr(stdscr,s,n)

#define inchstr(s) winchstr(stdscr,s)

#define innstr(s,n) winnstr(stdscr,s,n)

#define insch(c) winsch(stdscr,c)

#define insdelln(n) winsdelln(stdscr,n)

#define insertln() winsdelln(stdscr,1)

#define insnstr(s,n) winsnstr(stdscr,s,n)

#define insstr(s) winsstr(stdscr,s)

#define instr(s) winstr(stdscr,s)

#define move(y,x) wmove(stdscr,y,x)

#define refresh() wrefresh(stdscr)

#define scrl(n) wscrl(stdscr,n)

#define setscrreg(t,b) wsetscrreg(stdscr,t,b)

#define standend() wstandend(stdscr)

#define standout() wstandout(stdscr)

#define timeout(delay) wtimeout(stdscr,delay)

#define wdeleteln(win) winsdelln(win,-1)

#define winsertln(win) winsdelln(win,1)

#define mvwaddch(win,y,x,ch) (wmove(win,y,x) == ERR ? ERR : waddch(win,ch))

#define mvwaddchnstr(win,y,x,str,n) (wmove(win,y,x) == ERR ? ERR : waddchnstr(win,str,n))

#define mvwaddchstr(win,y,x,str) (wmove(win,y,x) == ERR ? ERR : waddchnstr(win,str,-1))

#define mvwaddnstr(win,y,x,str,n) (wmove(win,y,x) == ERR ? ERR : waddnstr(win,str,n))

#define mvwaddstr(win,y,x,str) (wmove(win,y,x) == ERR ? ERR : waddnstr(win,str,-1))

#define mvwdelch(win,y,x) (wmove(win,y,x) == ERR ? ERR : wdelch(win))

#define mvwchgat(win,y,x,n,a,c,o) (wmove(win,y,x) == ERR ? ERR : wchgat(win,n,a,c,o))

#define mvwgetch(win,y,x) (wmove(win,y,x) == ERR ? ERR : wgetch(win))

#define mvwgetnstr(win,y,x,str,n) (wmove(win,y,x) == ERR ? ERR : wgetnstr(win,str,n))

#define mvwgetstr(win,y,x,str) (wmove(win,y,x) == ERR ? ERR : wgetstr(win,str))

#define mvwhline(win,y,x,c,n) (wmove(win,y,x) == ERR ? ERR : whline(win,c,n))

#define mvwinch(win,y,x) (wmove(win,y,x) == ERR ? NCURSES\_CAST(chtype, ERR) : winch(win))

#define mvwinchnstr(win,y,x,s,n) (wmove(win,y,x) == ERR ? ERR : winchnstr(win,s,n))

#define mvwinchstr(win,y,x,s) (wmove(win,y,x) == ERR ? ERR : winchstr(win,s))

#define mvwinnstr(win,y,x,s,n) (wmove(win,y,x) == ERR ? ERR : winnstr(win,s,n))

#define mvwinsch(win,y,x,c) (wmove(win,y,x) == ERR ? ERR : winsch(win,c))

#define mvwinsnstr(win,y,x,s,n) (wmove(win,y,x) == ERR ? ERR : winsnstr(win,s,n))

#define mvwinsstr(win,y,x,s) (wmove(win,y,x) == ERR ? ERR : winsstr(win,s))

#define mvwinstr(win,y,x,s) (wmove(win,y,x) == ERR ? ERR : winstr(win,s))

#define mvwvline(win,y,x,c,n) (wmove(win,y,x) == ERR ? ERR : wvline(win,c,n))

#define mvaddch(y,x,ch) mvwaddch(stdscr,y,x,ch)

#define mvaddchnstr(y,x,str,n) mvwaddchnstr(stdscr,y,x,str,n)

#define mvaddchstr(y,x,str) mvwaddchstr(stdscr,y,x,str)

#define mvaddnstr(y,x,str,n) mvwaddnstr(stdscr,y,x,str,n)

#define mvaddstr(y,x,str) mvwaddstr(stdscr,y,x,str)

#define mvchgat(y,x,n,a,c,o) mvwchgat(stdscr,y,x,n,a,c,o)

#define mvdelch(y,x) mvwdelch(stdscr,y,x)

#define mvgetch(y,x) mvwgetch(stdscr,y,x)

#define mvgetnstr(y,x,str,n) mvwgetnstr(stdscr,y,x,str,n)

#define mvgetstr(y,x,str) mvwgetstr(stdscr,y,x,str)

#define mvhline(y,x,c,n) mvwhline(stdscr,y,x,c,n)

#define mvinch(y,x) mvwinch(stdscr,y,x)

#define mvinchnstr(y,x,s,n) mvwinchnstr(stdscr,y,x,s,n)

#define mvinchstr(y,x,s) mvwinchstr(stdscr,y,x,s)

#define mvinnstr(y,x,s,n) mvwinnstr(stdscr,y,x,s,n)

#define mvinsch(y,x,c) mvwinsch(stdscr,y,x,c)

#define mvinsnstr(y,x,s,n) mvwinsnstr(stdscr,y,x,s,n)

#define mvinsstr(y,x,s) mvwinsstr(stdscr,y,x,s)

#define mvinstr(y,x,s) mvwinstr(stdscr,y,x,s)

#define mvvline(y,x,c,n) mvwvline(stdscr,y,x,c,n)

#define getbkgd(win) ((win)->\_bkgd)

#define slk\_attr\_off(a,v) ((v) ? ERR : slk\_attroff(a))

#define slk\_attr\_on(a,v) ((v) ? ERR : slk\_attron(a))

#define wattr\_set(win,a,p,opts) ((win)->\_attrs = (((a) & ~A\_COLOR) | (attr\_t)COLOR\_PAIR(p)), OK)

#define vw\_printw vwprintw

#define vw\_scanw vwscanw

#define vsscanf(a,b,c) \_nc\_vsscanf(a,b,c)

#define is\_cleared(win) ((win) ? (win)->\_clear : FALSE)

#define is\_idcok(win) ((win) ? (win)->\_idcok : FALSE)

#define is\_idlok(win) ((win) ? (win)->\_idlok : FALSE)

#define is\_immedok(win) ((win) ? (win)->\_immed : FALSE)

#define is\_keypad(win) ((win) ? (win)->\_use\_keypad : FALSE)

#define is\_leaveok(win) ((win) ? (win)->\_leaveok : FALSE)

#define is\_nodelay(win) ((win) ? ((win)->\_delay == 0) : FALSE)

#define is\_notimeout(win) ((win) ? (win)->\_notimeout : FALSE)

#define is\_pad(win) ((win) ? ((win)->\_flags & \_ISPAD) != 0 : FALSE)

#define is\_scrollok(win) ((win) ? (win)->\_scroll : FALSE)

#define is\_subwin(win) ((win) ? ((win)->\_flags & \_SUBWIN) != 0 : FALSE)

#define is\_syncok(win) ((win) ? (win)->\_sync : FALSE)

#define wgetparent(win) ((win) ? (win)->\_parent : 0)

#define wgetscrreg(win,t,b) ((win) ? (\*(t) = (win)->\_regtop, \*(b) = (win)->\_regbottom, OK) : ERR)

#define KEY\_CODE\_YES 0400

#define KEY\_MIN 0401

#define KEY\_BREAK 0401

#define KEY\_SRESET 0530

#define KEY\_RESET 0531

#define KEY\_DOWN 0402

#define KEY\_UP 0403

#define KEY\_LEFT 0404

#define KEY\_RIGHT 0405

#define KEY\_HOME 0406

#define KEY\_BACKSPACE 0407

#define KEY\_F0 0410

#define KEY\_F(n) (KEY\_F0+(n))

#define KEY\_DL 0510

#define KEY\_IL 0511

#define KEY\_DC 0512

#define KEY\_IC 0513

#define KEY\_EIC 0514

#define KEY\_CLEAR 0515

#define KEY\_EOS 0516

#define KEY\_EOL 0517

#define KEY\_SF 0520

#define KEY\_SR 0521

#define KEY\_NPAGE 0522

#define KEY\_PPAGE 0523

#define KEY\_STAB 0524

#define KEY\_CTAB 0525

#define KEY\_CATAB 0526

#define KEY\_ENTER 0527

#define KEY\_PRINT 0532

#define KEY\_LL 0533

#define KEY\_A1 0534

#define KEY\_A3 0535

#define KEY\_B2 0536

#define KEY\_C1 0537

#define KEY\_C3 0540

#define KEY\_BTAB 0541

#define KEY\_BEG 0542

#define KEY\_CANCEL 0543

#define KEY\_CLOSE 0544

#define KEY\_COMMAND 0545

#define KEY\_COPY 0546

#define KEY\_CREATE 0547

#define KEY\_END 0550

#define KEY\_EXIT 0551

#define KEY\_FIND 0552

#define KEY\_HELP 0553

#define KEY\_MARK 0554

#define KEY\_MESSAGE 0555

#define KEY\_MOVE 0556

#define KEY\_NEXT 0557

#define KEY\_OPEN 0560

#define KEY\_OPTIONS 0561

#define KEY\_PREVIOUS 0562

#define KEY\_REDO 0563

#define KEY\_REFERENCE 0564

#define KEY\_REFRESH 0565

#define KEY\_REPLACE 0566

#define KEY\_RESTART 0567

#define KEY\_RESUME 0570

#define KEY\_SAVE 0571

#define KEY\_SBEG 0572

#define KEY\_SCANCEL 0573

#define KEY\_SCOMMAND 0574

#define KEY\_SCOPY 0575

#define KEY\_SCREATE 0576

#define KEY\_SDC 0577

#define KEY\_SDL 0600

#define KEY\_SELECT 0601

#define KEY\_SEND 0602

#define KEY\_SEOL 0603

#define KEY\_SEXIT 0604

#define KEY\_SFIND 0605

#define KEY\_SHELP 0606

#define KEY\_SHOME 0607

#define KEY\_SIC 0610

#define KEY\_SLEFT 0611

#define KEY\_SMESSAGE 0612

#define KEY\_SMOVE 0613

#define KEY\_SNEXT 0614

#define KEY\_SOPTIONS 0615

#define KEY\_SPREVIOUS 0616

#define KEY\_SPRINT 0617

#define KEY\_SREDO 0620

#define KEY\_SREPLACE 0621

#define KEY\_SRIGHT 0622

#define KEY\_SRSUME 0623

#define KEY\_SSAVE 0624

#define KEY\_SSUSPEND 0625

#define KEY\_SUNDO 0626

#define KEY\_SUSPEND 0627

#define KEY\_UNDO 0630

#define KEY\_MOUSE 0631

#define KEY\_RESIZE 0632

#define KEY\_EVENT 0633

#define KEY\_MAX 0777

#define \_XOPEN\_CURSES 1

#define NCURSES\_WACS(c) (&\_nc\_wacs[(unsigned char)c])

#define WACS\_BSSB NCURSES\_WACS('l')

#define WACS\_SSBB NCURSES\_WACS('m')

#define WACS\_BBSS NCURSES\_WACS('k')

#define WACS\_SBBS NCURSES\_WACS('j')

#define WACS\_SBSS NCURSES\_WACS('u')

#define WACS\_SSSB NCURSES\_WACS('t')

#define WACS\_SSBS NCURSES\_WACS('v')

#define WACS\_BSSS NCURSES\_WACS('w')

#define WACS\_BSBS NCURSES\_WACS('q')

#define WACS\_SBSB NCURSES\_WACS('x')

#define WACS\_SSSS NCURSES\_WACS('n')

#define WACS\_ULCORNER WACS\_BSSB

#define WACS\_LLCORNER WACS\_SSBB

#define WACS\_URCORNER WACS\_BBSS

#define WACS\_LRCORNER WACS\_SBBS

#define WACS\_RTEE WACS\_SBSS

#define WACS\_LTEE WACS\_SSSB

#define WACS\_BTEE WACS\_SSBS

#define WACS\_TTEE WACS\_BSSS

#define WACS\_HLINE WACS\_BSBS

#define WACS\_VLINE WACS\_SBSB

#define WACS\_PLUS WACS\_SSSS

#define WACS\_S1 NCURSES\_WACS('o')

#define WACS\_S9 NCURSES\_WACS('s')

#define WACS\_DIAMOND NCURSES\_WACS('`')

#define WACS\_CKBOARD NCURSES\_WACS('a')

#define WACS\_DEGREE NCURSES\_WACS('f')

#define WACS\_PLMINUS NCURSES\_WACS('g')

#define WACS\_BULLET NCURSES\_WACS('~')

#define WACS\_LARROW NCURSES\_WACS(',')

#define WACS\_RARROW NCURSES\_WACS('+')

#define WACS\_DARROW NCURSES\_WACS('.')

#define WACS\_UARROW NCURSES\_WACS('-')

#define WACS\_BOARD NCURSES\_WACS('h')

#define WACS\_LANTERN NCURSES\_WACS('i')

#define WACS\_BLOCK NCURSES\_WACS('0')

#define WACS\_S3 NCURSES\_WACS('p')

#define WACS\_S7 NCURSES\_WACS('r')

#define WACS\_LEQUAL NCURSES\_WACS('y')

#define WACS\_GEQUAL NCURSES\_WACS('z')

#define WACS\_PI NCURSES\_WACS('{')

#define WACS\_NEQUAL NCURSES\_WACS('|')

#define WACS\_STERLING NCURSES\_WACS('}')

#define WACS\_BDDB NCURSES\_WACS('C')

#define WACS\_DDBB NCURSES\_WACS('D')

#define WACS\_BBDD NCURSES\_WACS('B')

#define WACS\_DBBD NCURSES\_WACS('A')

#define WACS\_DBDD NCURSES\_WACS('G')

#define WACS\_DDDB NCURSES\_WACS('F')

#define WACS\_DDBD NCURSES\_WACS('H')

#define WACS\_BDDD NCURSES\_WACS('I')

#define WACS\_BDBD NCURSES\_WACS('R')

#define WACS\_DBDB NCURSES\_WACS('Y')

#define WACS\_DDDD NCURSES\_WACS('E')

#define WACS\_D\_ULCORNER WACS\_BDDB

#define WACS\_D\_LLCORNER WACS\_DDBB

#define WACS\_D\_URCORNER WACS\_BBDD

#define WACS\_D\_LRCORNER WACS\_DBBD

#define WACS\_D\_RTEE WACS\_DBDD

#define WACS\_D\_LTEE WACS\_DDDB

#define WACS\_D\_BTEE WACS\_DDBD

#define WACS\_D\_TTEE WACS\_BDDD

#define WACS\_D\_HLINE WACS\_BDBD

#define WACS\_D\_VLINE WACS\_DBDB

#define WACS\_D\_PLUS WACS\_DDDD

#define WACS\_BTTB NCURSES\_WACS('L')

#define WACS\_TTBB NCURSES\_WACS('M')

#define WACS\_BBTT NCURSES\_WACS('K')

#define WACS\_TBBT NCURSES\_WACS('J')

#define WACS\_TBTT NCURSES\_WACS('U')

#define WACS\_TTTB NCURSES\_WACS('T')

#define WACS\_TTBT NCURSES\_WACS('V')

#define WACS\_BTTT NCURSES\_WACS('W')

#define WACS\_BTBT NCURSES\_WACS('Q')

#define WACS\_TBTB NCURSES\_WACS('X')

#define WACS\_TTTT NCURSES\_WACS('N')

#define WACS\_T\_ULCORNER WACS\_BTTB

#define WACS\_T\_LLCORNER WACS\_TTBB

#define WACS\_T\_URCORNER WACS\_BBTT

#define WACS\_T\_LRCORNER WACS\_TBBT

#define WACS\_T\_RTEE WACS\_TBTT

#define WACS\_T\_LTEE WACS\_TTTB

#define WACS\_T\_BTEE WACS\_TTBT

#define WACS\_T\_TTEE WACS\_BTTT

#define WACS\_T\_HLINE WACS\_BTBT

#define WACS\_T\_VLINE WACS\_TBTB

#define WACS\_T\_PLUS WACS\_TTTT

#define add\_wch(c) wadd\_wch(stdscr,c)

#define add\_wchnstr(str,n) wadd\_wchnstr(stdscr,str,n)

#define add\_wchstr(str) wadd\_wchstr(stdscr,str)

#define addnwstr(wstr,n) waddnwstr(stdscr,wstr,n)

#define addwstr(wstr) waddwstr(stdscr,wstr)

#define bkgrnd(c) wbkgrnd(stdscr,c)

#define bkgrndset(c) wbkgrndset(stdscr,c)

#define border\_set(l,r,t,b,tl,tr,bl,br) wborder\_set(stdscr,l,r,t,b,tl,tr,bl,br)

#define box\_set(w,v,h) wborder\_set(w,v,v,h,h,0,0,0,0)

#define echo\_wchar(c) wecho\_wchar(stdscr,c)

#define get\_wch(c) wget\_wch(stdscr,c)

#define get\_wstr(t) wget\_wstr(stdscr,t)

#define getbkgrnd(wch) wgetbkgrnd(stdscr,wch)

#define getn\_wstr(t,n) wgetn\_wstr(stdscr,t,n)

#define hline\_set(c,n) whline\_set(stdscr,c,n)

#define in\_wch(c) win\_wch(stdscr,c)

#define in\_wchnstr(c,n) win\_wchnstr(stdscr,c,n)

#define in\_wchstr(c) win\_wchstr(stdscr,c)

#define innwstr(c,n) winnwstr(stdscr,c,n)

#define ins\_nwstr(t,n) wins\_nwstr(stdscr,t,n)

#define ins\_wch(c) wins\_wch(stdscr,c)

#define ins\_wstr(t) wins\_wstr(stdscr,t)

#define inwstr(c) winwstr(stdscr,c)

#define vline\_set(c,n) wvline\_set(stdscr,c,n)

#define wadd\_wchstr(win,str) wadd\_wchnstr(win,str,-1)

#define waddwstr(win,wstr) waddnwstr(win,wstr,-1)

#define wget\_wstr(w,t) wgetn\_wstr(w,t,-1)

#define win\_wchstr(w,c) win\_wchnstr(w,c,-1)

#define wins\_wstr(w,t) wins\_nwstr(w,t,-1)

#define wgetbkgrnd(win,wch) (\*wch = win->\_bkgrnd, OK)

#define mvadd\_wch(y,x,c) mvwadd\_wch(stdscr,y,x,c)

#define mvadd\_wchnstr(y,x,s,n) mvwadd\_wchnstr(stdscr,y,x,s,n)

#define mvadd\_wchstr(y,x,s) mvwadd\_wchstr(stdscr,y,x,s)

#define mvaddnwstr(y,x,wstr,n) mvwaddnwstr(stdscr,y,x,wstr,n)

#define mvaddwstr(y,x,wstr) mvwaddwstr(stdscr,y,x,wstr)

#define mvget\_wch(y,x,c) mvwget\_wch(stdscr,y,x,c)

#define mvget\_wstr(y,x,t) mvwget\_wstr(stdscr,y,x,t)

#define mvgetn\_wstr(y,x,t,n) mvwgetn\_wstr(stdscr,y,x,t,n)

#define mvhline\_set(y,x,c,n) mvwhline\_set(stdscr,y,x,c,n)

#define mvin\_wch(y,x,c) mvwin\_wch(stdscr,y,x,c)

#define mvin\_wchnstr(y,x,c,n) mvwin\_wchnstr(stdscr,y,x,c,n)

#define mvin\_wchstr(y,x,c) mvwin\_wchstr(stdscr,y,x,c)

#define mvinnwstr(y,x,c,n) mvwinnwstr(stdscr,y,x,c,n)

#define mvins\_nwstr(y,x,t,n) mvwins\_nwstr(stdscr,y,x,t,n)

#define mvins\_wch(y,x,c) mvwins\_wch(stdscr,y,x,c)

#define mvins\_wstr(y,x,t) mvwins\_wstr(stdscr,y,x,t)

#define mvinwstr(y,x,c) mvwinwstr(stdscr,y,x,c)

#define mvvline\_set(y,x,c,n) mvwvline\_set(stdscr,y,x,c,n)

#define mvwadd\_wch(win,y,x,c) (wmove(win,y,x) == ERR ? ERR : wadd\_wch(win,c))

#define mvwadd\_wchnstr(win,y,x,s,n) (wmove(win,y,x) == ERR ? ERR : wadd\_wchnstr(win,s,n))

#define mvwadd\_wchstr(win,y,x,s) (wmove(win,y,x) == ERR ? ERR : wadd\_wchstr(win,s))

#define mvwaddnwstr(win,y,x,wstr,n) (wmove(win,y,x) == ERR ? ERR : waddnwstr(win,wstr,n))

#define mvwaddwstr(win,y,x,wstr) (wmove(win,y,x) == ERR ? ERR : waddwstr(win,wstr))

#define mvwget\_wch(win,y,x,c) (wmove(win,y,x) == ERR ? ERR : wget\_wch(win,c))

#define mvwget\_wstr(win,y,x,t) (wmove(win,y,x) == ERR ? ERR : wget\_wstr(win,t))

#define mvwgetn\_wstr(win,y,x,t,n) (wmove(win,y,x) == ERR ? ERR : wgetn\_wstr(win,t,n))

#define mvwhline\_set(win,y,x,c,n) (wmove(win,y,x) == ERR ? ERR : whline\_set(win,c,n))

#define mvwin\_wch(win,y,x,c) (wmove(win,y,x) == ERR ? ERR : win\_wch(win,c))

#define mvwin\_wchnstr(win,y,x,c,n) (wmove(win,y,x) == ERR ? ERR : win\_wchnstr(win,c,n))

#define mvwin\_wchstr(win,y,x,c) (wmove(win,y,x) == ERR ? ERR : win\_wchstr(win,c))

#define mvwinnwstr(win,y,x,c,n) (wmove(win,y,x) == ERR ? ERR : winnwstr(win,c,n))

#define mvwins\_nwstr(win,y,x,t,n) (wmove(win,y,x) == ERR ? ERR : wins\_nwstr(win,t,n))

#define mvwins\_wch(win,y,x,c) (wmove(win,y,x) == ERR ? ERR : wins\_wch(win,c))

#define mvwins\_wstr(win,y,x,t) (wmove(win,y,x) == ERR ? ERR : wins\_wstr(win,t))

#define mvwinwstr(win,y,x,c) (wmove(win,y,x) == ERR ? ERR : winwstr(win,c))

#define mvwvline\_set(win,y,x,c,n) (wmove(win,y,x) == ERR ? ERR : wvline\_set(win,c,n))

#define NCURSES\_MOUSE\_MASK(b,m) ((m) << (((b) - 1) \* 6))

#define NCURSES\_BUTTON\_RELEASED 001L

#define NCURSES\_BUTTON\_PRESSED 002L

#define NCURSES\_BUTTON\_CLICKED 004L

#define NCURSES\_DOUBLE\_CLICKED 010L

#define NCURSES\_TRIPLE\_CLICKED 020L

#define NCURSES\_RESERVED\_EVENT 040L

#define BUTTON1\_RELEASED NCURSES\_MOUSE\_MASK(1, NCURSES\_BUTTON\_RELEASED)

#define BUTTON1\_PRESSED NCURSES\_MOUSE\_MASK(1, NCURSES\_BUTTON\_PRESSED)

#define BUTTON1\_CLICKED NCURSES\_MOUSE\_MASK(1, NCURSES\_BUTTON\_CLICKED)

#define BUTTON1\_DOUBLE\_CLICKED NCURSES\_MOUSE\_MASK(1, NCURSES\_DOUBLE\_CLICKED)

#define BUTTON1\_TRIPLE\_CLICKED NCURSES\_MOUSE\_MASK(1, NCURSES\_TRIPLE\_CLICKED)

#define BUTTON2\_RELEASED NCURSES\_MOUSE\_MASK(2, NCURSES\_BUTTON\_RELEASED)

#define BUTTON2\_PRESSED NCURSES\_MOUSE\_MASK(2, NCURSES\_BUTTON\_PRESSED)

#define BUTTON2\_CLICKED NCURSES\_MOUSE\_MASK(2, NCURSES\_BUTTON\_CLICKED)

#define BUTTON2\_DOUBLE\_CLICKED NCURSES\_MOUSE\_MASK(2, NCURSES\_DOUBLE\_CLICKED)

#define BUTTON2\_TRIPLE\_CLICKED NCURSES\_MOUSE\_MASK(2, NCURSES\_TRIPLE\_CLICKED)

#define BUTTON3\_RELEASED NCURSES\_MOUSE\_MASK(3, NCURSES\_BUTTON\_RELEASED)

#define BUTTON3\_PRESSED NCURSES\_MOUSE\_MASK(3, NCURSES\_BUTTON\_PRESSED)

#define BUTTON3\_CLICKED NCURSES\_MOUSE\_MASK(3, NCURSES\_BUTTON\_CLICKED)

#define BUTTON3\_DOUBLE\_CLICKED NCURSES\_MOUSE\_MASK(3, NCURSES\_DOUBLE\_CLICKED)

#define BUTTON3\_TRIPLE\_CLICKED NCURSES\_MOUSE\_MASK(3, NCURSES\_TRIPLE\_CLICKED)

#define BUTTON4\_RELEASED NCURSES\_MOUSE\_MASK(4, NCURSES\_BUTTON\_RELEASED)

#define BUTTON4\_PRESSED NCURSES\_MOUSE\_MASK(4, NCURSES\_BUTTON\_PRESSED)

#define BUTTON4\_CLICKED NCURSES\_MOUSE\_MASK(4, NCURSES\_BUTTON\_CLICKED)

#define BUTTON4\_DOUBLE\_CLICKED NCURSES\_MOUSE\_MASK(4, NCURSES\_DOUBLE\_CLICKED)

#define BUTTON4\_TRIPLE\_CLICKED NCURSES\_MOUSE\_MASK(4, NCURSES\_TRIPLE\_CLICKED)

#define BUTTON5\_RELEASED NCURSES\_MOUSE\_MASK(5, NCURSES\_BUTTON\_RELEASED)

#define BUTTON5\_PRESSED NCURSES\_MOUSE\_MASK(5, NCURSES\_BUTTON\_PRESSED)

#define BUTTON5\_CLICKED NCURSES\_MOUSE\_MASK(5, NCURSES\_BUTTON\_CLICKED)

#define BUTTON5\_DOUBLE\_CLICKED NCURSES\_MOUSE\_MASK(5, NCURSES\_DOUBLE\_CLICKED)

#define BUTTON5\_TRIPLE\_CLICKED NCURSES\_MOUSE\_MASK(5, NCURSES\_TRIPLE\_CLICKED)

#define BUTTON\_CTRL NCURSES\_MOUSE\_MASK(6, 0001L)

#define BUTTON\_SHIFT NCURSES\_MOUSE\_MASK(6, 0002L)

#define BUTTON\_ALT NCURSES\_MOUSE\_MASK(6, 0004L)

#define REPORT\_MOUSE\_POSITION NCURSES\_MOUSE\_MASK(6, 0010L)

#define BUTTON1\_RESERVED\_EVENT NCURSES\_MOUSE\_MASK(1, NCURSES\_RESERVED\_EVENT)

#define BUTTON2\_RESERVED\_EVENT NCURSES\_MOUSE\_MASK(2, NCURSES\_RESERVED\_EVENT)

#define BUTTON3\_RESERVED\_EVENT NCURSES\_MOUSE\_MASK(3, NCURSES\_RESERVED\_EVENT)

#define BUTTON4\_RESERVED\_EVENT NCURSES\_MOUSE\_MASK(4, NCURSES\_RESERVED\_EVENT)

#define ALL\_MOUSE\_EVENTS (REPORT\_MOUSE\_POSITION - 1)

#define BUTTON\_RELEASE(e, x) ((e) & NCURSES\_MOUSE\_MASK(x, 001))

#define BUTTON\_PRESS(e, x) ((e) & NCURSES\_MOUSE\_MASK(x, 002))

#define BUTTON\_CLICK(e, x) ((e) & NCURSES\_MOUSE\_MASK(x, 004))

#define BUTTON\_DOUBLE\_CLICK(e, x) ((e) & NCURSES\_MOUSE\_MASK(x, 010))

#define BUTTON\_TRIPLE\_CLICK(e, x) ((e) & NCURSES\_MOUSE\_MASK(x, 020))

#define BUTTON\_RESERVED\_EVENT(e, x) ((e) & NCURSES\_MOUSE\_MASK(x, 040))

#define mouse\_trafo(y,x,to\_screen) wmouse\_trafo(stdscr,y,x,to\_screen)

#define \_tracech\_t \_tracecchar\_t

#define \_tracech\_t2 \_tracecchar\_t2

#define TRACE\_DISABLE 0x0000

#define TRACE\_TIMES 0x0001

#define TRACE\_TPUTS 0x0002

#define TRACE\_UPDATE 0x0004

#define TRACE\_MOVE 0x0008

#define TRACE\_CHARPUT 0x0010

#define TRACE\_ORDINARY 0x001F

#define TRACE\_CALLS 0x0020

#define TRACE\_VIRTPUT 0x0040

#define TRACE\_IEVENT 0x0080

#define TRACE\_BITS 0x0100

#define TRACE\_ICALLS 0x0200

#define TRACE\_CCALLS 0x0400

#define TRACE\_DATABASE 0x0800

#define TRACE\_ATTRS 0x1000

#define TRACE\_SHIFT 13

#define TRACE\_MAXIMUM ((1 << TRACE\_SHIFT) - 1)

#define OPTIMIZE\_MVCUR 0x01

#define OPTIMIZE\_HASHMAP 0x02

#define OPTIMIZE\_SCROLL 0x04

#define OPTIMIZE\_ALL 0xff

typedef unsigned long int chtype;

typedef chtype attr\_t;

struct pdat {

short \_pad\_y;

short \_pad\_x;

short \_pad\_top;

short \_pad\_left;

short \_pad\_bottom;

short \_pad\_right;

};

typedef struct screen SCREEN;

typedef struct \_win\_st WINDOW;

typedef unsigned long int mmask\_t;

typedef unsigned char bool;

typedef unsigned char NCURSES\_BOOL;

typedef int (\*NCURSES\_OUTC) (int);

typedef int (\*NCURSES\_WINDOW\_CB) (WINDOW \*, void \*);

typedef int (\*NCURSES\_SCREEN\_CB) (SCREEN \*, void \*);

struct \_win\_st {

short \_cury; /\* current cursor position \*/

short \_curx;

short \_maxy; /\* maximums of x and y, NOT window size \*/

short \_maxx;

short \_begy; /\* screen coords of upper-left-hand corner \*/

short \_begx;

short \_flags; /\* window state flags \*/

attr\_t \_attrs; /\* current attribute for non-space character \*/

chtype \_bkgd; /\* current background char/attribute pair \*/

bool \_notimeout; /\* no time out on function-key entry? \*/

bool \_clear; /\* consider all data in the window invalid? \*/

bool \_leaveok; /\* OK to not reset cursor on exit? \*/

bool \_scroll; /\* OK to scroll this window? \*/

bool \_idlok; /\* OK to use insert/delete line? \*/

bool \_idcok; /\* OK to use insert/delete char? \*/

bool \_immed; /\* window in immed mode? (not yet used) \*/

bool \_sync; /\* window in sync mode? \*/

bool \_use\_keypad; /\* process function keys into KEY\_ symbols? \*/

int \_delay; /\* 0 = nodelay, <0 = blocking, >0 = delay \*/

struct ldat \*\_line; /\* the actual line data \*/

short \_regtop; /\* top line of scrolling region \*/

short \_regbottom; /\* bottom line of scrolling region \*/

int \_parx; /\* x coordinate of this window in parent \*/

int \_pary; /\* y coordinate of this window in parent \*/

WINDOW \*\_parent; /\* pointer to parent if a sub-window \*/

struct pdat \_pad;

short \_yoffset; /\* real begy is \_begy + \_yoffset \*/

cchar\_t \_bkgrnd; /\* current background char/attribute pair \*/

};

extern int COLORS;

extern int COLOR\_PAIRS;

extern int COLS;

extern int LINES;

extern chtype acs\_map[];

extern int add\_wch(cchar\_t \*);

extern int add\_wchnstr(cchar\_t \*, int);

extern int add\_wchstr(cchar\_t \*);

extern int addch(const chtype);

extern int addchnstr(const chtype \*, int);

extern int addchstr(const chtype \*);

extern int addnstr(const char \*, int);

extern int addnwstr(wchar\_t \*, int);

extern int addstr(const char \*);

extern int addwstr(wchar\_t \*);

extern int assume\_default\_colors(int, int);

extern int attr\_get(attr\_t \*, short \*, void \*);

extern int attr\_off(attr\_t, void \*);

extern int attr\_on(attr\_t, void \*);

extern int attr\_set(attr\_t, short, void \*);

extern int attroff(int);

extern int attron(int);

extern int attrset(int);

extern int baudrate(void);

extern int beep(void);

extern int bkgd(chtype);

extern void bkgdset(chtype);

extern int bkgrnd(cchar\_t \*);

extern void bkgrndset(cchar\_t \*);

extern int border(chtype, chtype, chtype, chtype, chtype, chtype, chtype,

chtype);

extern int border\_set(cchar\_t \*, cchar\_t \*, cchar\_t \*, cchar\_t \*,

cchar\_t \*, cchar\_t \*, cchar\_t \*, cchar\_t \*);

extern int box(WINDOW \*, chtype, chtype);

extern int box\_set(WINDOW \*, cchar\_t \*, cchar\_t \*);

extern unsigned char can\_change\_color(void);

extern int cbreak(void);

extern int chgat(int, attr\_t, short, const void \*);

extern int clear(void);

extern int clearok(WINDOW \*, unsigned char);

extern int clrtobot(void);

extern int clrtoeol(void);

extern int color\_content(short, short \*, short \*, short \*);

extern int color\_set(short, void \*);

extern int copywin(const WINDOW \*, WINDOW \*, int, int, int, int, int, int,

int);

extern int curs\_set(int);

extern WINDOW \*curscr;

extern const char \*curses\_version(void);

extern int def\_prog\_mode(void);

extern int def\_shell\_mode(void);

extern int define\_key(const char \*, int);

extern int delay\_output(int);

extern int delch(void);

extern int deleteln(void);

extern void delscreen(SCREEN \*);

extern int delwin(WINDOW \*);

extern WINDOW \*derwin(WINDOW \*, int, int, int, int);

extern int doupdate(void);

extern WINDOW \*dupwin(WINDOW \*);

extern int echo(void);

extern int echo\_wchar(cchar\_t \*);

extern int echochar(const chtype);

extern int endwin(void);

extern int erase(void);

extern char erasechar(void);

extern int erasewchar(wchar\_t \*);

extern void filter(void);

extern int flash(void);

extern int flushinp(void);

extern int get\_wch(wint\_t \*);

extern int get\_wstr(wint\_t \*);

extern chtype getbkgd(WINDOW \*);

extern int getbkgrnd(cchar\_t \*);

extern int getcchar(cchar\_t \*, wchar\_t \*, attr\_t \*, short \*, void \*);

extern int getch(void);

extern int getmouse(MEVENT \*);

extern int getn\_wstr(wint\_t \*, int);

extern int getnstr(char \*, int);

extern int getstr(char \*);

extern WINDOW \*getwin(FILE \*);

extern int halfdelay(int);

extern unsigned char has\_colors(void);

extern unsigned char has\_ic(void);

extern unsigned char has\_il(void);

extern int has\_key(void);

extern bool has\_mouse(void);

extern int hline(chtype, int);

extern int hline\_set(cchar\_t \*, int);

extern void idcok(WINDOW \*, unsigned char);

extern int idlok(WINDOW \*, unsigned char);

extern void immedok(WINDOW \*, unsigned char);

extern int in\_wch(cchar\_t \*);

extern int in\_wchnstr(cchar\_t \*, int);

extern int in\_wchstr(cchar\_t \*);

extern chtype inch(void);

extern int inchnstr(chtype \*, int);

extern int inchstr(chtype \*);

extern int init\_color(short, short, short, short);

extern int init\_pair(short, short, short);

extern WINDOW \*initscr(void);

extern int innstr(char \*, int);

extern int innwstr(wchar\_t \*, int);

extern int ins\_nwstr(wchar\_t \*, int);

extern int ins\_wch(cchar\_t \*);

extern int ins\_wstr(wchar\_t \*);

extern int insch(chtype);

extern int insdelln(int);

extern int insertln(void);

extern int insnstr(const char \*, int);

extern int insstr(const char \*);

extern int instr(char \*);

extern int intrflush(WINDOW \*, unsigned char);

extern int inwstr(wchar\_t \*);

extern unsigned char is\_linetouched(WINDOW \*, int);

extern unsigned char is\_wintouched(WINDOW \*);

extern unsigned char isendwin(void);

extern char \*key\_name(wchar\_t);

extern char \*keybound(int, int);

extern char \*keyname(int);

extern int keyok(int, unsigned char);

extern int keypad(WINDOW \*, unsigned char);

extern char killchar(void);

extern int killwchar(wchar\_t \*);

extern int leaveok(WINDOW \*, unsigned char);

extern char \*longname(void);

extern int mcprint(void);

extern int meta(WINDOW \*, unsigned char);

extern bool mouse\_trafo(int \*, int \*, bool);

extern int mouseinterval(int);

extern mmask\_t mousemask(mmask\_t, mmask\_t \*);

extern int move(int, int);

extern int mvadd\_wch(int, int, cchar\_t \*);

extern int mvadd\_wchnstr(int, int, cchar\_t \*, int);

extern int mvadd\_wchstr(int, int, cchar\_t \*);

extern int mvaddch(const chtype, const chtype, const chtype);

extern int mvaddchnstr(int, int, const chtype \*, int);

extern int mvaddchstr(int, int, const chtype \*);

extern int mvaddnstr(int, int, const char \*, int);

extern int mvaddnwstr(int, int, wchar\_t \*, int);

extern int mvaddstr(int, int, const char \*);

extern int mvaddwstr(int, int, wchar\_t \*);

extern int mvchgat(int, int, int, attr\_t, short, const void \*);

extern int mvcur(int, int, int, int);

extern int mvdelch(int, int);

extern int mvderwin(WINDOW \*, int, int);

extern int mvget\_wch(int, int, wint\_t \*);

extern int mvget\_wstr(int, int, wint\_t \*);

extern int mvgetch(int, int);

extern int mvgetn\_wstr(int, int, wint\_t \*, int);

extern int mvgetnstr(int, int, char \*, int);

extern int mvgetstr(int, int, char \*);

extern int mvhline(int, int, chtype, int);

extern int mvhline\_set(int, int, cchar\_t \*, int);

extern int mvin\_wch(int, int, cchar\_t \*);

extern int mvin\_wchnstr(int, int, cchar\_t \*, int);

extern int mvin\_wchstr(int, int, cchar\_t \*);

extern chtype mvinch(int, int);

extern int mvinchnstr(int, int, chtype \*, int);

extern int mvinchstr(int, int, chtype \*);

extern int mvinnstr(int, int, char \*, int);

extern int mvinnwstr(int, int, wchar\_t \*, int);

extern int mvins\_nwstr(int, int, wchar\_t \*, int);

extern int mvins\_wch(int, int, cchar\_t \*);

extern int mvins\_wstr(int, int, wchar\_t \*);

extern int mvinsch(int, int, chtype);

extern int mvinsnstr(int, int, const char \*, int);

extern int mvinsstr(int, int, const char \*);

extern int mvinstr(int, int, char \*);

extern int mvinwstr(int, int, wchar\_t \*);

extern int mvprintw(int, int, const char \*, ...);

extern int mvscanw(int, int, char \*, ...);

extern int mvvline(int, int, chtype, int);

extern int mvvline\_set(int, int, cchar\_t \*, int);

extern int mvwadd\_wch(WINDOW \*, int, int, cchar\_t \*);

extern int mvwadd\_wchnstr(WINDOW \*, int, int, cchar\_t \*, int);

extern int mvwadd\_wchstr(WINDOW \*, int, int, cchar\_t \*);

extern int mvwaddch(const chtype, const chtype, const chtype,

const chtype);

extern int mvwaddchnstr(WINDOW \*, int, int, const chtype \*, int);

extern int mvwaddchstr(WINDOW \*, int, int, const chtype \*);

extern int mvwaddnstr(WINDOW \*, int, int, const char \*, int);

extern int mvwaddnwstr(WINDOW \*, int, int, wchar\_t \*, int);

extern int mvwaddstr(WINDOW \*, int, int, const char \*);

extern int mvwaddwstr(WINDOW \*, int, int, wchar\_t \*);

extern int mvwchgat(WINDOW \*, int, int, int, attr\_t, short, const void \*);

extern int mvwdelch(WINDOW \*, int, int);

extern int mvwget\_wch(WINDOW \*, int, int, wint\_t \*);

extern int mvwget\_wstr(WINDOW \*, int, int, wint\_t \*);

extern int mvwgetch(WINDOW \*, int, int);

extern int mvwgetn\_wstr(WINDOW \*, int, int, wint\_t \*, int);

extern int mvwgetnstr(WINDOW \*, int, int, char \*, int);

extern int mvwgetstr(WINDOW \*, int, int, char \*);

extern int mvwhline(WINDOW \*, int, int, chtype, int);

extern int mvwhline\_set(WINDOW \*, int, int, cchar\_t \*, int);

extern int mvwin(WINDOW \*, int, int);

extern int mvwin\_wch(WINDOW \*, int, int, cchar\_t \*);

extern int mvwin\_wchnstr(WINDOW \*, int, int, cchar\_t \*, int);

extern int mvwin\_wchstr(WINDOW \*, int, int, cchar\_t \*);

extern chtype mvwinch(WINDOW \*, int, int);

extern int mvwinchnstr(WINDOW \*, int, int, chtype \*, int);

extern int mvwinchstr(WINDOW \*, int, int, chtype \*);

extern int mvwinnstr(WINDOW \*, int, int, char \*, int);

extern int mvwinnwstr(WINDOW \*, int, int, wchar\_t \*, int);

extern int mvwins\_nwstr(WINDOW \*, int, int, wchar\_t \*, int);

extern int mvwins\_wch(WINDOW \*, int, int, cchar\_t \*);

extern int mvwins\_wstr(WINDOW \*, int, int, wchar\_t \*);

extern int mvwinsch(WINDOW \*, int, int, chtype);

extern int mvwinsnstr(WINDOW \*, int, int, const char \*, int);

extern int mvwinsstr(WINDOW \*, int, int, const char \*);

extern int mvwinstr(WINDOW \*, int, int, char \*);

extern int mvwinwstr(WINDOW \*, int, int, wchar\_t \*);

extern int mvwprintw(WINDOW \*, int, int, const char \*, ...);

extern int mvwscanw(WINDOW \*, int, int, char \*, ...);

extern int mvwvline(WINDOW \*, int, int, chtype, int);

extern int mvwvline\_set(WINDOW \*, int, int, cchar\_t \*, int);

extern int napms(int);

extern WINDOW \*newpad(int, int);

extern WINDOW \*newscr;

extern SCREEN \*newterm(char \*, FILE \*, FILE \*);

extern WINDOW \*newwin(int, int, int, int);

extern int nl(void);

extern int nocbreak(void);

extern int nodelay(WINDOW \*, unsigned char);

extern int noecho(void);

extern int nonl(void);

extern void noqiflush(void);

extern int noraw(void);

extern int notimeout(WINDOW \*, unsigned char);

extern int overlay(const WINDOW \*, WINDOW \*);

extern int overwrite(const WINDOW \*, WINDOW \*);

extern int pair\_content(short, short \*, short \*);

extern int pecho\_wchar(WINDOW \*, cchar\_t \*);

extern int pechochar(const chtype, const chtype);

extern int pnoutrefresh(WINDOW \*, int, int, int, int, int, int);

extern int prefresh(WINDOW \*, int, int, int, int, int, int);

extern int printw(const char \*, ...);

extern int putwin(WINDOW \*, FILE \*);

extern void qiflush(void);

extern int raw(void);

extern int redrawwin(WINDOW \*);

extern int refresh(void);

extern int reset\_prog\_mode(void);

extern int reset\_shell\_mode(void);

extern int resetty(void);

extern int resizeterm(int, int);

extern int ripoffline(int, int (\*)(WINDOW \*, int));

extern int savetty(void);

extern int scanw(char \*, ...);

extern int scr\_dump(const char \*);

extern int scr\_init(const char \*);

extern int scr\_restore(const char \*);

extern int scr\_set(const char \*);

extern int scrl(int);

extern int scroll(WINDOW \*);

extern int scrollok(WINDOW \*, unsigned char);

extern SCREEN \*set\_term(SCREEN \*);

extern int setcchar(cchar\_t \*, wchar\_t \*, attr\_t, short, void \*);

extern int setscrreg(int, int);

extern attr\_t slk\_attr(void);

extern int slk\_attr\_off(const attr\_t, void \*);

extern int slk\_attr\_on(attr\_t, void \*);

extern int slk\_attr\_set(const attr\_t, short, void \*);

extern int slk\_attroff(const chtype);

extern int slk\_attron(const chtype);

extern int slk\_attrset(const chtype);

extern int slk\_clear(void);

extern int slk\_color(short);

extern int slk\_init(int);

extern char \*slk\_label(int);

extern int slk\_noutrefresh(void);

extern int slk\_refresh(void);

extern int slk\_restore(void);

extern int slk\_set(int, const char \*, int);

extern int slk\_touch(void);

extern int slk\_wset(int, const wchar\_t \*, int);

extern int standend(void);

extern int standout(void);

extern int start\_color(void);

extern WINDOW \*stdscr;

extern WINDOW \*subpad(WINDOW \*, int, int, int, int);

extern WINDOW \*subwin(WINDOW \*, int, int, int, int);

extern int syncok(WINDOW \*, unsigned char);

extern attr\_t term\_attrs(void);

extern chtype termattrs(void);

extern char \*termname(void);

extern void timeout(int);

extern int touchline(WINDOW \*, int, int);

extern int touchwin(WINDOW \*);

extern int typeahead(int);

extern char \*unctrl(chtype);

extern int unget\_wch(wchar\_t);

extern int ungetch(int);

extern int ungetmouse(MEVENT \*);

extern int untouchwin(WINDOW \*);

extern int use\_default\_colors(void);

extern void use\_env(unsigned char);

extern int use\_extended\_names(unsigned char);

extern int vid\_attr(attr\_t, short, void \*);

extern int vid\_puts(attr\_t, short, void \*, int);

extern int vidattr(chtype);

extern int vidputs(chtype, NCURSES\_OUTC);

extern int vline(chtype, int);

extern int vline\_set(cchar\_t \*, int);

extern int vw\_printw(WINDOW \*, const char \*, va\_list);

extern int vw\_scanw(WINDOW \*, char \*, va\_list);

extern int vwprintw(WINDOW \*, const char \*, va\_list);

extern int vwscanw(WINDOW \*, char \*, va\_list);

extern int wadd\_wch(WINDOW \*, cchar\_t \*);

extern int wadd\_wchnstr(WINDOW \*, cchar\_t \*, int);

extern int wadd\_wchstr(WINDOW \*, cchar\_t \*);

extern int waddch(WINDOW \*, chtype);

extern int waddchnstr(WINDOW \*, const chtype \*, int);

extern int waddchstr(WINDOW \*, const chtype \*);

extern int waddnstr(WINDOW \*, const char \*, int);

extern int waddnwstr(WINDOW \*, wchar\_t \*, int);

extern int waddstr(WINDOW \*, const char \*);

extern int waddwstr(WINDOW \*, wchar\_t \*);

extern int wattr\_get(WINDOW \*, attr\_t \*, short \*, void \*);

extern int wattr\_off(WINDOW \*, attr\_t, void \*);

extern int wattr\_on(WINDOW \*, attr\_t, void \*);

extern int wattr\_set(WINDOW \*, attr\_t, short, void \*);

extern int wattroff(WINDOW \*, int);

extern int wattron(WINDOW \*, int);

extern int wattrset(WINDOW \*, int);

extern int wbkgd(WINDOW \*, chtype);

extern void wbkgdset(WINDOW \*, chtype);

extern int wbkgrnd(WINDOW \*, cchar\_t \*);

extern void wbkgrndset(WINDOW \*, cchar\_t \*);

extern int wborder(WINDOW \*, chtype, chtype, chtype, chtype, chtype,

chtype, chtype, chtype);

extern int wborder\_set(WINDOW \*, cchar\_t \*, cchar\_t \*, cchar\_t \*,

cchar\_t \*, cchar\_t \*, cchar\_t \*, cchar\_t \*,

cchar\_t \*);

extern int wchgat(WINDOW \*, int, attr\_t, short, const void \*);

extern int wclear(WINDOW \*);

extern int wclrtobot(WINDOW \*);

extern int wclrtoeol(WINDOW \*);

extern int wcolor\_set(WINDOW \*, short, void \*);

extern void wcursyncup(WINDOW \*);

extern int wdelch(WINDOW \*);

extern int wdeleteln(WINDOW \*);

extern int wecho\_wchar(WINDOW \*, cchar\_t \*);

extern int wechochar(const chtype, const chtype);

extern int werase(WINDOW \*);

extern int wget\_wch(WINDOW \*, wint\_t \*);

extern int wget\_wstr(WINDOW \*, wint\_t \*);

extern int wgetbkgrnd(WINDOW \*, cchar\_t \*);

extern int wgetch(WINDOW \*);

extern int wgetn\_wstr(WINDOW \*, wint\_t \*, int);

extern int wgetnstr(WINDOW \*, char \*, int);

extern int wgetstr(WINDOW \*, char \*);

extern int whline(WINDOW \*, chtype, int);

extern int whline\_set(WINDOW \*, cchar\_t \*, int);

extern int win\_wch(WINDOW \*, cchar\_t \*);

extern int win\_wchnstr(WINDOW \*, cchar\_t \*, int);

extern int win\_wchstr(WINDOW \*, cchar\_t \*);

extern chtype winch(WINDOW \*);

extern int winchnstr(WINDOW \*, chtype \*, int);

extern int winchstr(WINDOW \*, chtype \*);

extern int winnstr(WINDOW \*, char \*, int);

extern int winnwstr(WINDOW \*, wchar\_t \*, int);

extern int wins\_nwstr(WINDOW \*, wchar\_t \*, int);

extern int wins\_wch(WINDOW \*, cchar\_t \*);

extern int wins\_wstr(WINDOW \*, wchar\_t \*);

extern int winsch(WINDOW \*, chtype);

extern int winsdelln(WINDOW \*, int);

extern int winsertln(WINDOW \*);

extern int winsnstr(WINDOW \*, const char \*, int);

extern int winsstr(WINDOW \*, const char \*);

extern int winstr(WINDOW \*, char \*);

extern int winwstr(WINDOW \*, wchar\_t \*);

extern unsigned char wmouse\_trafo(const WINDOW \*, int \*, int \*, bool);

extern int wmove(WINDOW \*, int, int);

extern int wnoutrefresh(WINDOW \*);

extern int wprintw(WINDOW \*, const char \*, ...);

extern int wredrawln(WINDOW \*, int, int);

extern int wrefresh(WINDOW \*);

extern int wresize(WINDOW \*, int, int);

extern int wscanw(WINDOW \*, char \*, ...);

extern int wscrl(WINDOW \*, int);

extern int wsetscrreg(WINDOW \*, int, int);

extern int wstandend(WINDOW \*);

extern int wstandout(WINDOW \*);

extern void wsyncdown(WINDOW \*);

extern void wsyncup(WINDOW \*);

extern void wtimeout(WINDOW \*, int);

extern int wtouchln(WINDOW \*, int, int, int);

extern wchar\_t \*wunctrl(cchar\_t \*);

extern int wvline(WINDOW \*, chtype, int);

extern int wvline\_set(WINDOW \*, cchar\_t \*, int);

### **15.9.2 ncursesw/ncurses\_dll.h**

#define NCURSES\_API

#define NCURSES\_IMPEXP

#define NCURSES\_STATIC

#define NCURSES\_WRAPPED\_VAR(type,name) extern type NCURSES\_PUBLIC\_VAR(name)(void)

#define NCURSES\_PUBLIC\_VAR(name) \_nc\_ ##name

### **15.9.3 ncursesw/term.h**

#define NCURSES\_TERM\_H\_incl 1

#define NCURSES\_VERSION "5.9"

#define NCURSES\_SBOOL char

#define NCURSES\_XNAMES 1

#define TERMIOS 1

#define TTY struct termios

#define TCSANOW TCSETA

#define TCSADRAIN TCSETAW

#define TCSAFLUSH TCSETAF

#define tcsetattr(fd, cmd, arg) ioctl(fd, cmd, arg)

#define tcgetattr(fd, arg) ioctl(fd, TCGETA, arg)

#define cfgetospeed(t) ((t)->c\_cflag & CBAUD)

#define TCOFLUSH 1

#define TCIOFLUSH 2

#define tcflush(fd, arg) ioctl(fd, TCFLSH, arg)

#define GET\_TTY(fd, buf) tcgetattr(fd, buf)

#define SET\_TTY(fd, buf) tcsetattr(fd, TCSADRAIN, buf)

#define NAMESIZE 256

#define CUR cur\_term->type.

#define auto\_left\_margin CUR Booleans[0]

#define auto\_right\_margin CUR Booleans[1]

#define no\_esc\_ctlc CUR Booleans[2]

#define ceol\_standout\_glitch CUR Booleans[3]

#define eat\_newline\_glitch CUR Booleans[4]

#define erase\_overstrike CUR Booleans[5]

#define generic\_type CUR Booleans[6]

#define hard\_copy CUR Booleans[7]

#define has\_meta\_key CUR Booleans[8]

#define has\_status\_line CUR Booleans[9]

#define insert\_null\_glitch CUR Booleans[10]

#define memory\_above CUR Booleans[11]

#define memory\_below CUR Booleans[12]

#define move\_insert\_mode CUR Booleans[13]

#define move\_standout\_mode CUR Booleans[14]

#define over\_strike CUR Booleans[15]

#define status\_line\_esc\_ok CUR Booleans[16]

#define dest\_tabs\_magic\_smso CUR Booleans[17]

#define tilde\_glitch CUR Booleans[18]

#define transparent\_underline CUR Booleans[19]

#define xon\_xoff CUR Booleans[20]

#define needs\_xon\_xoff CUR Booleans[21]

#define prtr\_silent CUR Booleans[22]

#define hard\_cursor CUR Booleans[23]

#define non\_rev\_rmcup CUR Booleans[24]

#define no\_pad\_char CUR Booleans[25]

#define non\_dest\_scroll\_region CUR Booleans[26]

#define can\_change CUR Booleans[27]

#define back\_color\_erase CUR Booleans[28]

#define hue\_lightness\_saturation CUR Booleans[29]

#define col\_addr\_glitch CUR Booleans[30]

#define cr\_cancels\_micro\_mode CUR Booleans[31]

#define has\_print\_wheel CUR Booleans[32]

#define row\_addr\_glitch CUR Booleans[33]

#define semi\_auto\_right\_margin CUR Booleans[34]

#define cpi\_changes\_res CUR Booleans[35]

#define lpi\_changes\_res CUR Booleans[36]

#define columns CUR Numbers[0]

#define init\_tabs CUR Numbers[1]

#define lines CUR Numbers[2]

#define lines\_of\_memory CUR Numbers[3]

#define magic\_cookie\_glitch CUR Numbers[4]

#define padding\_baud\_rate CUR Numbers[5]

#define virtual\_terminal CUR Numbers[6]

#define width\_status\_line CUR Numbers[7]

#define num\_labels CUR Numbers[8]

#define label\_height CUR Numbers[9]

#define label\_width CUR Numbers[10]

#define max\_attributes CUR Numbers[11]

#define maximum\_windows CUR Numbers[12]

#define max\_colors CUR Numbers[13]

#define max\_pairs CUR Numbers[14]

#define no\_color\_video CUR Numbers[15]

#define buffer\_capacity CUR Numbers[16]

#define dot\_vert\_spacing CUR Numbers[17]

#define dot\_horz\_spacing CUR Numbers[18]

#define max\_micro\_address CUR Numbers[19]

#define max\_micro\_jump CUR Numbers[20]

#define micro\_col\_size CUR Numbers[21]

#define micro\_line\_size CUR Numbers[22]

#define number\_of\_pins CUR Numbers[23]

#define output\_res\_char CUR Numbers[24]

#define output\_res\_line CUR Numbers[25]

#define output\_res\_horz\_inch CUR Numbers[26]

#define output\_res\_vert\_inch CUR Numbers[27]

#define print\_rate CUR Numbers[28]

#define wide\_char\_size CUR Numbers[29]

#define buttons CUR Numbers[30]

#define bit\_image\_entwining CUR Numbers[31]

#define bit\_image\_type CUR Numbers[32]

#define back\_tab CUR Strings[0]

#define bell CUR Strings[1]

#define carriage\_return CUR Strings[2]

#define change\_scroll\_region CUR Strings[3]

#define clear\_all\_tabs CUR Strings[4]

#define clear\_screen CUR Strings[5]

#define clr\_eol CUR Strings[6]

#define clr\_eos CUR Strings[7]

#define column\_address CUR Strings[8]

#define command\_character CUR Strings[9]

#define cursor\_address CUR Strings[10]

#define cursor\_down CUR Strings[11]

#define cursor\_home CUR Strings[12]

#define cursor\_invisible CUR Strings[13]

#define cursor\_left CUR Strings[14]

#define cursor\_mem\_address CUR Strings[15]

#define cursor\_normal CUR Strings[16]

#define cursor\_right CUR Strings[17]

#define cursor\_to\_ll CUR Strings[18]

#define cursor\_up CUR Strings[19]

#define cursor\_visible CUR Strings[20]

#define delete\_character CUR Strings[21]

#define delete\_line CUR Strings[22]

#define dis\_status\_line CUR Strings[23]

#define down\_half\_line CUR Strings[24]

#define enter\_alt\_charset\_mode CUR Strings[25]

#define enter\_blink\_mode CUR Strings[26]

#define enter\_bold\_mode CUR Strings[27]

#define enter\_ca\_mode CUR Strings[28]

#define enter\_delete\_mode CUR Strings[29]

#define enter\_dim\_mode CUR Strings[30]

#define enter\_insert\_mode CUR Strings[31]

#define enter\_secure\_mode CUR Strings[32]

#define enter\_protected\_mode CUR Strings[33]

#define enter\_reverse\_mode CUR Strings[34]

#define enter\_standout\_mode CUR Strings[35]

#define enter\_underline\_mode CUR Strings[36]

#define erase\_chars CUR Strings[37]

#define exit\_alt\_charset\_mode CUR Strings[38]

#define exit\_attribute\_mode CUR Strings[39]

#define exit\_ca\_mode CUR Strings[40]

#define exit\_delete\_mode CUR Strings[41]

#define exit\_insert\_mode CUR Strings[42]

#define exit\_standout\_mode CUR Strings[43]

#define exit\_underline\_mode CUR Strings[44]

#define flash\_screen CUR Strings[45]

#define form\_feed CUR Strings[46]

#define from\_status\_line CUR Strings[47]

#define init\_1string CUR Strings[48]

#define init\_2string CUR Strings[49]

#define init\_3string CUR Strings[50]

#define init\_file CUR Strings[51]

#define insert\_character CUR Strings[52]

#define insert\_line CUR Strings[53]

#define insert\_padding CUR Strings[54]

#define key\_backspace CUR Strings[55]

#define key\_catab CUR Strings[56]

#define key\_clear CUR Strings[57]

#define key\_ctab CUR Strings[58]

#define key\_dc CUR Strings[59]

#define key\_dl CUR Strings[60]

#define key\_down CUR Strings[61]

#define key\_eic CUR Strings[62]

#define key\_eol CUR Strings[63]

#define key\_eos CUR Strings[64]

#define key\_f0 CUR Strings[65]

#define key\_f1 CUR Strings[66]

#define key\_f10 CUR Strings[67]

#define key\_f2 CUR Strings[68]

#define key\_f3 CUR Strings[69]

#define key\_f4 CUR Strings[70]

#define key\_f5 CUR Strings[71]

#define key\_f6 CUR Strings[72]

#define key\_f7 CUR Strings[73]

#define key\_f8 CUR Strings[74]

#define key\_f9 CUR Strings[75]

#define key\_home CUR Strings[76]

#define key\_ic CUR Strings[77]

#define key\_il CUR Strings[78]

#define key\_left CUR Strings[79]

#define key\_ll CUR Strings[80]

#define key\_npage CUR Strings[81]

#define key\_ppage CUR Strings[82]

#define key\_right CUR Strings[83]

#define key\_sf CUR Strings[84]

#define key\_sr CUR Strings[85]

#define key\_stab CUR Strings[86]

#define key\_up CUR Strings[87]

#define keypad\_local CUR Strings[88]

#define keypad\_xmit CUR Strings[89]

#define lab\_f0 CUR Strings[90]

#define lab\_f1 CUR Strings[91]

#define lab\_f10 CUR Strings[92]

#define lab\_f2 CUR Strings[93]

#define lab\_f3 CUR Strings[94]

#define lab\_f4 CUR Strings[95]

#define lab\_f5 CUR Strings[96]

#define lab\_f6 CUR Strings[97]

#define lab\_f7 CUR Strings[98]

#define lab\_f8 CUR Strings[99]

#define lab\_f9 CUR Strings[100]

#define meta\_off CUR Strings[101]

#define meta\_on CUR Strings[102]

#define newline CUR Strings[103]

#define pad\_char CUR Strings[104]

#define parm\_dch CUR Strings[105]

#define parm\_delete\_line CUR Strings[106]

#define parm\_down\_cursor CUR Strings[107]

#define parm\_ich CUR Strings[108]

#define parm\_index CUR Strings[109]

#define parm\_insert\_line CUR Strings[110]

#define parm\_left\_cursor CUR Strings[111]

#define parm\_right\_cursor CUR Strings[112]

#define parm\_rindex CUR Strings[113]

#define parm\_up\_cursor CUR Strings[114]

#define pkey\_key CUR Strings[115]

#define pkey\_local CUR Strings[116]

#define pkey\_xmit CUR Strings[117]

#define print\_screen CUR Strings[118]

#define prtr\_off CUR Strings[119]

#define prtr\_on CUR Strings[120]

#define repeat\_char CUR Strings[121]

#define reset\_1string CUR Strings[122]

#define reset\_2string CUR Strings[123]

#define reset\_3string CUR Strings[124]

#define reset\_file CUR Strings[125]

#define restore\_cursor CUR Strings[126]

#define row\_address CUR Strings[127]

#define save\_cursor CUR Strings[128]

#define scroll\_forward CUR Strings[129]

#define scroll\_reverse CUR Strings[130]

#define set\_attributes CUR Strings[131]

#define set\_tab CUR Strings[132]

#define set\_window CUR Strings[133]

#define tab CUR Strings[134]

#define to\_status\_line CUR Strings[135]

#define underline\_char CUR Strings[136]

#define up\_half\_line CUR Strings[137]

#define init\_prog CUR Strings[138]

#define key\_a1 CUR Strings[139]

#define key\_a3 CUR Strings[140]

#define key\_b2 CUR Strings[141]

#define key\_c1 CUR Strings[142]

#define key\_c3 CUR Strings[143]

#define prtr\_non CUR Strings[144]

#define char\_padding CUR Strings[145]

#define acs\_chars CUR Strings[146]

#define plab\_norm CUR Strings[147]

#define key\_btab CUR Strings[148]

#define enter\_xon\_mode CUR Strings[149]

#define exit\_xon\_mode CUR Strings[150]

#define enter\_am\_mode CUR Strings[151]

#define exit\_am\_mode CUR Strings[152]

#define xon\_character CUR Strings[153]

#define xoff\_character CUR Strings[154]

#define ena\_acs CUR Strings[155]

#define label\_on CUR Strings[156]

#define label\_off CUR Strings[157]

#define key\_beg CUR Strings[158]

#define key\_cancel CUR Strings[159]

#define key\_close CUR Strings[160]

#define key\_command CUR Strings[161]

#define key\_copy CUR Strings[162]

#define key\_create CUR Strings[163]

#define key\_end CUR Strings[164]

#define key\_enter CUR Strings[165]

#define key\_exit CUR Strings[166]

#define key\_find CUR Strings[167]

#define key\_help CUR Strings[168]

#define key\_mark CUR Strings[169]

#define key\_message CUR Strings[170]

#define key\_move CUR Strings[171]

#define key\_next CUR Strings[172]

#define key\_open CUR Strings[173]

#define key\_options CUR Strings[174]

#define key\_previous CUR Strings[175]

#define key\_print CUR Strings[176]

#define key\_redo CUR Strings[177]

#define key\_reference CUR Strings[178]

#define key\_refresh CUR Strings[179]

#define key\_replace CUR Strings[180]

#define key\_restart CUR Strings[181]

#define key\_resume CUR Strings[182]

#define key\_save CUR Strings[183]

#define key\_suspend CUR Strings[184]

#define key\_undo CUR Strings[185]

#define key\_sbeg CUR Strings[186]

#define key\_scancel CUR Strings[187]

#define key\_scommand CUR Strings[188]

#define key\_scopy CUR Strings[189]

#define key\_screate CUR Strings[190]

#define key\_sdc CUR Strings[191]

#define key\_sdl CUR Strings[192]

#define key\_select CUR Strings[193]

#define key\_send CUR Strings[194]

#define key\_seol CUR Strings[195]

#define key\_sexit CUR Strings[196]

#define key\_sfind CUR Strings[197]

#define key\_shelp CUR Strings[198]

#define key\_shome CUR Strings[199]

#define key\_sic CUR Strings[200]

#define key\_sleft CUR Strings[201]

#define key\_smessage CUR Strings[202]

#define key\_smove CUR Strings[203]

#define key\_snext CUR Strings[204]

#define key\_soptions CUR Strings[205]

#define key\_sprevious CUR Strings[206]

#define key\_sprint CUR Strings[207]

#define key\_sredo CUR Strings[208]

#define key\_sreplace CUR Strings[209]

#define key\_sright CUR Strings[210]

#define key\_srsume CUR Strings[211]

#define key\_ssave CUR Strings[212]

#define key\_ssuspend CUR Strings[213]

#define key\_sundo CUR Strings[214]

#define req\_for\_input CUR Strings[215]

#define key\_f11 CUR Strings[216]

#define key\_f12 CUR Strings[217]

#define key\_f13 CUR Strings[218]

#define key\_f14 CUR Strings[219]

#define key\_f15 CUR Strings[220]

#define key\_f16 CUR Strings[221]

#define key\_f17 CUR Strings[222]

#define key\_f18 CUR Strings[223]

#define key\_f19 CUR Strings[224]

#define key\_f20 CUR Strings[225]

#define key\_f21 CUR Strings[226]

#define key\_f22 CUR Strings[227]

#define key\_f23 CUR Strings[228]

#define key\_f24 CUR Strings[229]

#define key\_f25 CUR Strings[230]

#define key\_f26 CUR Strings[231]

#define key\_f27 CUR Strings[232]

#define key\_f28 CUR Strings[233]

#define key\_f29 CUR Strings[234]

#define key\_f30 CUR Strings[235]

#define key\_f31 CUR Strings[236]

#define key\_f32 CUR Strings[237]

#define key\_f33 CUR Strings[238]

#define key\_f34 CUR Strings[239]

#define key\_f35 CUR Strings[240]

#define key\_f36 CUR Strings[241]

#define key\_f37 CUR Strings[242]

#define key\_f38 CUR Strings[243]

#define key\_f39 CUR Strings[244]

#define key\_f40 CUR Strings[245]

#define key\_f41 CUR Strings[246]

#define key\_f42 CUR Strings[247]

#define key\_f43 CUR Strings[248]

#define key\_f44 CUR Strings[249]

#define key\_f45 CUR Strings[250]

#define key\_f46 CUR Strings[251]

#define key\_f47 CUR Strings[252]

#define key\_f48 CUR Strings[253]

#define key\_f49 CUR Strings[254]

#define key\_f50 CUR Strings[255]

#define key\_f51 CUR Strings[256]

#define key\_f52 CUR Strings[257]

#define key\_f53 CUR Strings[258]

#define key\_f54 CUR Strings[259]

#define key\_f55 CUR Strings[260]

#define key\_f56 CUR Strings[261]

#define key\_f57 CUR Strings[262]

#define key\_f58 CUR Strings[263]

#define key\_f59 CUR Strings[264]

#define key\_f60 CUR Strings[265]

#define key\_f61 CUR Strings[266]

#define key\_f62 CUR Strings[267]

#define key\_f63 CUR Strings[268]

#define clr\_bol CUR Strings[269]

#define clear\_margins CUR Strings[270]

#define set\_left\_margin CUR Strings[271]

#define set\_right\_margin CUR Strings[272]

#define label\_format CUR Strings[273]

#define set\_clock CUR Strings[274]

#define display\_clock CUR Strings[275]

#define remove\_clock CUR Strings[276]

#define create\_window CUR Strings[277]

#define goto\_window CUR Strings[278]

#define hangup CUR Strings[279]

#define dial\_phone CUR Strings[280]

#define quick\_dial CUR Strings[281]

#define tone CUR Strings[282]

#define pulse CUR Strings[283]

#define flash\_hook CUR Strings[284]

#define fixed\_pause CUR Strings[285]

#define wait\_tone CUR Strings[286]

#define user0 CUR Strings[287]

#define user1 CUR Strings[288]

#define user2 CUR Strings[289]

#define user3 CUR Strings[290]

#define user4 CUR Strings[291]

#define user5 CUR Strings[292]

#define user6 CUR Strings[293]

#define user7 CUR Strings[294]

#define user8 CUR Strings[295]

#define user9 CUR Strings[296]

#define orig\_pair CUR Strings[297]

#define orig\_colors CUR Strings[298]

#define initialize\_color CUR Strings[299]

#define initialize\_pair CUR Strings[300]

#define set\_color\_pair CUR Strings[301]

#define set\_foreground CUR Strings[302]

#define set\_background CUR Strings[303]

#define change\_char\_pitch CUR Strings[304]

#define change\_line\_pitch CUR Strings[305]

#define change\_res\_horz CUR Strings[306]

#define change\_res\_vert CUR Strings[307]

#define define\_char CUR Strings[308]

#define enter\_doublewide\_mode CUR Strings[309]

#define enter\_draft\_quality CUR Strings[310]

#define enter\_italics\_mode CUR Strings[311]

#define enter\_leftward\_mode CUR Strings[312]

#define enter\_micro\_mode CUR Strings[313]

#define enter\_near\_letter\_quality CUR Strings[314]

#define enter\_normal\_quality CUR Strings[315]

#define enter\_shadow\_mode CUR Strings[316]

#define enter\_subscript\_mode CUR Strings[317]

#define enter\_superscript\_mode CUR Strings[318]

#define enter\_upward\_mode CUR Strings[319]

#define exit\_doublewide\_mode CUR Strings[320]

#define exit\_italics\_mode CUR Strings[321]

#define exit\_leftward\_mode CUR Strings[322]

#define exit\_micro\_mode CUR Strings[323]

#define exit\_shadow\_mode CUR Strings[324]

#define exit\_subscript\_mode CUR Strings[325]

#define exit\_superscript\_mode CUR Strings[326]

#define exit\_upward\_mode CUR Strings[327]

#define micro\_column\_address CUR Strings[328]

#define micro\_down CUR Strings[329]

#define micro\_left CUR Strings[330]

#define micro\_right CUR Strings[331]

#define micro\_row\_address CUR Strings[332]

#define micro\_up CUR Strings[333]

#define order\_of\_pins CUR Strings[334]

#define parm\_down\_micro CUR Strings[335]

#define parm\_left\_micro CUR Strings[336]

#define parm\_right\_micro CUR Strings[337]

#define parm\_up\_micro CUR Strings[338]

#define select\_char\_set CUR Strings[339]

#define set\_bottom\_margin CUR Strings[340]

#define set\_bottom\_margin\_parm CUR Strings[341]

#define set\_left\_margin\_parm CUR Strings[342]

#define set\_right\_margin\_parm CUR Strings[343]

#define set\_top\_margin CUR Strings[344]

#define set\_top\_margin\_parm CUR Strings[345]

#define start\_bit\_image CUR Strings[346]

#define start\_char\_set\_def CUR Strings[347]

#define stop\_bit\_image CUR Strings[348]

#define stop\_char\_set\_def CUR Strings[349]

#define subscript\_characters CUR Strings[350]

#define superscript\_characters CUR Strings[351]

#define these\_cause\_cr CUR Strings[352]

#define zero\_motion CUR Strings[353]

#define char\_set\_names CUR Strings[354]

#define key\_mouse CUR Strings[355]

#define mouse\_info CUR Strings[356]

#define req\_mouse\_pos CUR Strings[357]

#define get\_mouse CUR Strings[358]

#define set\_a\_foreground CUR Strings[359]

#define set\_a\_background CUR Strings[360]

#define pkey\_plab CUR Strings[361]

#define device\_type CUR Strings[362]

#define code\_set\_init CUR Strings[363]

#define set0\_des\_seq CUR Strings[364]

#define set1\_des\_seq CUR Strings[365]

#define set2\_des\_seq CUR Strings[366]

#define set3\_des\_seq CUR Strings[367]

#define set\_lr\_margin CUR Strings[368]

#define set\_tb\_margin CUR Strings[369]

#define bit\_image\_repeat CUR Strings[370]

#define bit\_image\_newline CUR Strings[371]

#define bit\_image\_carriage\_return CUR Strings[372]

#define color\_names CUR Strings[373]

#define define\_bit\_image\_region CUR Strings[374]

#define end\_bit\_image\_region CUR Strings[375]

#define set\_color\_band CUR Strings[376]

#define set\_page\_length CUR Strings[377]

#define display\_pc\_char CUR Strings[378]

#define enter\_pc\_charset\_mode CUR Strings[379]

#define exit\_pc\_charset\_mode CUR Strings[380]

#define enter\_scancode\_mode CUR Strings[381]

#define exit\_scancode\_mode CUR Strings[382]

#define pc\_term\_options CUR Strings[383]

#define scancode\_escape CUR Strings[384]

#define alt\_scancode\_esc CUR Strings[385]

#define enter\_horizontal\_hl\_mode CUR Strings[386]

#define enter\_left\_hl\_mode CUR Strings[387]

#define enter\_low\_hl\_mode CUR Strings[388]

#define enter\_right\_hl\_mode CUR Strings[389]

#define enter\_top\_hl\_mode CUR Strings[390]

#define enter\_vertical\_hl\_mode CUR Strings[391]

#define set\_a\_attributes CUR Strings[392]

#define set\_pglen\_inch CUR Strings[393]

#define BOOLWRITE 37

#define NUMWRITE 33

#define STRWRITE 394

#define beehive\_glitch no\_esc\_ctlc

#define teleray\_glitch dest\_tabs\_magic\_smso

#define micro\_char\_size micro\_col\_size

#define termcap\_init2 CUR Strings[394]

#define termcap\_reset CUR Strings[395]

#define magic\_cookie\_glitch\_ul CUR Numbers[33]

#define backspaces\_with\_bs CUR Booleans[37]

#define crt\_no\_scrolling CUR Booleans[38]

#define no\_correctly\_working\_cr CUR Booleans[39]

#define carriage\_return\_delay CUR Numbers[34]

#define new\_line\_delay CUR Numbers[35]

#define linefeed\_if\_not\_lf CUR Strings[396]

#define backspace\_if\_not\_bs CUR Strings[397]

#define gnu\_has\_meta\_key CUR Booleans[40]

#define linefeed\_is\_newline CUR Booleans[41]

#define backspace\_delay CUR Numbers[36]

#define horizontal\_tab\_delay CUR Numbers[37]

#define number\_of\_function\_keys CUR Numbers[38]

#define other\_non\_function\_keys CUR Strings[398]

#define arrow\_key\_map CUR Strings[399]

#define has\_hardware\_tabs CUR Booleans[42]

#define return\_does\_clr\_eol CUR Booleans[43]

#define acs\_ulcorner CUR Strings[400]

#define acs\_llcorner CUR Strings[401]

#define acs\_urcorner CUR Strings[402]

#define acs\_lrcorner CUR Strings[403]

#define acs\_ltee CUR Strings[404]

#define acs\_rtee CUR Strings[405]

#define acs\_btee CUR Strings[406]

#define acs\_ttee CUR Strings[407]

#define acs\_hline CUR Strings[408]

#define acs\_vline CUR Strings[409]

#define acs\_plus CUR Strings[410]

#define memory\_lock CUR Strings[411]

#define memory\_unlock CUR Strings[412]

#define box\_chars\_1 CUR Strings[413]

#define BOOLCOUNT 44

#define NUMCOUNT 39

#define STRCOUNT 414

#define acs\_chars\_index 146

#define cur\_term NCURSES\_PUBLIC\_VAR(cur\_term())

#define boolnames NCURSES\_PUBLIC\_VAR(boolnames())

#define boolcodes NCURSES\_PUBLIC\_VAR(boolcodes())

#define boolfnames NCURSES\_PUBLIC\_VAR(boolfnames())

#define numnames NCURSES\_PUBLIC\_VAR(numnames())

#define numcodes NCURSES\_PUBLIC\_VAR(numcodes())

#define numfnames NCURSES\_PUBLIC\_VAR(numfnames())

#define strnames NCURSES\_PUBLIC\_VAR(strnames())

#define strcodes NCURSES\_PUBLIC\_VAR(strcodes())

#define strfnames NCURSES\_PUBLIC\_VAR(strfnames())

typedef struct termtype {

char \*term\_names;

char \*str\_table;

char \*Booleans;

short \*Numbers;

char \*\*Strings;

char \*ext\_str\_table;

char \*\*ext\_Names;

unsigned short num\_Booleans;

unsigned short num\_Numbers;

unsigned short num\_Strings;

unsigned short ext\_Booleans;

unsigned short ext\_Numbers;

unsigned short ext\_Strings;

} TERMTYPE;

typedef struct term {

TERMTYPE type;

short Filedes;

struct termios Ottyb;

struct termios Nttyb;

int \_baudrate;

char \*\_termname;

} TERMINAL;

extern TERMINAL \*cur\_term;

extern int del\_curterm(TERMINAL \*);

extern int putp(const char \*);

extern int restartterm(char \*, int, int \*);

extern TERMINAL \*set\_curterm(TERMINAL \*);

extern int setupterm(char \*, int, int \*);

extern int tgetent(char \*, const char \*);

extern int tgetflag(char \*);

extern int tgetnum(char \*);

extern char \*tgetstr(char \*, char \*\*);

extern char \*tgoto(const char \*, int, int);

extern int tigetflag(char \*);

extern int tigetnum(char \*);

extern char \*tigetstr(char \*);

extern char \*tparm(char \*, ...);

extern int tputs(const char \*, int, int (\*)(int));

extern char ttytype[];

### **15.9.4 ncursesw/unctrl.h**

#define NCURSES\_UNCTRL\_H\_incl 1

#define NCURSES\_VERSION "5.9"

## **15.10 Interface Definitions for libncursesw**

The interfaces defined on the following pages are included in libncursesw and are defined by this specification. Unless otherwise noted, these interfaces shall be included in the source standard.

Other interfaces listed in [Section 15.8](#ID_LIBNCURSESW) shall behave as described in the referenced base document.

## **15.11 Interfaces for libutil**

[Table 15-11](#ID_LIB_45_LIBUTIL_45_DEF) defines the library name and shared object name for the libutil library

**Table 15-11 libutil Definition**

|  |  |
| --- | --- |
| Library: | libutil |
| SONAME: | libutil.so.1 |

The behavior of the interfaces in this library is specified by the following specifications:

|  |
| --- |
| [LSB] [This Specification](#ID_STD_46_LSB) |

### **15.11.1 Utility Functions**

#### 15.11.1.1 Interfaces for Utility Functions

An LSB conforming implementation shall provide the generic functions for Utility Functions specified in [Table 15-12](#ID_TBL_45_LIBUTIL_45_UTILI_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 15-12 libutil - Utility Functions Function Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| forkpty [[LSB]](#ID_REFSTD_46_LIBUTIL_46_1) | login [[LSB]](#ID_REFSTD_46_LIBUTIL_46_1) | login\_tty [[LSB]](#ID_REFSTD_46_LIBUTIL_46_1) | logout [[LSB]](#ID_REFSTD_46_LIBUTIL_46_1) |
| logwtmp [[LSB]](#ID_REFSTD_46_LIBUTIL_46_1) | openpty [[LSB]](#ID_REFSTD_46_LIBUTIL_46_1) |  |  |

## **15.12 Data Definitions for libutil**

This section defines global identifiers and their values that are associated with interfaces contained in libutil. These definitions are organized into groups that correspond to system headers. This convention is used as a convenience for the reader, and does not imply the existence of these headers, or their content. Where an interface is defined as requiring a particular system header file all of the data definitions for that system header file presented here shall be in effect.

This section gives data definitions to promote binary application portability, not to repeat source interface definitions available elsewhere. System providers and application developers should use this ABI to supplement - not to replace - source interface definition specifications.

This specification uses the [ISO C (1999)](#ID_STD_46_ISOC99) C Language as the reference programming language, and data definitions are specified in ISO C format. The C language is used here as a convenient notation. Using a C language description of these data objects does not preclude their use by other programming languages.

### **15.12.1 pty.h**

extern int forkpty(int \*\_\_amaster, char \*\_\_name,

const struct termios \*\_\_termp,

const struct winsize \*\_\_winp);

extern int openpty(int \*\_\_amaster, int \*\_\_aslave, char \*\_\_name,

const struct termios \*\_\_termp,

const struct winsize \*\_\_winp);

## **15.13 Interface Definitions for libutil**

The interfaces defined on the following pages are included in libutil and are defined by this specification. Unless otherwise noted, these interfaces shall be included in the source standard.

Other interfaces listed in [Section 15.11](#ID_LIBUTIL) shall behave as described in the referenced base document.

#### forkpty

##### Name

forkpty — Create a new process attached to an available pseudo-terminal

##### Synopsis

#include <pty.h>

int forkpty(int \* *amaster*, char \* *name*, const struct termios \* *termp*, const struct winsize \* *winp*);

##### Description

The forkpty() function shall find and open a pseudo-terminal device pair in the same manner as the openpty() function. If a pseudo-terminal is available, forkpty() shall create a new process in the same manner as the fork() function, and prepares the new process for login in the same manner as login\_tty().

If *termp* is not null, it shall refer to a termios structure that shall be used to initialize the characteristics of the slave device. If *winp* is not null, it shall refer to a winsize structure used to initialize the window size of the slave device.

##### Return Value

On success, the parent process shall return the process id of the child, and the child shall return 0. On error, no new process shall be created, -1 shall be returned, and errno shall be set appropriately. On success, the parent process shall receive the file descriptor of the master side of the pseudo-terminal in the location referenced by *amaster*, and, if *name* is not NULL, the filename of the slave device in *name*.

##### Errors

EAGAIN

  Unable to create a new process.

ENOENT

  There are no available pseudo-terminals.

ENOMEM

  Insufficient memory was available.

#### login

##### Name

login — login utility function

##### Synopsis

#include <utmp.h>

void login (struct utmp \* *ut* );

##### Description

The login() function shall update the user accounting databases. The *ut* parameter shall reference a utmp structure for all fields except the following:

1. The *ut\_type* field shall be set to USER\_PROCESS.

2. The *ut\_pid* field shall be set to the process identifier for the current process.

3. The *ut\_line* field shall be set to the name of the controlling terminal device. The name shall be found by examining the device associated with the standard input, output and error streams in sequence, until one associated with a terminal device is found. If none of these streams refers to a terminal device, the *ut\_line* field shall be set to "???". If the terminal device is in the /dev directory hierarchy, the *ut\_line* field shall not contain the leading "/dev/", otherwise it shall be set to the final component of the pathname of the device. If the user accounting database imposes a limit on the size of the *ut\_line* field, it shall truncate the name, but any such limit shall not be smaller than UT\_LINESIZE (including a terminating null character).

##### Return Value

None

##### Errors

None

#### login\_tty

##### Name

login\_tty — Prepare a terminal for login

##### Synopsis

#include <utmp.h>

int login\_tty (int *fdr*);

##### Description

The login\_tty() function shall prepare the terminal device referenced by the file descriptor *fdr*. This function shall create a new session, make the terminal the controlling terminal for the current process, and set the standard input, output, and error streams of the current process to the terminal. If *fdr* is not the standard input, output or error stream, then login\_tty() shall close *fdr*.

##### Return Value

On success, login\_tty() shall return zero; otherwise -1 is returned, and errno shall be set appropriately.

##### Errors

ENOTTY

*fdr* does not refer to a terminal device.

#### logout

##### Name

logout — logout utility function

##### Synopsis

#include <utmp.h>

int logout (const char \* *line* );

##### Description

Given the device *line*, the logout() function shall search the user accounting database which is read by getutent() for an entry with the corresponding line, and with the type of USER\_PROCESS. If a corresponding entry is located, it shall be updated as follows:

1. The ut\_name field shall be set to zeroes (UT\_NAMESIZE NUL bytes).

2. The ut\_host field shall be set to zeroes (UT\_HOSTSIZE NUL bytes).

3. The ut\_tv shall be set to the current time of day.

4. The ut\_type field shall be set to DEAD\_PROCESS.

##### Return Value

On success, the logout() function shall return non-zero. Zero is returned if there was no entry to remove, or if the utmp file could not be opened or updated.

#### logwtmp

##### Name

logwtmp — append an entry to the wtmp file

##### Synopsis

#include <utmp.h>

void logwtmp (const char \* *line* , const char \* *name* , const char \* *host* );

##### Description

If the process has permission to update the user accounting databases, the logwtmp() function shall append a record to the user accounting database that records all logins and logouts. The record to be appended shall be constructed as follows:

1. The ut\_line field shall be initialized from *line*. If the user accounting database imposes a limit on the size of the *ut\_line* field, it shall truncate the value, but any such limit shall not be smaller than UT\_LINESIZE (including a terminating null character).

2. The ut\_name field shall be initialized from *name*. If the user accounting database imposes a limit on the size of the *ut\_name* field, it shall truncate the value, but any such limit shall not be smaller than UT\_NAMESIZE (including a terminating null character).

3. The ut\_host field shall be initialized from *host*. If the user accounting database imposes a limit on the size of the *ut\_host* field, it shall truncate the value, but any such limit shall not be smaller than UT\_HOSTSIZE (including a terminating null character).

4. If the *name* parameter does not refer to an empty string (i.e. ""), the ut\_type field shall be set to USER\_PROCESS; otherwise the ut\_type field shall be set to DEAD\_PROCESS.

5. The ut\_id field shall be set to the process identifier for the current process.

6. The ut\_tv field shall be set to the current time of day.

**Note:** If a process does not have write access to the the user accounting database, the logwtmp() function will not update it. Since the function does not return any value, an application has no way of knowing whether it succeeded or failed.

##### Return Value

None.

#### openpty

##### Name

openpty — find and open an available pseudo-terminal

##### Synopsis

#include <pty.h>

int openpty(int \**amaster*, int \**aslave*, char \**name*, const struct termios \**termp*, const struct winsize \**winp*);

##### Description

The openpty() function shall find an available pseudo-terminal and return file descriptors for the master and slave devices in the locations referenced by *amaster* and *aslave* respectively. If *name* is not NULL, the filename of the slave shall be placed in the user supplied buffer referenced by *name*. If *termp* is not NULL, it shall point to a termios structure used to initialize the terminal parameters of the slave pseudo-terminal device. If *winp* is not NULL, it shall point to a winsize structure used to initialize the window size parameters of the slave pseudo-terminal device.

##### Return Value

On success, zero is returned. On error, -1 is returned, and errno is set appropriately.

##### Errors

ENOENT

  There are no available pseudo-terminals.

# **V C++ Libraries**

# **16 Libraries**

An LSB-conforming implementation shall support some C++ libraries which provide interfaces for accessing the operating system, processor and other hardware in the system.

## **16.1 Interfaces for libstdcxx**

[Table 16-1](#ID_LIB_45_LIBSTDCXX_45_DEF) defines the library name and shared object name for the libstdcxx library

**Table 16-1 libstdcxx Definition**

|  |  |
| --- | --- |
| Library: | libstdcxx |
| SONAME: | libstdc++.so.6 |

Unless stated otherwise, all symbols are in the std:: namespace.

The behavior of the interfaces in this library is specified by the following specifications:

|  |
| --- |
| [CXXABI-1.86] [Itanium™ C++ ABI](#ID_STD_46_CXXABI) |
| [ISOCXX] [ISO/IEC 14882: 2003 C++ Language](#ID_STD_46_ISOCXX) |
| [LSB] [This Specification](#ID_STD_46_LSB) |

### **16.1.1 C++ Runtime Support**

#### 16.1.1.1 Interfaces for C++ Runtime Support

An LSB conforming implementation shall provide the generic methods for C++ Runtime Support specified in [Table 16-2](#ID_TBL_45_LIBSTDCXX_45_C_45_R_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-2 libstdcxx - C++ Runtime Support Function Interfaces**

|  |
| --- |
| \_\_gnu\_cxx::\_\_atomic\_add(int volatile\*, int)(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_gnu\_cxx::\_\_exchange\_and\_add(int volatile\*, int)(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_gnu\_cxx::\_\_verbose\_terminate\_handler()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| unexpected()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| set\_terminate(void (\*)())(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| set\_unexpected(void (\*)())(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| set\_new\_handler(void (\*)())(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_throw\_bad\_cast()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_throw\_bad\_alloc()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_throw\_bad\_typeid()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| uncaught\_exception()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_throw\_ios\_failure(char const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_throw\_logic\_error(char const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_throw\_range\_error(char const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_throw\_domain\_error(char const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_throw\_length\_error(char const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_throw\_out\_of\_range(char const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_throw\_bad\_exception()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_throw\_runtime\_error(char const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_throw\_overflow\_error(char const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_throw\_underflow\_error(char const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_throw\_invalid\_argument(char const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| terminate()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| operator delete[](void\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| operator delete[](void\*, nothrow\_t const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| operator delete(void\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| operator delete(void\*, nothrow\_t const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_cxa\_allocate\_exception(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxa\_bad\_cast(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxa\_bad\_typeid(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxa\_begin\_catch(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxa\_call\_unexpected(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxa\_current\_exception\_type(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxa\_demangle(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxa\_end\_catch(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxa\_free\_exception(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxa\_get\_exception\_ptr(CXXABI\_1.3.1) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxa\_get\_globals(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxa\_get\_globals\_fast(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxa\_guard\_abort(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxa\_guard\_acquire(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxa\_guard\_release(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxa\_pure\_virtual(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxa\_rethrow(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxa\_throw(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxa\_vec\_cctor(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxa\_vec\_cleanup(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxa\_vec\_ctor(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxa\_vec\_delete(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxa\_vec\_delete2(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxa\_vec\_delete3(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxa\_vec\_dtor(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxa\_vec\_new(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxa\_vec\_new2(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxa\_vec\_new3(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_dynamic\_cast(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_gxx\_personality\_v0(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

An LSB conforming implementation shall provide the generic data interfaces for C++ Runtime Support specified in [Table 16-3](#ID_TBL_45_LIBSTDCXX_45_C_45_R_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-3 libstdcxx - C++ Runtime Support Data Interfaces**

|  |
| --- |
| cin(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| cerr(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| clog(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| cout(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| wcin(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| wcerr(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| wclog(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| wcout(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| nothrow(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.2 C++ type descriptors for built-in types**

#### 16.1.2.1 Interfaces for C++ type descriptors for built-in types

No external methods are defined for libstdcxx - C++ type descriptors for built-in types in this part of the specification. See also the relevant architecture specific part of this specification.

An LSB conforming implementation shall provide the generic data interfaces for C++ type descriptors for built-in types specified in [Table 16-4](#ID_TBL_45_LIBSTDCXX_45_C_45_T_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-4 libstdcxx - C++ type descriptors for built-in types Data Interfaces**

|  |
| --- |
| typeinfo for signed char const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for bool const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for char const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for double const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for long double const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for float const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for unsigned char const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for int const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for unsigned int const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for long const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for unsigned long const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for short const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for unsigned short const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for void const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for wchar\_t const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for long long const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for unsigned long long const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for signed char\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for bool\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for char\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for double\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for long double\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for float\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for unsigned char\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for int\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for unsigned int\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for long\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for unsigned long\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for short\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for unsigned short\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for void\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for wchar\_t\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for long long\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for unsigned long long\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for signed char(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for bool(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for char(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for double(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for long double(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for float(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for unsigned char(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for int(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for unsigned int(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for long(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for unsigned long(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for short(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for unsigned short(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for void(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for wchar\_t(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for long long(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for unsigned long long(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for signed char const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for bool const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for char const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for double const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for long double const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for float const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for unsigned char const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for int const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for unsigned int const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for long const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for unsigned long const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for short const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for unsigned short const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for void const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for wchar\_t const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for long long const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for unsigned long long const\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for signed char\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for bool\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for char\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for double\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for long double\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for float\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for unsigned char\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for int\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for unsigned int\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for long\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for unsigned long\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for short\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for unsigned short\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for void\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for wchar\_t\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for long long\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for unsigned long long\*(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for signed char(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for bool(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for char(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for double(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for long double(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for float(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for unsigned char(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for int(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for unsigned int(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for long(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for unsigned long(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for short(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for unsigned short(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for void(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for wchar\_t(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for long long(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for unsigned long long(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.3 C++ \_Rb\_tree**

#### 16.1.3.1 Interfaces for C++ \_Rb\_tree

An LSB conforming implementation shall provide the generic methods for C++ \_Rb\_tree specified in [Table 16-5](#ID_TBL_45_LIBSTDCXX_45_C_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-5 libstdcxx - C++ \_Rb\_tree Function Interfaces**

|  |
| --- |
| \_Rb\_tree\_decrement(\_Rb\_tree\_node\_base const\*)(GLIBCXX\_3.4) [[LSB]](#ID_REFSTD_46_LIBSTDCXX_46_3) |
| \_Rb\_tree\_decrement(\_Rb\_tree\_node\_base\*)(GLIBCXX\_3.4) [[LSB]](#ID_REFSTD_46_LIBSTDCXX_46_3) |
| \_Rb\_tree\_increment(\_Rb\_tree\_node\_base const\*)(GLIBCXX\_3.4) [[LSB]](#ID_REFSTD_46_LIBSTDCXX_46_3) |
| \_Rb\_tree\_increment(\_Rb\_tree\_node\_base\*)(GLIBCXX\_3.4) [[LSB]](#ID_REFSTD_46_LIBSTDCXX_46_3) |
| \_Rb\_tree\_black\_count(\_Rb\_tree\_node\_base const\*, \_Rb\_tree\_node\_base const\*)(GLIBCXX\_3.4) [[LSB]](#ID_REFSTD_46_LIBSTDCXX_46_3) |
| \_Rb\_tree\_rotate\_left(\_Rb\_tree\_node\_base\*, \_Rb\_tree\_node\_base\*&)(GLIBCXX\_3.4) [[LSB]](#ID_REFSTD_46_LIBSTDCXX_46_3) |
| \_Rb\_tree\_rotate\_right(\_Rb\_tree\_node\_base\*, \_Rb\_tree\_node\_base\*&)(GLIBCXX\_3.4) [[LSB]](#ID_REFSTD_46_LIBSTDCXX_46_3) |
| \_Rb\_tree\_rebalance\_for\_erase(\_Rb\_tree\_node\_base\*, \_Rb\_tree\_node\_base&)(GLIBCXX\_3.4) [[LSB]](#ID_REFSTD_46_LIBSTDCXX_46_3) |
| \_Rb\_tree\_insert\_and\_rebalance(bool, \_Rb\_tree\_node\_base\*, \_Rb\_tree\_node\_base\*, \_Rb\_tree\_node\_base&)(GLIBCXX\_3.4) [[LSB]](#ID_REFSTD_46_LIBSTDCXX_46_3) |

### **16.1.4 Class type\_info**

#### 16.1.4.1 Class data for type\_info

The virtual table for the std::type\_info class is described by [Table 16-6](#ID_CLS_45_24325_45_0)

**Table 16-6 Primary vtable for type\_info**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for type\_info |
| vfunc[0]: | type\_info::~type\_info() |
| vfunc[1]: | type\_info::~type\_info() |
| vfunc[2]: | type\_info::\_\_is\_pointer\_p() const |
| vfunc[3]: | type\_info::\_\_is\_function\_p() const |
| vfunc[4]: | type\_info::\_\_do\_catch(type\_info const\*, void\*\*, unsigned int) const |
| vfunc[5]: | type\_info::\_\_do\_upcast(\_\_cxxabiv1::\_\_class\_type\_info const\*, void\*\*) const |

The Run Time Type Information for the std::type\_info class is described by [Table 16-7](#ID_RTTI_45_24030)

**Table 16-7 typeinfo for type\_info**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_class\_type\_info |
| Name | typeinfo name for type\_info |

#### 16.1.4.2 Interfaces for Class type\_info

An LSB conforming implementation shall provide the generic methods for Class std::type\_info specified in [Table 16-8](#ID_TBL_45_LIBSTDCXX_45_CLASS_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-8 libstdcxx - Class type\_info Function Interfaces**

|  |
| --- |
| type\_info::\_\_do\_catch(type\_info const\*, void\*\*, unsigned int) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| type\_info::\_\_do\_upcast(\_\_cxxabiv1::\_\_class\_type\_info const\*, void\*\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| type\_info::\_\_is\_pointer\_p() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| type\_info::\_\_is\_function\_p() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| type\_info::~type\_info()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| type\_info::~type\_info()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| type\_info::~type\_info()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::type\_info specified in [Table 16-9](#ID_TBL_45_LIBSTDCXX_45_CLASS_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-9 libstdcxx - Class type\_info Data Interfaces**

|  |
| --- |
| typeinfo for type\_info(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for type\_info(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for type\_info(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.5 Class \_\_cxxabiv1::\_\_enum\_type\_info**

#### 16.1.5.1 Class data for \_\_cxxabiv1::\_\_enum\_type\_info

The virtual table for the \_\_cxxabiv1::\_\_enum\_type\_info class is described by [Table 16-10](#ID_CLS_45_24328_45_0)

**Table 16-10 Primary vtable for \_\_cxxabiv1::\_\_enum\_type\_info**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for \_\_cxxabiv1::\_\_enum\_type\_info |
| vfunc[0]: | \_\_cxxabiv1::\_\_enum\_type\_info::~\_\_enum\_type\_info() |
| vfunc[1]: | \_\_cxxabiv1::\_\_enum\_type\_info::~\_\_enum\_type\_info() |
| vfunc[2]: | type\_info::\_\_is\_pointer\_p() const |
| vfunc[3]: | type\_info::\_\_is\_function\_p() const |
| vfunc[4]: | type\_info::\_\_do\_catch(type\_info const\*, void\*\*, unsigned int) const |
| vfunc[5]: | type\_info::\_\_do\_upcast(\_\_cxxabiv1::\_\_class\_type\_info const\*, void\*\*) const |

The Run Time Type Information for the \_\_cxxabiv1::\_\_enum\_type\_info class is described by [Table 16-11](#ID_RTTI_45_24099)

**Table 16-11 typeinfo for \_\_cxxabiv1::\_\_enum\_type\_info**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for \_\_cxxabiv1::\_\_enum\_type\_info |

#### 16.1.5.2 Interfaces for Class \_\_cxxabiv1::\_\_enum\_type\_info

An LSB conforming implementation shall provide the generic methods for Class \_\_cxxabiv1::\_\_enum\_type\_info specified in [Table 16-12](#ID_TBL_45_LIBSTDCXX_45_CLAST_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-12 libstdcxx - Class \_\_cxxabiv1::\_\_enum\_type\_info Function Interfaces**

|  |
| --- |
| \_\_cxxabiv1::\_\_enum\_type\_info::~\_\_enum\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_enum\_type\_info::~\_\_enum\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_enum\_type\_info::~\_\_enum\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

An LSB conforming implementation shall provide the generic data interfaces for Class \_\_cxxabiv1::\_\_enum\_type\_info specified in [Table 16-13](#ID_TBL_45_LIBSTDCXX_45_CLAST_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-13 libstdcxx - Class \_\_cxxabiv1::\_\_enum\_type\_info Data Interfaces**

|  |
| --- |
| typeinfo for \_\_cxxabiv1::\_\_enum\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for \_\_cxxabiv1::\_\_enum\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for \_\_cxxabiv1::\_\_enum\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.6 Class \_\_cxxabiv1::\_\_array\_type\_info**

#### 16.1.6.1 Class data for \_\_cxxabiv1::\_\_array\_type\_info

The virtual table for the \_\_cxxabiv1::\_\_array\_type\_info class is described by [Table 16-14](#ID_CLS_45_24330_45_0)

**Table 16-14 Primary vtable for \_\_cxxabiv1::\_\_array\_type\_info**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for \_\_cxxabiv1::\_\_array\_type\_info |
| vfunc[0]: | \_\_cxxabiv1::\_\_array\_type\_info::~\_\_array\_type\_info() |
| vfunc[1]: | \_\_cxxabiv1::\_\_array\_type\_info::~\_\_array\_type\_info() |
| vfunc[2]: | type\_info::\_\_is\_pointer\_p() const |
| vfunc[3]: | type\_info::\_\_is\_function\_p() const |
| vfunc[4]: | type\_info::\_\_do\_catch(type\_info const\*, void\*\*, unsigned int) const |
| vfunc[5]: | type\_info::\_\_do\_upcast(\_\_cxxabiv1::\_\_class\_type\_info const\*, void\*\*) const |

The Run Time Type Information for the \_\_cxxabiv1::\_\_array\_type\_info class is described by [Table 16-15](#ID_RTTI_45_24101)

**Table 16-15 typeinfo for \_\_cxxabiv1::\_\_array\_type\_info**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for \_\_cxxabiv1::\_\_array\_type\_info |

#### 16.1.6.2 Interfaces for Class \_\_cxxabiv1::\_\_array\_type\_info

An LSB conforming implementation shall provide the generic methods for Class \_\_cxxabiv1::\_\_array\_type\_info specified in [Table 16-16](#ID_TBL_45_LIBSTDCXX_45_CLASU_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-16 libstdcxx - Class \_\_cxxabiv1::\_\_array\_type\_info Function Interfaces**

|  |
| --- |
| \_\_cxxabiv1::\_\_array\_type\_info::~\_\_array\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_array\_type\_info::~\_\_array\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_array\_type\_info::~\_\_array\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

An LSB conforming implementation shall provide the generic data interfaces for Class \_\_cxxabiv1::\_\_array\_type\_info specified in [Table 16-17](#ID_TBL_45_LIBSTDCXX_45_CLASU_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-17 libstdcxx - Class \_\_cxxabiv1::\_\_array\_type\_info Data Interfaces**

|  |
| --- |
| typeinfo for \_\_cxxabiv1::\_\_array\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for \_\_cxxabiv1::\_\_array\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for \_\_cxxabiv1::\_\_array\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.7 Class \_\_cxxabiv1::\_\_class\_type\_info**

#### 16.1.7.1 Class data for \_\_cxxabiv1::\_\_class\_type\_info

The virtual table for the \_\_cxxabiv1::\_\_class\_type\_info class is described by [Table 16-18](#ID_CLS_45_23981_45_0)

**Table 16-18 Primary vtable for \_\_cxxabiv1::\_\_class\_type\_info**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for \_\_cxxabiv1::\_\_class\_type\_info |
| vfunc[0]: | \_\_cxxabiv1::\_\_class\_type\_info::~\_\_class\_type\_info() |
| vfunc[1]: | \_\_cxxabiv1::\_\_class\_type\_info::~\_\_class\_type\_info() |
| vfunc[2]: | type\_info::\_\_is\_pointer\_p() const |
| vfunc[3]: | type\_info::\_\_is\_function\_p() const |
| vfunc[4]: | \_\_cxxabiv1::\_\_class\_type\_info::\_\_do\_catch(type\_info const\*, void\*\*, unsigned int) const |
| vfunc[5]: | \_\_cxxabiv1::\_\_class\_type\_info::\_\_do\_upcast(\_\_cxxabiv1::\_\_class\_type\_info const\*, void\*\*) const |
| vfunc[6]: | \_\_cxxabiv1::\_\_class\_type\_info::\_\_do\_upcast(\_\_cxxabiv1::\_\_class\_type\_info const\*, void const\*, \_\_cxxabiv1::\_\_class\_type\_info::\_\_upcast\_result&) const |

The Run Time Type Information for the \_\_cxxabiv1::\_\_class\_type\_info class is described by [Table 16-19](#ID_RTTI_45_24031)

**Table 16-19 typeinfo for \_\_cxxabiv1::\_\_class\_type\_info**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for \_\_cxxabiv1::\_\_class\_type\_info |

#### 16.1.7.2 Interfaces for Class \_\_cxxabiv1::\_\_class\_type\_info

An LSB conforming implementation shall provide the generic methods for Class \_\_cxxabiv1::\_\_class\_type\_info specified in [Table 16-20](#ID_TBL_45_LIBSTDCXX_45_CLASV_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-20 libstdcxx - Class \_\_cxxabiv1::\_\_class\_type\_info Function Interfaces**

|  |
| --- |
| \_\_cxxabiv1::\_\_class\_type\_info::~\_\_class\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_class\_type\_info::~\_\_class\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_class\_type\_info::~\_\_class\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_class\_type\_info::\_\_do\_catch(type\_info const\*, void\*\*, unsigned int) const(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_class\_type\_info::\_\_do\_upcast(\_\_cxxabiv1::\_\_class\_type\_info const\*, void const\*, \_\_cxxabiv1::\_\_class\_type\_info::\_\_upcast\_result&) const(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_class\_type\_info::\_\_do\_upcast(\_\_cxxabiv1::\_\_class\_type\_info const\*, void\*\*) const(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

An LSB conforming implementation shall provide the generic data interfaces for Class \_\_cxxabiv1::\_\_class\_type\_info specified in [Table 16-21](#ID_TBL_45_LIBSTDCXX_45_CLASV_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-21 libstdcxx - Class \_\_cxxabiv1::\_\_class\_type\_info Data Interfaces**

|  |
| --- |
| typeinfo for \_\_cxxabiv1::\_\_class\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for \_\_cxxabiv1::\_\_class\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for \_\_cxxabiv1::\_\_class\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.8 Class \_\_cxxabiv1::\_\_pbase\_type\_info**

#### 16.1.8.1 Class data for \_\_cxxabiv1::\_\_pbase\_type\_info

The virtual table for the \_\_cxxabiv1::\_\_pbase\_type\_info class is described by [Table 16-22](#ID_CLS_45_24327_45_0)

**Table 16-22 Primary vtable for \_\_cxxabiv1::\_\_pbase\_type\_info**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for \_\_cxxabiv1::\_\_pbase\_type\_info |
| vfunc[0]: | \_\_cxxabiv1::\_\_pbase\_type\_info::~\_\_pbase\_type\_info() |
| vfunc[1]: | \_\_cxxabiv1::\_\_pbase\_type\_info::~\_\_pbase\_type\_info() |
| vfunc[2]: | type\_info::\_\_is\_pointer\_p() const |
| vfunc[3]: | type\_info::\_\_is\_function\_p() const |
| vfunc[4]: | \_\_cxxabiv1::\_\_pbase\_type\_info::\_\_do\_catch(type\_info const\*, void\*\*, unsigned int) const |
| vfunc[5]: | type\_info::\_\_do\_upcast(\_\_cxxabiv1::\_\_class\_type\_info const\*, void\*\*) const |
| vfunc[6]: | \_\_cxxabiv1::\_\_pbase\_type\_info::\_\_pointer\_catch(\_\_cxxabiv1::\_\_pbase\_type\_info const\*, void\*\*, unsigned int) const |

The Run Time Type Information for the \_\_cxxabiv1::\_\_pbase\_type\_info class is described by [Table 16-23](#ID_RTTI_45_24032)

**Table 16-23 typeinfo for \_\_cxxabiv1::\_\_pbase\_type\_info**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for \_\_cxxabiv1::\_\_pbase\_type\_info |

#### 16.1.8.2 Interfaces for Class \_\_cxxabiv1::\_\_pbase\_type\_info

An LSB conforming implementation shall provide the generic methods for Class \_\_cxxabiv1::\_\_pbase\_type\_info specified in [Table 16-24](#ID_TBL_45_LIBSTDCXX_45_CLASW_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-24 libstdcxx - Class \_\_cxxabiv1::\_\_pbase\_type\_info Function Interfaces**

|  |
| --- |
| \_\_cxxabiv1::\_\_pbase\_type\_info::~\_\_pbase\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_pbase\_type\_info::~\_\_pbase\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_pbase\_type\_info::~\_\_pbase\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_pbase\_type\_info::\_\_do\_catch(type\_info const\*, void\*\*, unsigned int) const(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_pbase\_type\_info::\_\_pointer\_catch(\_\_cxxabiv1::\_\_pbase\_type\_info const\*, void\*\*, unsigned int) const(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

An LSB conforming implementation shall provide the generic data interfaces for Class \_\_cxxabiv1::\_\_pbase\_type\_info specified in [Table 16-25](#ID_TBL_45_LIBSTDCXX_45_CLASW_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-25 libstdcxx - Class \_\_cxxabiv1::\_\_pbase\_type\_info Data Interfaces**

|  |
| --- |
| typeinfo for \_\_cxxabiv1::\_\_pbase\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for \_\_cxxabiv1::\_\_pbase\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for \_\_cxxabiv1::\_\_pbase\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.9 Class \_\_cxxabiv1::\_\_pointer\_type\_info**

#### 16.1.9.1 Class data for \_\_cxxabiv1::\_\_pointer\_type\_info

The virtual table for the \_\_cxxabiv1::\_\_pointer\_type\_info class is described by [Table 16-26](#ID_CLS_45_23983_45_0)

**Table 16-26 Primary vtable for \_\_cxxabiv1::\_\_pointer\_type\_info**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for \_\_cxxabiv1::\_\_pointer\_type\_info |
| vfunc[0]: | \_\_cxxabiv1::\_\_pointer\_type\_info::~\_\_pointer\_type\_info() |
| vfunc[1]: | \_\_cxxabiv1::\_\_pointer\_type\_info::~\_\_pointer\_type\_info() |
| vfunc[2]: | \_\_cxxabiv1::\_\_pointer\_type\_info::\_\_is\_pointer\_p() const |
| vfunc[3]: | type\_info::\_\_is\_function\_p() const |
| vfunc[4]: | \_\_cxxabiv1::\_\_pbase\_type\_info::\_\_do\_catch(type\_info const\*, void\*\*, unsigned int) const |
| vfunc[5]: | type\_info::\_\_do\_upcast(\_\_cxxabiv1::\_\_class\_type\_info const\*, void\*\*) const |
| vfunc[6]: | \_\_cxxabiv1::\_\_pointer\_type\_info::\_\_pointer\_catch(\_\_cxxabiv1::\_\_pbase\_type\_info const\*, void\*\*, unsigned int) const |

The Run Time Type Information for the \_\_cxxabiv1::\_\_pointer\_type\_info class is described by [Table 16-27](#ID_RTTI_45_24098)

**Table 16-27 typeinfo for \_\_cxxabiv1::\_\_pointer\_type\_info**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for \_\_cxxabiv1::\_\_pointer\_type\_info |

#### 16.1.9.2 Interfaces for Class \_\_cxxabiv1::\_\_pointer\_type\_info

An LSB conforming implementation shall provide the generic methods for Class \_\_cxxabiv1::\_\_pointer\_type\_info specified in [Table 16-28](#ID_TBL_45_LIBSTDCXX_45_CLASX_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-28 libstdcxx - Class \_\_cxxabiv1::\_\_pointer\_type\_info Function Interfaces**

|  |
| --- |
| \_\_cxxabiv1::\_\_pointer\_type\_info::~\_\_pointer\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_pointer\_type\_info::~\_\_pointer\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_pointer\_type\_info::~\_\_pointer\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_pointer\_type\_info::\_\_is\_pointer\_p() const(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_pointer\_type\_info::\_\_pointer\_catch(\_\_cxxabiv1::\_\_pbase\_type\_info const\*, void\*\*, unsigned int) const(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

An LSB conforming implementation shall provide the generic data interfaces for Class \_\_cxxabiv1::\_\_pointer\_type\_info specified in [Table 16-29](#ID_TBL_45_LIBSTDCXX_45_CLASX_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-29 libstdcxx - Class \_\_cxxabiv1::\_\_pointer\_type\_info Data Interfaces**

|  |
| --- |
| typeinfo for \_\_cxxabiv1::\_\_pointer\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for \_\_cxxabiv1::\_\_pointer\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for \_\_cxxabiv1::\_\_pointer\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.10 Class \_\_cxxabiv1::\_\_function\_type\_info**

#### 16.1.10.1 Class data for \_\_cxxabiv1::\_\_function\_type\_info

The virtual table for the \_\_cxxabiv1::\_\_function\_type\_info class is described by [Table 16-30](#ID_CLS_45_24329_45_0)

**Table 16-30 Primary vtable for \_\_cxxabiv1::\_\_function\_type\_info**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for \_\_cxxabiv1::\_\_function\_type\_info |
| vfunc[0]: | \_\_cxxabiv1::\_\_function\_type\_info::~\_\_function\_type\_info() |
| vfunc[1]: | \_\_cxxabiv1::\_\_function\_type\_info::~\_\_function\_type\_info() |
| vfunc[2]: | type\_info::\_\_is\_pointer\_p() const |
| vfunc[3]: | \_\_cxxabiv1::\_\_function\_type\_info::\_\_is\_function\_p() const |
| vfunc[4]: | type\_info::\_\_do\_catch(type\_info const\*, void\*\*, unsigned int) const |
| vfunc[5]: | type\_info::\_\_do\_upcast(\_\_cxxabiv1::\_\_class\_type\_info const\*, void\*\*) const |

The Run Time Type Information for the \_\_cxxabiv1::\_\_function\_type\_info class is described by [Table 16-31](#ID_RTTI_45_24100)

**Table 16-31 typeinfo for \_\_cxxabiv1::\_\_function\_type\_info**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for \_\_cxxabiv1::\_\_function\_type\_info |

#### 16.1.10.2 Interfaces for Class \_\_cxxabiv1::\_\_function\_type\_info

An LSB conforming implementation shall provide the generic methods for Class \_\_cxxabiv1::\_\_function\_type\_info specified in [Table 16-32](#ID_TBL_45_LIBSTDCXX_45_CLASY_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-32 libstdcxx - Class \_\_cxxabiv1::\_\_function\_type\_info Function Interfaces**

|  |
| --- |
| \_\_cxxabiv1::\_\_function\_type\_info::~\_\_function\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_function\_type\_info::~\_\_function\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_function\_type\_info::~\_\_function\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_function\_type\_info::\_\_is\_function\_p() const(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

An LSB conforming implementation shall provide the generic data interfaces for Class \_\_cxxabiv1::\_\_function\_type\_info specified in [Table 16-33](#ID_TBL_45_LIBSTDCXX_45_CLASY_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-33 libstdcxx - Class \_\_cxxabiv1::\_\_function\_type\_info Data Interfaces**

|  |
| --- |
| typeinfo for \_\_cxxabiv1::\_\_function\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for \_\_cxxabiv1::\_\_function\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for \_\_cxxabiv1::\_\_function\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.11 Class \_\_cxxabiv1::\_\_si\_class\_type\_info**

#### 16.1.11.1 Class data for \_\_cxxabiv1::\_\_si\_class\_type\_info

The virtual table for the \_\_cxxabiv1::\_\_si\_class\_type\_info class is described by [Table 16-34](#ID_CLS_45_23979_45_0)

**Table 16-34 Primary vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| vfunc[0]: | \_\_cxxabiv1::\_\_si\_class\_type\_info::~\_\_si\_class\_type\_info() |
| vfunc[1]: | \_\_cxxabiv1::\_\_si\_class\_type\_info::~\_\_si\_class\_type\_info() |
| vfunc[2]: | type\_info::\_\_is\_pointer\_p() const |
| vfunc[3]: | type\_info::\_\_is\_function\_p() const |
| vfunc[4]: | \_\_cxxabiv1::\_\_class\_type\_info::\_\_do\_catch(type\_info const\*, void\*\*, unsigned int) const |
| vfunc[5]: | \_\_cxxabiv1::\_\_class\_type\_info::\_\_do\_upcast(\_\_cxxabiv1::\_\_class\_type\_info const\*, void\*\*) const |
| vfunc[6]: | \_\_cxxabiv1::\_\_si\_class\_type\_info::\_\_do\_upcast(\_\_cxxabiv1::\_\_class\_type\_info const\*, void const\*, \_\_cxxabiv1::\_\_class\_type\_info::\_\_upcast\_result&) const |

The Run Time Type Information for the \_\_cxxabiv1::\_\_si\_class\_type\_info class is described by [Table 16-35](#ID_RTTI_45_24094)

**Table 16-35 typeinfo for \_\_cxxabiv1::\_\_si\_class\_type\_info**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for \_\_cxxabiv1::\_\_si\_class\_type\_info |

#### 16.1.11.2 Interfaces for Class \_\_cxxabiv1::\_\_si\_class\_type\_info

An LSB conforming implementation shall provide the generic methods for Class \_\_cxxabiv1::\_\_si\_class\_type\_info specified in [Table 16-36](#ID_TBL_45_LIBSTDCXX_45_CLASZ_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-36 libstdcxx - Class \_\_cxxabiv1::\_\_si\_class\_type\_info Function Interfaces**

|  |
| --- |
| \_\_cxxabiv1::\_\_si\_class\_type\_info::~\_\_si\_class\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_si\_class\_type\_info::~\_\_si\_class\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_si\_class\_type\_info::~\_\_si\_class\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_si\_class\_type\_info::\_\_do\_upcast(\_\_cxxabiv1::\_\_class\_type\_info const\*, void const\*, \_\_cxxabiv1::\_\_class\_type\_info::\_\_upcast\_result&) const(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

An LSB conforming implementation shall provide the generic data interfaces for Class \_\_cxxabiv1::\_\_si\_class\_type\_info specified in [Table 16-37](#ID_TBL_45_LIBSTDCXX_45_CLASZ_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-37 libstdcxx - Class \_\_cxxabiv1::\_\_si\_class\_type\_info Data Interfaces**

|  |
| --- |
| typeinfo for \_\_cxxabiv1::\_\_si\_class\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for \_\_cxxabiv1::\_\_si\_class\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.12 Class \_\_cxxabiv1::\_\_vmi\_class\_type\_info**

#### 16.1.12.1 Class data for \_\_cxxabiv1::\_\_vmi\_class\_type\_info

The virtual table for the \_\_cxxabiv1::\_\_vmi\_class\_type\_info class is described by [Table 16-38](#ID_CLS_45_23980_45_0)

**Table 16-38 Primary vtable for \_\_cxxabiv1::\_\_vmi\_class\_type\_info**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for \_\_cxxabiv1::\_\_vmi\_class\_type\_info |
| vfunc[0]: | \_\_cxxabiv1::\_\_vmi\_class\_type\_info::~\_\_vmi\_class\_type\_info() |
| vfunc[1]: | \_\_cxxabiv1::\_\_vmi\_class\_type\_info::~\_\_vmi\_class\_type\_info() |
| vfunc[2]: | type\_info::\_\_is\_pointer\_p() const |
| vfunc[3]: | type\_info::\_\_is\_function\_p() const |
| vfunc[4]: | \_\_cxxabiv1::\_\_class\_type\_info::\_\_do\_catch(type\_info const\*, void\*\*, unsigned int) const |
| vfunc[5]: | \_\_cxxabiv1::\_\_class\_type\_info::\_\_do\_upcast(\_\_cxxabiv1::\_\_class\_type\_info const\*, void\*\*) const |
| vfunc[6]: | \_\_cxxabiv1::\_\_vmi\_class\_type\_info::\_\_do\_upcast(\_\_cxxabiv1::\_\_class\_type\_info const\*, void const\*, \_\_cxxabiv1::\_\_class\_type\_info::\_\_upcast\_result&) const |

The Run Time Type Information for the \_\_cxxabiv1::\_\_vmi\_class\_type\_info class is described by [Table 16-39](#ID_RTTI_45_24093)

**Table 16-39 typeinfo for \_\_cxxabiv1::\_\_vmi\_class\_type\_info**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for \_\_cxxabiv1::\_\_vmi\_class\_type\_info |

#### 16.1.12.2 Interfaces for Class \_\_cxxabiv1::\_\_vmi\_class\_type\_info

An LSB conforming implementation shall provide the generic methods for Class \_\_cxxabiv1::\_\_vmi\_class\_type\_info specified in [Table 16-40](#ID_TBL_45_LIBSTDCXX_45_CLATA_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-40 libstdcxx - Class \_\_cxxabiv1::\_\_vmi\_class\_type\_info Function Interfaces**

|  |
| --- |
| \_\_cxxabiv1::\_\_vmi\_class\_type\_info::~\_\_vmi\_class\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_vmi\_class\_type\_info::~\_\_vmi\_class\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_vmi\_class\_type\_info::~\_\_vmi\_class\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_vmi\_class\_type\_info::\_\_do\_upcast(\_\_cxxabiv1::\_\_class\_type\_info const\*, void const\*, \_\_cxxabiv1::\_\_class\_type\_info::\_\_upcast\_result&) const(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

An LSB conforming implementation shall provide the generic data interfaces for Class \_\_cxxabiv1::\_\_vmi\_class\_type\_info specified in [Table 16-41](#ID_TBL_45_LIBSTDCXX_45_CLATA_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-41 libstdcxx - Class \_\_cxxabiv1::\_\_vmi\_class\_type\_info Data Interfaces**

|  |
| --- |
| typeinfo for \_\_cxxabiv1::\_\_vmi\_class\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for \_\_cxxabiv1::\_\_vmi\_class\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for \_\_cxxabiv1::\_\_vmi\_class\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.13 Class \_\_cxxabiv1::\_\_fundamental\_type\_info**

#### 16.1.13.1 Class data for \_\_cxxabiv1::\_\_fundamental\_type\_info

The virtual table for the \_\_cxxabiv1::\_\_fundamental\_type\_info class is described by [Table 16-42](#ID_CLS_45_23982_45_0)

**Table 16-42 Primary vtable for \_\_cxxabiv1::\_\_fundamental\_type\_info**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for \_\_cxxabiv1::\_\_fundamental\_type\_info |
| vfunc[0]: | \_\_cxxabiv1::\_\_fundamental\_type\_info::~\_\_fundamental\_type\_info() |
| vfunc[1]: | \_\_cxxabiv1::\_\_fundamental\_type\_info::~\_\_fundamental\_type\_info() |
| vfunc[2]: | type\_info::\_\_is\_pointer\_p() const |
| vfunc[3]: | type\_info::\_\_is\_function\_p() const |
| vfunc[4]: | type\_info::\_\_do\_catch(type\_info const\*, void\*\*, unsigned int) const |
| vfunc[5]: | type\_info::\_\_do\_upcast(\_\_cxxabiv1::\_\_class\_type\_info const\*, void\*\*) const |

The Run Time Type Information for the \_\_cxxabiv1::\_\_fundamental\_type\_info class is described by [Table 16-43](#ID_RTTI_45_24102)

**Table 16-43 typeinfo for \_\_cxxabiv1::\_\_fundamental\_type\_info**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for \_\_cxxabiv1::\_\_fundamental\_type\_info |

#### 16.1.13.2 Interfaces for Class \_\_cxxabiv1::\_\_fundamental\_type\_info

An LSB conforming implementation shall provide the generic methods for Class \_\_cxxabiv1::\_\_fundamental\_type\_info specified in [Table 16-44](#ID_TBL_45_LIBSTDCXX_45_CLATB_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-44 libstdcxx - Class \_\_cxxabiv1::\_\_fundamental\_type\_info Function Interfaces**

|  |
| --- |
| \_\_cxxabiv1::\_\_fundamental\_type\_info::~\_\_fundamental\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_fundamental\_type\_info::~\_\_fundamental\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_fundamental\_type\_info::~\_\_fundamental\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

An LSB conforming implementation shall provide the generic data interfaces for Class \_\_cxxabiv1::\_\_fundamental\_type\_info specified in [Table 16-45](#ID_TBL_45_LIBSTDCXX_45_CLATB_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-45 libstdcxx - Class \_\_cxxabiv1::\_\_fundamental\_type\_info Data Interfaces**

|  |
| --- |
| typeinfo for \_\_cxxabiv1::\_\_fundamental\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for \_\_cxxabiv1::\_\_fundamental\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for \_\_cxxabiv1::\_\_fundamental\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.14 Class \_\_cxxabiv1::\_\_pointer\_to\_member\_type\_info**

#### 16.1.14.1 Class data for \_\_cxxabiv1::\_\_pointer\_to\_member\_type\_info

The virtual table for the \_\_cxxabiv1::\_\_pointer\_to\_member\_type\_info class is described by [Table 16-46](#ID_CLS_45_24326_45_0)

**Table 16-46 Primary vtable for \_\_cxxabiv1::\_\_pointer\_to\_member\_type\_info**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for \_\_cxxabiv1::\_\_pointer\_to\_member\_type\_info |
| vfunc[0]: | \_\_cxxabiv1::\_\_pointer\_to\_member\_type\_info::~\_\_pointer\_to\_member\_type\_info() |
| vfunc[1]: | \_\_cxxabiv1::\_\_pointer\_to\_member\_type\_info::~\_\_pointer\_to\_member\_type\_info() |
| vfunc[2]: | type\_info::\_\_is\_pointer\_p() const |
| vfunc[3]: | type\_info::\_\_is\_function\_p() const |
| vfunc[4]: | \_\_cxxabiv1::\_\_pbase\_type\_info::\_\_do\_catch(type\_info const\*, void\*\*, unsigned int) const |
| vfunc[5]: | type\_info::\_\_do\_upcast(\_\_cxxabiv1::\_\_class\_type\_info const\*, void\*\*) const |
| vfunc[6]: | \_\_cxxabiv1::\_\_pointer\_to\_member\_type\_info::\_\_pointer\_catch(\_\_cxxabiv1::\_\_pbase\_type\_info const\*, void\*\*, unsigned int) const |

The Run Time Type Information for the \_\_cxxabiv1::\_\_pointer\_to\_member\_type\_info class is described by [Table 16-47](#ID_RTTI_45_24097)

**Table 16-47 typeinfo for \_\_cxxabiv1::\_\_pointer\_to\_member\_type\_info**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for \_\_cxxabiv1::\_\_pointer\_to\_member\_type\_info |

#### 16.1.14.2 Interfaces for Class \_\_cxxabiv1::\_\_pointer\_to\_member\_type\_info

An LSB conforming implementation shall provide the generic methods for Class \_\_cxxabiv1::\_\_pointer\_to\_member\_type\_info specified in [Table 16-48](#ID_TBL_45_LIBSTDCXX_45_CLATC_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-48 libstdcxx - Class \_\_cxxabiv1::\_\_pointer\_to\_member\_type\_info Function Interfaces**

|  |
| --- |
| \_\_cxxabiv1::\_\_pointer\_to\_member\_type\_info::~\_\_pointer\_to\_member\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_pointer\_to\_member\_type\_info::~\_\_pointer\_to\_member\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_pointer\_to\_member\_type\_info::~\_\_pointer\_to\_member\_type\_info()(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_cxxabiv1::\_\_pointer\_to\_member\_type\_info::\_\_pointer\_catch(\_\_cxxabiv1::\_\_pbase\_type\_info const\*, void\*\*, unsigned int) const(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

An LSB conforming implementation shall provide the generic data interfaces for Class \_\_cxxabiv1::\_\_pointer\_to\_member\_type\_info specified in [Table 16-49](#ID_TBL_45_LIBSTDCXX_45_CLATC_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-49 libstdcxx - Class \_\_cxxabiv1::\_\_pointer\_to\_member\_type\_info Data Interfaces**

|  |
| --- |
| typeinfo for \_\_cxxabiv1::\_\_pointer\_to\_member\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for \_\_cxxabiv1::\_\_pointer\_to\_member\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for \_\_cxxabiv1::\_\_pointer\_to\_member\_type\_info(CXXABI\_1.3) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.15 Class \_\_gnu\_cxx::stdio\_filebuf<char, char\_traits<char> >**

#### 16.1.15.1 Interfaces for Class \_\_gnu\_cxx::stdio\_filebuf<char, char\_traits<char> >

No external methods are defined for libstdcxx - Class \_\_gnu\_cxx::stdio\_filebuf<char, std::char\_traits<char> > in this part of the specification. See also the relevant architecture specific part of this specification.

An LSB conforming implementation shall provide the generic data interfaces for Class \_\_gnu\_cxx::stdio\_filebuf<char, std::char\_traits<char> > specified in [Table 16-50](#ID_TBL_45_LIBSTDCXX_45_CLATD_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-50 libstdcxx - Class \_\_gnu\_cxx::stdio\_filebuf<char, char\_traits<char> > Data Interfaces**

|  |
| --- |
| typeinfo for \_\_gnu\_cxx::stdio\_filebuf<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for \_\_gnu\_cxx::stdio\_filebuf<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.16 Class \_\_gnu\_cxx::stdio\_filebuf<wchar\_t, char\_traits<wchar\_t> >**

#### 16.1.16.1 Interfaces for Class \_\_gnu\_cxx::stdio\_filebuf<wchar\_t, char\_traits<wchar\_t> >

No external methods are defined for libstdcxx - Class \_\_gnu\_cxx::stdio\_filebuf<wchar\_t, std::char\_traits<wchar\_t> > in this part of the specification. See also the relevant architecture specific part of this specification.

An LSB conforming implementation shall provide the generic data interfaces for Class \_\_gnu\_cxx::stdio\_filebuf<wchar\_t, std::char\_traits<wchar\_t> > specified in [Table 16-51](#ID_TBL_45_LIBSTDCXX_45_CLATE_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-51 libstdcxx - Class \_\_gnu\_cxx::stdio\_filebuf<wchar\_t, char\_traits<wchar\_t> > Data Interfaces**

|  |
| --- |
| typeinfo for \_\_gnu\_cxx::stdio\_filebuf<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for \_\_gnu\_cxx::stdio\_filebuf<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.17 Class \_\_gnu\_cxx::\_\_pool\_alloc\_base**

#### 16.1.17.1 Interfaces for Class \_\_gnu\_cxx::\_\_pool\_alloc\_base

An LSB conforming implementation shall provide the generic methods for Class \_\_gnu\_cxx::\_\_pool\_alloc\_base specified in [Table 16-52](#ID_TBL_45_LIBSTDCXX_45_CLATF_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-52 libstdcxx - Class \_\_gnu\_cxx::\_\_pool\_alloc\_base Function Interfaces**

|  |
| --- |
| \_\_gnu\_cxx::\_\_pool\_alloc\_base::\_M\_get\_mutex()(GLIBCXX\_3.4.2) [[LSB]](#ID_REFSTD_46_LIBSTDCXX_46_3) |

### **16.1.18 Class \_\_gnu\_cxx::stdio\_sync\_filebuf<char, char\_traits<char> >**

#### 16.1.18.1 Class data for \_\_gnu\_cxx::stdio\_sync\_filebuf<char, char\_traits<char> >

The virtual table for the \_\_gnu\_cxx::stdio\_sync\_filebuf<char, std::char\_traits<char> > class is described by [Table 16-53](#ID_CLS_45_29351_45_0)

**Table 16-53 Primary vtable for \_\_gnu\_cxx::stdio\_sync\_filebuf<char, char\_traits<char> >**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for \_\_gnu\_cxx::stdio\_sync\_filebuf<char, char\_traits<char> > |
| vfunc[0]: | \_\_gnu\_cxx::stdio\_sync\_filebuf<char, char\_traits<char> >::~stdio\_sync\_filebuf() |
| vfunc[1]: | \_\_gnu\_cxx::stdio\_sync\_filebuf<char, char\_traits<char> >::~stdio\_sync\_filebuf() |
| vfunc[2]: | basic\_streambuf<char, char\_traits<char> >::imbue(locale const&) |
| vfunc[3]: | See architecture specific part. |
| vfunc[4]: | See architecture specific part. |
| vfunc[5]: | \_\_gnu\_cxx::stdio\_sync\_filebuf<char, char\_traits<char> >::seekpos(fpos<\_\_mbstate\_t>, \_Ios\_Openmode) |
| vfunc[6]: | \_\_gnu\_cxx::stdio\_sync\_filebuf<char, char\_traits<char> >::sync() |
| vfunc[7]: | basic\_streambuf<char, char\_traits<char> >::showmanyc() |
| vfunc[8]: | See architecture specific part. |
| vfunc[9]: | \_\_gnu\_cxx::stdio\_sync\_filebuf<char, char\_traits<char> >::underflow() |
| vfunc[10]: | \_\_gnu\_cxx::stdio\_sync\_filebuf<char, char\_traits<char> >::uflow() |
| vfunc[11]: | \_\_gnu\_cxx::stdio\_sync\_filebuf<char, char\_traits<char> >::pbackfail(int) |
| vfunc[12]: | See architecture specific part. |
| vfunc[13]: | \_\_gnu\_cxx::stdio\_sync\_filebuf<char, char\_traits<char> >::overflow(int) |

#### 16.1.18.2 Interfaces for Class \_\_gnu\_cxx::stdio\_sync\_filebuf<char, char\_traits<char> >

An LSB conforming implementation shall provide the generic methods for Class \_\_gnu\_cxx::stdio\_sync\_filebuf<char, std::char\_traits<char> > specified in [Table 16-54](#ID_TBL_45_LIBSTDCXX_45_CLATG_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-54 libstdcxx - Class \_\_gnu\_cxx::stdio\_sync\_filebuf<char, char\_traits<char> > Function Interfaces**

|  |
| --- |
| \_\_gnu\_cxx::stdio\_sync\_filebuf<char, char\_traits<char> >::file()(GLIBCXX\_3.4.2) [[LSB]](#ID_REFSTD_46_LIBSTDCXX_46_3) |

An LSB conforming implementation shall provide the generic data interfaces for Class \_\_gnu\_cxx::stdio\_sync\_filebuf<char, std::char\_traits<char> > specified in [Table 16-55](#ID_TBL_45_LIBSTDCXX_45_CLATG_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-55 libstdcxx - Class \_\_gnu\_cxx::stdio\_sync\_filebuf<char, char\_traits<char> > Data Interfaces**

|  |
| --- |
| typeinfo for \_\_gnu\_cxx::stdio\_sync\_filebuf<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for \_\_gnu\_cxx::stdio\_sync\_filebuf<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for \_\_gnu\_cxx::stdio\_sync\_filebuf<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.19 Class \_\_gnu\_cxx::stdio\_sync\_filebuf<wchar\_t, char\_traits<wchar\_t> >**

#### 16.1.19.1 Class data for \_\_gnu\_cxx::stdio\_sync\_filebuf<wchar\_t, char\_traits<wchar\_t> >

The virtual table for the \_\_gnu\_cxx::stdio\_sync\_filebuf<wchar\_t, std::char\_traits<wchar\_t> > class is described by [Table 16-56](#ID_CLS_45_29211_45_0)

**Table 16-56 Primary vtable for \_\_gnu\_cxx::stdio\_sync\_filebuf<wchar\_t, char\_traits<wchar\_t> >**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for \_\_gnu\_cxx::stdio\_sync\_filebuf<wchar\_t, char\_traits<wchar\_t> > |
| vfunc[0]: | \_\_gnu\_cxx::stdio\_sync\_filebuf<wchar\_t, char\_traits<wchar\_t> >::~stdio\_sync\_filebuf() |
| vfunc[1]: | \_\_gnu\_cxx::stdio\_sync\_filebuf<wchar\_t, char\_traits<wchar\_t> >::~stdio\_sync\_filebuf() |
| vfunc[2]: | basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::imbue(locale const&) |
| vfunc[3]: | See architecture specific part. |
| vfunc[4]: | See architecture specific part. |
| vfunc[5]: | \_\_gnu\_cxx::stdio\_sync\_filebuf<wchar\_t, char\_traits<wchar\_t> >::seekpos(fpos<\_\_mbstate\_t>, \_Ios\_Openmode) |
| vfunc[6]: | \_\_gnu\_cxx::stdio\_sync\_filebuf<wchar\_t, char\_traits<wchar\_t> >::sync() |
| vfunc[7]: | basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::showmanyc() |
| vfunc[8]: | See architecture specific part. |
| vfunc[9]: | \_\_gnu\_cxx::stdio\_sync\_filebuf<wchar\_t, char\_traits<wchar\_t> >::underflow() |
| vfunc[10]: | \_\_gnu\_cxx::stdio\_sync\_filebuf<wchar\_t, char\_traits<wchar\_t> >::uflow() |
| vfunc[11]: | \_\_gnu\_cxx::stdio\_sync\_filebuf<wchar\_t, char\_traits<wchar\_t> >::pbackfail(unsigned int) |
| vfunc[12]: | See architecture specific part. |
| vfunc[13]: | \_\_gnu\_cxx::stdio\_sync\_filebuf<wchar\_t, char\_traits<wchar\_t> >::overflow(unsigned int) |

#### 16.1.19.2 Interfaces for Class \_\_gnu\_cxx::stdio\_sync\_filebuf<wchar\_t, char\_traits<wchar\_t> >

An LSB conforming implementation shall provide the generic methods for Class \_\_gnu\_cxx::stdio\_sync\_filebuf<wchar\_t, std::char\_traits<wchar\_t> > specified in [Table 16-57](#ID_TBL_45_LIBSTDCXX_45_CLATH_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-57 libstdcxx - Class \_\_gnu\_cxx::stdio\_sync\_filebuf<wchar\_t, char\_traits<wchar\_t> > Function Interfaces**

|  |
| --- |
| \_\_gnu\_cxx::stdio\_sync\_filebuf<wchar\_t, char\_traits<wchar\_t> >::file()(GLIBCXX\_3.4.2) [[LSB]](#ID_REFSTD_46_LIBSTDCXX_46_3) |

An LSB conforming implementation shall provide the generic data interfaces for Class \_\_gnu\_cxx::stdio\_sync\_filebuf<wchar\_t, std::char\_traits<wchar\_t> > specified in [Table 16-58](#ID_TBL_45_LIBSTDCXX_45_CLATH_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-58 libstdcxx - Class \_\_gnu\_cxx::stdio\_sync\_filebuf<wchar\_t, char\_traits<wchar\_t> > Data Interfaces**

|  |
| --- |
| typeinfo for \_\_gnu\_cxx::stdio\_sync\_filebuf<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for \_\_gnu\_cxx::stdio\_sync\_filebuf<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for \_\_gnu\_cxx::stdio\_sync\_filebuf<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.20 Class exception**

#### 16.1.20.1 Class data for exception

The virtual table for the std::exception class is described by [Table 16-59](#ID_CLS_45_24321_45_0)

**Table 16-59 Primary vtable for exception**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for exception |
| vfunc[0]: | exception::~exception() |
| vfunc[1]: | exception::~exception() |
| vfunc[2]: | exception::what() const |

The Run Time Type Information for the std::exception class is described by [Table 16-60](#ID_RTTI_45_24002)

**Table 16-60 typeinfo for exception**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_class\_type\_info |
| Name | typeinfo name for exception |

#### 16.1.20.2 Interfaces for Class exception

An LSB conforming implementation shall provide the generic methods for Class std::exception specified in [Table 16-61](#ID_TBL_45_LIBSTDCXX_45_CLATI_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-61 libstdcxx - Class exception Function Interfaces**

|  |
| --- |
| exception::what() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| exception::~exception()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| exception::~exception()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| exception::~exception()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::exception specified in [Table 16-62](#ID_TBL_45_LIBSTDCXX_45_CLATI_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-62 libstdcxx - Class exception Data Interfaces**

|  |
| --- |
| typeinfo for exception(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for exception(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for exception(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.21 Class bad\_typeid**

#### 16.1.21.1 Class data for bad\_typeid

The virtual table for the std::bad\_typeid class is described by [Table 16-63](#ID_CLS_45_24323_45_0)

**Table 16-63 Primary vtable for bad\_typeid**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for bad\_typeid |
| vfunc[0]: | bad\_typeid::~bad\_typeid() |
| vfunc[1]: | bad\_typeid::~bad\_typeid() |
| vfunc[2]: | exception::what() const |

The Run Time Type Information for the std::bad\_typeid class is described by [Table 16-64](#ID_RTTI_45_24095)

**Table 16-64 typeinfo for bad\_typeid**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for bad\_typeid |

#### 16.1.21.2 Interfaces for Class bad\_typeid

An LSB conforming implementation shall provide the generic methods for Class std::bad\_typeid specified in [Table 16-65](#ID_TBL_45_LIBSTDCXX_45_CLATJ_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-65 libstdcxx - Class bad\_typeid Function Interfaces**

|  |
| --- |
| bad\_typeid::~bad\_typeid()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bad\_typeid::~bad\_typeid()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bad\_typeid::~bad\_typeid()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::bad\_typeid specified in [Table 16-66](#ID_TBL_45_LIBSTDCXX_45_CLATJ_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-66 libstdcxx - Class bad\_typeid Data Interfaces**

|  |
| --- |
| typeinfo for bad\_typeid(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for bad\_typeid(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for bad\_typeid(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.22 Class logic\_error**

#### 16.1.22.1 Class data for logic\_error

The virtual table for the std::logic\_error class is described by [Table 16-67](#ID_CLS_45_24313_45_0)

**Table 16-67 Primary vtable for logic\_error**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for logic\_error |
| vfunc[0]: | logic\_error::~logic\_error() |
| vfunc[1]: | logic\_error::~logic\_error() |
| vfunc[2]: | logic\_error::what() const |

The Run Time Type Information for the std::logic\_error class is described by [Table 16-68](#ID_RTTI_45_23998)

**Table 16-68 typeinfo for logic\_error**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for logic\_error |

#### 16.1.22.2 Interfaces for Class logic\_error

An LSB conforming implementation shall provide the generic methods for Class std::logic\_error specified in [Table 16-69](#ID_TBL_45_LIBSTDCXX_45_CLATK_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-69 libstdcxx - Class logic\_error Function Interfaces**

|  |
| --- |
| logic\_error::what() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| logic\_error::logic\_error(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| logic\_error::logic\_error(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| logic\_error::~logic\_error()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| logic\_error::~logic\_error()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| logic\_error::~logic\_error()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::logic\_error specified in [Table 16-70](#ID_TBL_45_LIBSTDCXX_45_CLATK_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-70 libstdcxx - Class logic\_error Data Interfaces**

|  |
| --- |
| typeinfo for logic\_error(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for logic\_error(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for logic\_error(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.23 Class range\_error**

#### 16.1.23.1 Class data for range\_error

The virtual table for the std::range\_error class is described by [Table 16-71](#ID_CLS_45_24243_45_0)

**Table 16-71 Primary vtable for range\_error**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for range\_error |
| vfunc[0]: | range\_error::~range\_error() |
| vfunc[1]: | range\_error::~range\_error() |
| vfunc[2]: | runtime\_error::what() const |

The Run Time Type Information for the std::range\_error class is described by [Table 16-72](#ID_RTTI_45_24054)

**Table 16-72 typeinfo for range\_error**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for range\_error |

#### 16.1.23.2 Interfaces for Class range\_error

An LSB conforming implementation shall provide the generic methods for Class std::range\_error specified in [Table 16-73](#ID_TBL_45_LIBSTDCXX_45_CLATL_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-73 libstdcxx - Class range\_error Function Interfaces**

|  |
| --- |
| range\_error::range\_error(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| range\_error::range\_error(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| range\_error::~range\_error()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| range\_error::~range\_error()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::range\_error specified in [Table 16-74](#ID_TBL_45_LIBSTDCXX_45_CLATL_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-74 libstdcxx - Class range\_error Data Interfaces**

|  |
| --- |
| typeinfo for range\_error(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for range\_error(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for range\_error(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.24 Class domain\_error**

#### 16.1.24.1 Class data for domain\_error

The virtual table for the std::domain\_error class is described by [Table 16-75](#ID_CLS_45_24247_45_0)

**Table 16-75 Primary vtable for domain\_error**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for domain\_error |
| vfunc[0]: | domain\_error::~domain\_error() |
| vfunc[1]: | domain\_error::~domain\_error() |
| vfunc[2]: | logic\_error::what() const |

The Run Time Type Information for the std::domain\_error class is described by [Table 16-76](#ID_RTTI_45_24058)

**Table 16-76 typeinfo for domain\_error**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for domain\_error |

#### 16.1.24.2 Interfaces for Class domain\_error

An LSB conforming implementation shall provide the generic methods for Class std::domain\_error specified in [Table 16-77](#ID_TBL_45_LIBSTDCXX_45_CLATM_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-77 libstdcxx - Class domain\_error Function Interfaces**

|  |
| --- |
| domain\_error::domain\_error(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| domain\_error::domain\_error(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| domain\_error::~domain\_error()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| domain\_error::~domain\_error()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::domain\_error specified in [Table 16-78](#ID_TBL_45_LIBSTDCXX_45_CLATM_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-78 libstdcxx - Class domain\_error Data Interfaces**

|  |
| --- |
| typeinfo for domain\_error(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for domain\_error(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for domain\_error(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.25 Class length\_error**

#### 16.1.25.1 Class data for length\_error

The virtual table for the std::length\_error class is described by [Table 16-79](#ID_CLS_45_24245_45_0)

**Table 16-79 Primary vtable for length\_error**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for length\_error |
| vfunc[0]: | length\_error::~length\_error() |
| vfunc[1]: | length\_error::~length\_error() |
| vfunc[2]: | logic\_error::what() const |

The Run Time Type Information for the std::length\_error class is described by [Table 16-80](#ID_RTTI_45_24056)

**Table 16-80 typeinfo for length\_error**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for length\_error |

#### 16.1.25.2 Interfaces for Class length\_error

An LSB conforming implementation shall provide the generic methods for Class std::length\_error specified in [Table 16-81](#ID_TBL_45_LIBSTDCXX_45_CLATN_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-81 libstdcxx - Class length\_error Function Interfaces**

|  |
| --- |
| length\_error::length\_error(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| length\_error::length\_error(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| length\_error::~length\_error()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| length\_error::~length\_error()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::length\_error specified in [Table 16-82](#ID_TBL_45_LIBSTDCXX_45_CLATN_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-82 libstdcxx - Class length\_error Data Interfaces**

|  |
| --- |
| typeinfo for length\_error(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for length\_error(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for length\_error(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.26 Class out\_of\_range**

#### 16.1.26.1 Class data for out\_of\_range

The virtual table for the std::out\_of\_range class is described by [Table 16-83](#ID_CLS_45_24244_45_0)

**Table 16-83 Primary vtable for out\_of\_range**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for out\_of\_range |
| vfunc[0]: | out\_of\_range::~out\_of\_range() |
| vfunc[1]: | out\_of\_range::~out\_of\_range() |
| vfunc[2]: | logic\_error::what() const |

The Run Time Type Information for the std::out\_of\_range class is described by [Table 16-84](#ID_RTTI_45_24055)

**Table 16-84 typeinfo for out\_of\_range**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for out\_of\_range |

#### 16.1.26.2 Interfaces for Class out\_of\_range

An LSB conforming implementation shall provide the generic methods for Class std::out\_of\_range specified in [Table 16-85](#ID_TBL_45_LIBSTDCXX_45_CLATO_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-85 libstdcxx - Class out\_of\_range Function Interfaces**

|  |
| --- |
| out\_of\_range::out\_of\_range(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| out\_of\_range::out\_of\_range(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| out\_of\_range::~out\_of\_range()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| out\_of\_range::~out\_of\_range()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::out\_of\_range specified in [Table 16-86](#ID_TBL_45_LIBSTDCXX_45_CLATO_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-86 libstdcxx - Class out\_of\_range Data Interfaces**

|  |
| --- |
| typeinfo for out\_of\_range(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for out\_of\_range(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for out\_of\_range(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.27 Class bad\_exception**

#### 16.1.27.1 Class data for bad\_exception

The virtual table for the std::bad\_exception class is described by [Table 16-87](#ID_CLS_45_24320_45_0)

**Table 16-87 Primary vtable for bad\_exception**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for bad\_exception |
| vfunc[0]: | bad\_exception::~bad\_exception() |
| vfunc[1]: | bad\_exception::~bad\_exception() |
| vfunc[2]: | exception::what() const |

The Run Time Type Information for the std::bad\_exception class is described by [Table 16-88](#ID_RTTI_45_24091)

**Table 16-88 typeinfo for bad\_exception**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for bad\_exception |

#### 16.1.27.2 Interfaces for Class bad\_exception

An LSB conforming implementation shall provide the generic methods for Class std::bad\_exception specified in [Table 16-89](#ID_TBL_45_LIBSTDCXX_45_CLATP_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-89 libstdcxx - Class bad\_exception Function Interfaces**

|  |
| --- |
| bad\_exception::~bad\_exception()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bad\_exception::~bad\_exception()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bad\_exception::~bad\_exception()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::bad\_exception specified in [Table 16-90](#ID_TBL_45_LIBSTDCXX_45_CLATP_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-90 libstdcxx - Class bad\_exception Data Interfaces**

|  |
| --- |
| typeinfo for bad\_exception(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for bad\_exception(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for bad\_exception(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.28 Class runtime\_error**

#### 16.1.28.1 Class data for runtime\_error

The virtual table for the std::runtime\_error class is described by [Table 16-91](#ID_CLS_45_24312_45_0)

**Table 16-91 Primary vtable for runtime\_error**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for runtime\_error |
| vfunc[0]: | runtime\_error::~runtime\_error() |
| vfunc[1]: | runtime\_error::~runtime\_error() |
| vfunc[2]: | runtime\_error::what() const |

The Run Time Type Information for the std::runtime\_error class is described by [Table 16-92](#ID_RTTI_45_23999)

**Table 16-92 typeinfo for runtime\_error**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for runtime\_error |

#### 16.1.28.2 Interfaces for Class runtime\_error

An LSB conforming implementation shall provide the generic methods for Class std::runtime\_error specified in [Table 16-93](#ID_TBL_45_LIBSTDCXX_45_CLATQ_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-93 libstdcxx - Class runtime\_error Function Interfaces**

|  |
| --- |
| runtime\_error::what() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| runtime\_error::runtime\_error(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| runtime\_error::runtime\_error(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| runtime\_error::~runtime\_error()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| runtime\_error::~runtime\_error()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| runtime\_error::~runtime\_error()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::runtime\_error specified in [Table 16-94](#ID_TBL_45_LIBSTDCXX_45_CLATQ_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-94 libstdcxx - Class runtime\_error Data Interfaces**

|  |
| --- |
| typeinfo for runtime\_error(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for runtime\_error(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for runtime\_error(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.29 Class overflow\_error**

#### 16.1.29.1 Class data for overflow\_error

The virtual table for the std::overflow\_error class is described by [Table 16-95](#ID_CLS_45_24242_45_0)

**Table 16-95 Primary vtable for overflow\_error**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for overflow\_error |
| vfunc[0]: | overflow\_error::~overflow\_error() |
| vfunc[1]: | overflow\_error::~overflow\_error() |
| vfunc[2]: | runtime\_error::what() const |

The Run Time Type Information for the std::overflow\_error class is described by [Table 16-96](#ID_RTTI_45_24053)

**Table 16-96 typeinfo for overflow\_error**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for overflow\_error |

#### 16.1.29.2 Interfaces for Class overflow\_error

An LSB conforming implementation shall provide the generic methods for Class std::overflow\_error specified in [Table 16-97](#ID_TBL_45_LIBSTDCXX_45_CLATR_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-97 libstdcxx - Class overflow\_error Function Interfaces**

|  |
| --- |
| overflow\_error::overflow\_error(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| overflow\_error::overflow\_error(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| overflow\_error::~overflow\_error()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| overflow\_error::~overflow\_error()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::overflow\_error specified in [Table 16-98](#ID_TBL_45_LIBSTDCXX_45_CLATR_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-98 libstdcxx - Class overflow\_error Data Interfaces**

|  |
| --- |
| typeinfo for overflow\_error(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for overflow\_error(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for overflow\_error(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.30 Class underflow\_error**

#### 16.1.30.1 Class data for underflow\_error

The virtual table for the std::underflow\_error class is described by [Table 16-99](#ID_CLS_45_24241_45_0)

**Table 16-99 Primary vtable for underflow\_error**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for underflow\_error |
| vfunc[0]: | underflow\_error::~underflow\_error() |
| vfunc[1]: | underflow\_error::~underflow\_error() |
| vfunc[2]: | runtime\_error::what() const |

The Run Time Type Information for the std::underflow\_error class is described by [Table 16-100](#ID_RTTI_45_24052)

**Table 16-100 typeinfo for underflow\_error**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for underflow\_error |

#### 16.1.30.2 Interfaces for Class underflow\_error

An LSB conforming implementation shall provide the generic methods for Class std::underflow\_error specified in [Table 16-101](#ID_TBL_45_LIBSTDCXX_45_CLATS_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-101 libstdcxx - Class underflow\_error Function Interfaces**

|  |
| --- |
| underflow\_error::underflow\_error(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| underflow\_error::underflow\_error(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| underflow\_error::~underflow\_error()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| underflow\_error::~underflow\_error()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::underflow\_error specified in [Table 16-102](#ID_TBL_45_LIBSTDCXX_45_CLATS_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-102 libstdcxx - Class underflow\_error Data Interfaces**

|  |
| --- |
| typeinfo for underflow\_error(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for underflow\_error(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for underflow\_error(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.31 Class invalid\_argument**

#### 16.1.31.1 Class data for invalid\_argument

The virtual table for the std::invalid\_argument class is described by [Table 16-103](#ID_CLS_45_24246_45_0)

**Table 16-103 Primary vtable for invalid\_argument**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for invalid\_argument |
| vfunc[0]: | invalid\_argument::~invalid\_argument() |
| vfunc[1]: | invalid\_argument::~invalid\_argument() |
| vfunc[2]: | logic\_error::what() const |

The Run Time Type Information for the std::invalid\_argument class is described by [Table 16-104](#ID_RTTI_45_24057)

**Table 16-104 typeinfo for invalid\_argument**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for invalid\_argument |

#### 16.1.31.2 Interfaces for Class invalid\_argument

An LSB conforming implementation shall provide the generic methods for Class std::invalid\_argument specified in [Table 16-105](#ID_TBL_45_LIBSTDCXX_45_CLATT_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-105 libstdcxx - Class invalid\_argument Function Interfaces**

|  |
| --- |
| invalid\_argument::invalid\_argument(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| invalid\_argument::invalid\_argument(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| invalid\_argument::~invalid\_argument()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| invalid\_argument::~invalid\_argument()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::invalid\_argument specified in [Table 16-106](#ID_TBL_45_LIBSTDCXX_45_CLATT_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-106 libstdcxx - Class invalid\_argument Data Interfaces**

|  |
| --- |
| typeinfo for invalid\_argument(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for invalid\_argument(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for invalid\_argument(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.32 Class bad\_cast**

#### 16.1.32.1 Class data for bad\_cast

The virtual table for the std::bad\_cast class is described by [Table 16-107](#ID_CLS_45_24324_45_0)

**Table 16-107 Primary vtable for bad\_cast**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for bad\_cast |
| vfunc[0]: | bad\_cast::~bad\_cast() |
| vfunc[1]: | bad\_cast::~bad\_cast() |
| vfunc[2]: | exception::what() const |

The Run Time Type Information for the std::bad\_cast class is described by [Table 16-108](#ID_RTTI_45_24096)

**Table 16-108 typeinfo for bad\_cast**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for bad\_cast |

#### 16.1.32.2 Interfaces for Class bad\_cast

An LSB conforming implementation shall provide the generic methods for Class std::bad\_cast specified in [Table 16-109](#ID_TBL_45_LIBSTDCXX_45_CLATU_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-109 libstdcxx - Class bad\_cast Function Interfaces**

|  |
| --- |
| bad\_cast::~bad\_cast()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bad\_cast::~bad\_cast()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bad\_cast::~bad\_cast()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::bad\_cast specified in [Table 16-110](#ID_TBL_45_LIBSTDCXX_45_CLATU_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-110 libstdcxx - Class bad\_cast Data Interfaces**

|  |
| --- |
| typeinfo for bad\_cast(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for bad\_cast(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for bad\_cast(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.33 Class bad\_alloc**

#### 16.1.33.1 Class data for bad\_alloc

The virtual table for the std::bad\_alloc class is described by [Table 16-111](#ID_CLS_45_24322_45_0)

**Table 16-111 Primary vtable for bad\_alloc**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for bad\_alloc |
| vfunc[0]: | bad\_alloc::~bad\_alloc() |
| vfunc[1]: | bad\_alloc::~bad\_alloc() |
| vfunc[2]: | exception::what() const |

The Run Time Type Information for the std::bad\_alloc class is described by [Table 16-112](#ID_RTTI_45_24092)

**Table 16-112 typeinfo for bad\_alloc**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for bad\_alloc |

#### 16.1.33.2 Interfaces for Class bad\_alloc

An LSB conforming implementation shall provide the generic methods for Class std::bad\_alloc specified in [Table 16-113](#ID_TBL_45_LIBSTDCXX_45_CLATV_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-113 libstdcxx - Class bad\_alloc Function Interfaces**

|  |
| --- |
| bad\_alloc::~bad\_alloc()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bad\_alloc::~bad\_alloc()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bad\_alloc::~bad\_alloc()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::bad\_alloc specified in [Table 16-114](#ID_TBL_45_LIBSTDCXX_45_CLATV_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-114 libstdcxx - Class bad\_alloc Data Interfaces**

|  |
| --- |
| typeinfo for bad\_alloc(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for bad\_alloc(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for bad\_alloc(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.34 struct \_\_numeric\_limits\_base**

#### 16.1.34.1 Interfaces for struct \_\_numeric\_limits\_base

No external methods are defined for libstdcxx - struct \_\_numeric\_limits\_base in this part of the specification. See also the relevant architecture specific part of this specification.

An LSB conforming implementation shall provide the generic data interfaces for struct \_\_numeric\_limits\_base specified in [Table 16-115](#ID_TBL_45_LIBSTDCXX_45_STRUC_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-115 libstdcxx - struct \_\_numeric\_limits\_base Data Interfaces**

|  |
| --- |
| \_\_numeric\_limits\_base::has\_denorm(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_numeric\_limits\_base::is\_bounded(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_numeric\_limits\_base::is\_integer(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_numeric\_limits\_base::round\_style(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_numeric\_limits\_base::has\_infinity(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_numeric\_limits\_base::max\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_numeric\_limits\_base::min\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_numeric\_limits\_base::has\_quiet\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_numeric\_limits\_base::is\_specialized(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_numeric\_limits\_base::max\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_numeric\_limits\_base::min\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_numeric\_limits\_base::has\_denorm\_loss(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_numeric\_limits\_base::tinyness\_before(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_numeric\_limits\_base::has\_signaling\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_numeric\_limits\_base::radix(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_numeric\_limits\_base::traps(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_numeric\_limits\_base::digits(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_numeric\_limits\_base::digits10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_numeric\_limits\_base::is\_exact(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_numeric\_limits\_base::is\_iec559(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_numeric\_limits\_base::is\_modulo(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_numeric\_limits\_base::is\_signed(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.35 struct numeric\_limits<long double>**

#### 16.1.35.1 Interfaces for struct numeric\_limits<long double>

No external methods are defined for libstdcxx - struct numeric\_limits<long double> in this part of the specification. See also the relevant architecture specific part of this specification.

An LSB conforming implementation shall provide the generic data interfaces for struct numeric\_limits<long double> specified in [Table 16-116](#ID_TBL_45_LIBSTDCXX_45_STRUD_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-116 libstdcxx - struct numeric\_limits<long double> Data Interfaces**

|  |
| --- |
| numeric\_limits<long double>::has\_denorm(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long double>::is\_bounded(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long double>::is\_integer(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long double>::round\_style(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long double>::has\_infinity(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long double>::max\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long double>::min\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long double>::has\_quiet\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long double>::is\_specialized(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long double>::max\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long double>::min\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long double>::has\_denorm\_loss(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long double>::tinyness\_before(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long double>::has\_signaling\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long double>::radix(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long double>::traps(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long double>::digits(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long double>::digits10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long double>::is\_exact(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long double>::is\_iec559(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long double>::is\_modulo(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long double>::is\_signed(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.36 struct numeric\_limits<long long>**

#### 16.1.36.1 Interfaces for struct numeric\_limits<long long>

No external methods are defined for libstdcxx - struct numeric\_limits<long long> in this part of the specification. See also the relevant architecture specific part of this specification.

An LSB conforming implementation shall provide the generic data interfaces for struct numeric\_limits<long long> specified in [Table 16-117](#ID_TBL_45_LIBSTDCXX_45_STRUE_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-117 libstdcxx - struct numeric\_limits<long long> Data Interfaces**

|  |
| --- |
| numeric\_limits<long long>::has\_denorm(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long long>::is\_bounded(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long long>::is\_integer(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long long>::round\_style(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long long>::has\_infinity(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long long>::max\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long long>::min\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long long>::has\_quiet\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long long>::is\_specialized(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long long>::max\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long long>::min\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long long>::has\_denorm\_loss(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long long>::tinyness\_before(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long long>::has\_signaling\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long long>::radix(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long long>::traps(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long long>::digits(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long long>::digits10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long long>::is\_exact(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long long>::is\_iec559(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long long>::is\_modulo(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long long>::is\_signed(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.37 struct numeric\_limits<unsigned long long>**

#### 16.1.37.1 Interfaces for struct numeric\_limits<unsigned long long>

No external methods are defined for libstdcxx - struct numeric\_limits<unsigned long long> in this part of the specification. See also the relevant architecture specific part of this specification.

An LSB conforming implementation shall provide the generic data interfaces for struct numeric\_limits<unsigned long long> specified in [Table 16-118](#ID_TBL_45_LIBSTDCXX_45_STRUF_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-118 libstdcxx - struct numeric\_limits<unsigned long long> Data Interfaces**

|  |
| --- |
| numeric\_limits<unsigned long long>::has\_denorm(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long long>::is\_bounded(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long long>::is\_integer(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long long>::round\_style(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long long>::has\_infinity(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long long>::max\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long long>::min\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long long>::has\_quiet\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long long>::is\_specialized(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long long>::max\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long long>::min\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long long>::has\_denorm\_loss(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long long>::tinyness\_before(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long long>::has\_signaling\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long long>::radix(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long long>::traps(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long long>::digits(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long long>::digits10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long long>::is\_exact(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long long>::is\_iec559(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long long>::is\_modulo(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long long>::is\_signed(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.38 struct numeric\_limits<float>**

#### 16.1.38.1 Interfaces for struct numeric\_limits<float>

No external methods are defined for libstdcxx - struct numeric\_limits<float> in this part of the specification. See also the relevant architecture specific part of this specification.

An LSB conforming implementation shall provide the generic data interfaces for struct numeric\_limits<float> specified in [Table 16-119](#ID_TBL_45_LIBSTDCXX_45_STRUG_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-119 libstdcxx - struct numeric\_limits<float> Data Interfaces**

|  |
| --- |
| numeric\_limits<float>::has\_denorm(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<float>::is\_bounded(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<float>::is\_integer(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<float>::round\_style(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<float>::has\_infinity(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<float>::max\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<float>::min\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<float>::has\_quiet\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<float>::is\_specialized(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<float>::max\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<float>::min\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<float>::has\_denorm\_loss(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<float>::tinyness\_before(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<float>::has\_signaling\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<float>::radix(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<float>::traps(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<float>::digits(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<float>::digits10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<float>::is\_exact(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<float>::is\_iec559(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<float>::is\_modulo(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<float>::is\_signed(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.39 struct numeric\_limits<double>**

#### 16.1.39.1 Interfaces for struct numeric\_limits<double>

No external methods are defined for libstdcxx - struct numeric\_limits<double> in this part of the specification. See also the relevant architecture specific part of this specification.

An LSB conforming implementation shall provide the generic data interfaces for struct numeric\_limits<double> specified in [Table 16-120](#ID_TBL_45_LIBSTDCXX_45_STRUH_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-120 libstdcxx - struct numeric\_limits<double> Data Interfaces**

|  |
| --- |
| numeric\_limits<double>::has\_denorm(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<double>::is\_bounded(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<double>::is\_integer(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<double>::round\_style(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<double>::has\_infinity(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<double>::max\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<double>::min\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<double>::has\_quiet\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<double>::is\_specialized(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<double>::max\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<double>::min\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<double>::has\_denorm\_loss(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<double>::tinyness\_before(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<double>::has\_signaling\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<double>::radix(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<double>::traps(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<double>::digits(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<double>::digits10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<double>::is\_exact(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<double>::is\_iec559(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<double>::is\_modulo(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<double>::is\_signed(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.40 struct numeric\_limits<short>**

#### 16.1.40.1 Interfaces for struct numeric\_limits<short>

No external methods are defined for libstdcxx - struct numeric\_limits<short> in this part of the specification. See also the relevant architecture specific part of this specification.

An LSB conforming implementation shall provide the generic data interfaces for struct numeric\_limits<short> specified in [Table 16-121](#ID_TBL_45_LIBSTDCXX_45_STRUI_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-121 libstdcxx - struct numeric\_limits<short> Data Interfaces**

|  |
| --- |
| numeric\_limits<short>::has\_denorm(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<short>::is\_bounded(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<short>::is\_integer(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<short>::round\_style(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<short>::has\_infinity(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<short>::max\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<short>::min\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<short>::has\_quiet\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<short>::is\_specialized(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<short>::max\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<short>::min\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<short>::has\_denorm\_loss(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<short>::tinyness\_before(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<short>::has\_signaling\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<short>::radix(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<short>::traps(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<short>::digits(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<short>::digits10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<short>::is\_exact(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<short>::is\_iec559(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<short>::is\_modulo(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<short>::is\_signed(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.41 struct numeric\_limits<unsigned short>**

#### 16.1.41.1 Interfaces for struct numeric\_limits<unsigned short>

No external methods are defined for libstdcxx - struct numeric\_limits<unsigned short> in this part of the specification. See also the relevant architecture specific part of this specification.

An LSB conforming implementation shall provide the generic data interfaces for struct numeric\_limits<unsigned short> specified in [Table 16-122](#ID_TBL_45_LIBSTDCXX_45_STRUJ_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-122 libstdcxx - struct numeric\_limits<unsigned short> Data Interfaces**

|  |
| --- |
| numeric\_limits<unsigned short>::has\_denorm(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned short>::is\_bounded(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned short>::is\_integer(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned short>::round\_style(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned short>::has\_infinity(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned short>::max\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned short>::min\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned short>::has\_quiet\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned short>::is\_specialized(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned short>::max\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned short>::min\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned short>::has\_denorm\_loss(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned short>::tinyness\_before(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned short>::has\_signaling\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned short>::radix(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned short>::traps(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned short>::digits(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned short>::digits10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned short>::is\_exact(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned short>::is\_iec559(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned short>::is\_modulo(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned short>::is\_signed(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.42 struct numeric\_limits<int>**

#### 16.1.42.1 Interfaces for struct numeric\_limits<int>

No external methods are defined for libstdcxx - struct numeric\_limits<int> in this part of the specification. See also the relevant architecture specific part of this specification.

An LSB conforming implementation shall provide the generic data interfaces for struct numeric\_limits<int> specified in [Table 16-123](#ID_TBL_45_LIBSTDCXX_45_STRUK_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-123 libstdcxx - struct numeric\_limits<int> Data Interfaces**

|  |
| --- |
| numeric\_limits<int>::has\_denorm(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<int>::is\_bounded(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<int>::is\_integer(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<int>::round\_style(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<int>::has\_infinity(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<int>::max\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<int>::min\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<int>::has\_quiet\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<int>::is\_specialized(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<int>::max\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<int>::min\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<int>::has\_denorm\_loss(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<int>::tinyness\_before(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<int>::has\_signaling\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<int>::radix(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<int>::traps(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<int>::digits(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<int>::digits10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<int>::is\_exact(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<int>::is\_iec559(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<int>::is\_modulo(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<int>::is\_signed(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.43 struct numeric\_limits<unsigned int>**

#### 16.1.43.1 Interfaces for struct numeric\_limits<unsigned int>

No external methods are defined for libstdcxx - struct numeric\_limits<unsigned int> in this part of the specification. See also the relevant architecture specific part of this specification.

An LSB conforming implementation shall provide the generic data interfaces for struct numeric\_limits<unsigned int> specified in [Table 16-124](#ID_TBL_45_LIBSTDCXX_45_STRUL_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-124 libstdcxx - struct numeric\_limits<unsigned int> Data Interfaces**

|  |
| --- |
| numeric\_limits<unsigned int>::has\_denorm(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned int>::is\_bounded(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned int>::is\_integer(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned int>::round\_style(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned int>::has\_infinity(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned int>::max\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned int>::min\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned int>::has\_quiet\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned int>::is\_specialized(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned int>::max\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned int>::min\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned int>::has\_denorm\_loss(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned int>::tinyness\_before(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned int>::has\_signaling\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned int>::radix(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned int>::traps(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned int>::digits(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned int>::digits10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned int>::is\_exact(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned int>::is\_iec559(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned int>::is\_modulo(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned int>::is\_signed(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.44 struct numeric\_limits<long>**

#### 16.1.44.1 Interfaces for struct numeric\_limits<long>

No external methods are defined for libstdcxx - struct numeric\_limits<long> in this part of the specification. See also the relevant architecture specific part of this specification.

An LSB conforming implementation shall provide the generic data interfaces for struct numeric\_limits<long> specified in [Table 16-125](#ID_TBL_45_LIBSTDCXX_45_STRUM_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-125 libstdcxx - struct numeric\_limits<long> Data Interfaces**

|  |
| --- |
| numeric\_limits<long>::has\_denorm(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long>::is\_bounded(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long>::is\_integer(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long>::round\_style(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long>::has\_infinity(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long>::max\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long>::min\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long>::has\_quiet\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long>::is\_specialized(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long>::max\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long>::min\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long>::has\_denorm\_loss(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long>::tinyness\_before(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long>::has\_signaling\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long>::radix(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long>::traps(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long>::digits(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long>::digits10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long>::is\_exact(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long>::is\_iec559(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long>::is\_modulo(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<long>::is\_signed(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.45 struct numeric\_limits<unsigned long>**

#### 16.1.45.1 Interfaces for struct numeric\_limits<unsigned long>

No external methods are defined for libstdcxx - struct numeric\_limits<unsigned long> in this part of the specification. See also the relevant architecture specific part of this specification.

An LSB conforming implementation shall provide the generic data interfaces for struct numeric\_limits<unsigned long> specified in [Table 16-126](#ID_TBL_45_LIBSTDCXX_45_STRUN_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-126 libstdcxx - struct numeric\_limits<unsigned long> Data Interfaces**

|  |
| --- |
| numeric\_limits<unsigned long>::has\_denorm(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long>::is\_bounded(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long>::is\_integer(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long>::round\_style(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long>::has\_infinity(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long>::max\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long>::min\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long>::has\_quiet\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long>::is\_specialized(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long>::max\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long>::min\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long>::has\_denorm\_loss(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long>::tinyness\_before(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long>::has\_signaling\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long>::radix(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long>::traps(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long>::digits(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long>::digits10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long>::is\_exact(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long>::is\_iec559(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long>::is\_modulo(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned long>::is\_signed(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.46 struct numeric\_limits<wchar\_t>**

#### 16.1.46.1 Interfaces for struct numeric\_limits<wchar\_t>

No external methods are defined for libstdcxx - struct numeric\_limits<wchar\_t> in this part of the specification. See also the relevant architecture specific part of this specification.

An LSB conforming implementation shall provide the generic data interfaces for struct numeric\_limits<wchar\_t> specified in [Table 16-127](#ID_TBL_45_LIBSTDCXX_45_STRUO_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-127 libstdcxx - struct numeric\_limits<wchar\_t> Data Interfaces**

|  |
| --- |
| numeric\_limits<wchar\_t>::has\_denorm(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<wchar\_t>::is\_bounded(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<wchar\_t>::is\_integer(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<wchar\_t>::round\_style(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<wchar\_t>::has\_infinity(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<wchar\_t>::max\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<wchar\_t>::min\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<wchar\_t>::has\_quiet\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<wchar\_t>::is\_specialized(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<wchar\_t>::max\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<wchar\_t>::min\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<wchar\_t>::has\_denorm\_loss(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<wchar\_t>::tinyness\_before(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<wchar\_t>::has\_signaling\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<wchar\_t>::radix(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<wchar\_t>::traps(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<wchar\_t>::digits(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<wchar\_t>::digits10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<wchar\_t>::is\_exact(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<wchar\_t>::is\_iec559(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<wchar\_t>::is\_modulo(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<wchar\_t>::is\_signed(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.47 struct numeric\_limits<unsigned char>**

#### 16.1.47.1 Interfaces for struct numeric\_limits<unsigned char>

No external methods are defined for libstdcxx - struct numeric\_limits<unsigned char> in this part of the specification. See also the relevant architecture specific part of this specification.

An LSB conforming implementation shall provide the generic data interfaces for struct numeric\_limits<unsigned char> specified in [Table 16-128](#ID_TBL_45_LIBSTDCXX_45_STRUP_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-128 libstdcxx - struct numeric\_limits<unsigned char> Data Interfaces**

|  |
| --- |
| numeric\_limits<unsigned char>::has\_denorm(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned char>::is\_bounded(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned char>::is\_integer(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned char>::round\_style(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned char>::has\_infinity(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned char>::max\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned char>::min\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned char>::has\_quiet\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned char>::is\_specialized(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned char>::max\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned char>::min\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned char>::has\_denorm\_loss(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned char>::tinyness\_before(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned char>::has\_signaling\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned char>::radix(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned char>::traps(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned char>::digits(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned char>::digits10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned char>::is\_exact(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned char>::is\_iec559(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned char>::is\_modulo(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<unsigned char>::is\_signed(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.48 struct numeric\_limits<signed char>**

#### 16.1.48.1 Interfaces for struct numeric\_limits<signed char>

No external methods are defined for libstdcxx - struct numeric\_limits<signed char> in this part of the specification. See also the relevant architecture specific part of this specification.

An LSB conforming implementation shall provide the generic data interfaces for struct numeric\_limits<signed char> specified in [Table 16-129](#ID_TBL_45_LIBSTDCXX_45_STRUQ_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-129 libstdcxx - struct numeric\_limits<signed char> Data Interfaces**

|  |
| --- |
| numeric\_limits<signed char>::has\_denorm(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<signed char>::is\_bounded(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<signed char>::is\_integer(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<signed char>::round\_style(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<signed char>::has\_infinity(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<signed char>::max\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<signed char>::min\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<signed char>::has\_quiet\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<signed char>::is\_specialized(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<signed char>::max\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<signed char>::min\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<signed char>::has\_denorm\_loss(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<signed char>::tinyness\_before(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<signed char>::has\_signaling\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<signed char>::radix(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<signed char>::traps(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<signed char>::digits(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<signed char>::digits10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<signed char>::is\_exact(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<signed char>::is\_iec559(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<signed char>::is\_modulo(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<signed char>::is\_signed(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.49 struct numeric\_limits<char>**

#### 16.1.49.1 Interfaces for struct numeric\_limits<char>

No external methods are defined for libstdcxx - struct numeric\_limits<char> in this part of the specification. See also the relevant architecture specific part of this specification.

An LSB conforming implementation shall provide the generic data interfaces for struct numeric\_limits<char> specified in [Table 16-130](#ID_TBL_45_LIBSTDCXX_45_STRUR_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-130 libstdcxx - struct numeric\_limits<char> Data Interfaces**

|  |
| --- |
| numeric\_limits<char>::has\_denorm(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<char>::is\_bounded(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<char>::is\_integer(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<char>::round\_style(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<char>::has\_infinity(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<char>::max\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<char>::min\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<char>::has\_quiet\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<char>::is\_specialized(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<char>::max\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<char>::min\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<char>::has\_denorm\_loss(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<char>::tinyness\_before(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<char>::has\_signaling\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<char>::radix(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<char>::traps(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<char>::digits(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<char>::digits10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<char>::is\_exact(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<char>::is\_iec559(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<char>::is\_modulo(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<char>::is\_signed(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.50 struct numeric\_limits<bool>**

#### 16.1.50.1 Interfaces for struct numeric\_limits<bool>

No external methods are defined for libstdcxx - struct numeric\_limits<bool> in this part of the specification. See also the relevant architecture specific part of this specification.

An LSB conforming implementation shall provide the generic data interfaces for struct numeric\_limits<bool> specified in [Table 16-131](#ID_TBL_45_LIBSTDCXX_45_STRUS_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-131 libstdcxx - struct numeric\_limits<bool> Data Interfaces**

|  |
| --- |
| numeric\_limits<bool>::has\_denorm(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<bool>::is\_bounded(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<bool>::is\_integer(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<bool>::round\_style(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<bool>::has\_infinity(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<bool>::max\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<bool>::min\_exponent(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<bool>::has\_quiet\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<bool>::is\_specialized(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<bool>::max\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<bool>::min\_exponent10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<bool>::has\_denorm\_loss(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<bool>::tinyness\_before(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<bool>::has\_signaling\_NaN(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<bool>::radix(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<bool>::traps(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<bool>::digits(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<bool>::digits10(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<bool>::is\_exact(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<bool>::is\_iec559(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<bool>::is\_modulo(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numeric\_limits<bool>::is\_signed(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.51 Class ctype\_base**

#### 16.1.51.1 Class data for ctype\_base

The Run Time Type Information for the std::ctype\_base class is described by [Table 16-132](#ID_RTTI_45_24006)

**Table 16-132 typeinfo for ctype\_base**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_class\_type\_info |
| Name | typeinfo name for ctype\_base |

#### 16.1.51.2 Interfaces for Class ctype\_base

No external methods are defined for libstdcxx - Class std::ctype\_base in this part of the specification. See also the relevant architecture specific part of this specification.

An LSB conforming implementation shall provide the generic data interfaces for Class std::ctype\_base specified in [Table 16-133](#ID_TBL_45_LIBSTDCXX_45_CLATW_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-133 libstdcxx - Class ctype\_base Data Interfaces**

|  |
| --- |
| ctype\_base::alnum(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype\_base::alpha(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype\_base::cntrl(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype\_base::digit(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype\_base::graph(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype\_base::lower(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype\_base::print(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype\_base::punct(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype\_base::space(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype\_base::upper(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype\_base::xdigit(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for ctype\_base(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for ctype\_base(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.52 Class \_\_ctype\_abstract\_base<char>**

#### 16.1.52.1 Class data for \_\_ctype\_abstract\_base<char>

The virtual table for the std::\_\_ctype\_abstract\_base<char> class is described by [Table 16-134](#ID_CLS_45_24301_45_0)

**Table 16-134 Primary vtable for \_\_ctype\_abstract\_base<char>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for \_\_ctype\_abstract\_base<char> |
| vfunc[0]: | NULL or \_\_ctype\_abstract\_base<char>::~\_\_ctype\_abstract\_base() |
| vfunc[1]: | NULL or \_\_ctype\_abstract\_base<char>::~\_\_ctype\_abstract\_base() |
| vfunc[2]: | \_\_cxa\_pure\_virtual |
| vfunc[3]: | \_\_cxa\_pure\_virtual |
| vfunc[4]: | \_\_cxa\_pure\_virtual |
| vfunc[5]: | \_\_cxa\_pure\_virtual |
| vfunc[6]: | \_\_cxa\_pure\_virtual |
| vfunc[7]: | \_\_cxa\_pure\_virtual |
| vfunc[8]: | \_\_cxa\_pure\_virtual |
| vfunc[9]: | \_\_cxa\_pure\_virtual |
| vfunc[10]: | \_\_cxa\_pure\_virtual |
| vfunc[11]: | \_\_cxa\_pure\_virtual |
| vfunc[12]: | \_\_cxa\_pure\_virtual |
| vfunc[13]: | \_\_cxa\_pure\_virtual |

#### 16.1.52.2 Interfaces for Class \_\_ctype\_abstract\_base<char>

No external methods are defined for libstdcxx - Class std::\_\_ctype\_abstract\_base<char> in this part of the specification. See also the relevant architecture specific part of this specification.

An LSB conforming implementation shall provide the generic data interfaces for Class std::\_\_ctype\_abstract\_base<char> specified in [Table 16-135](#ID_TBL_45_LIBSTDCXX_45_CLATX_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-135 libstdcxx - Class \_\_ctype\_abstract\_base<char> Data Interfaces**

|  |
| --- |
| typeinfo for \_\_ctype\_abstract\_base<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for \_\_ctype\_abstract\_base<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for \_\_ctype\_abstract\_base<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.53 Class \_\_ctype\_abstract\_base<wchar\_t>**

#### 16.1.53.1 Class data for \_\_ctype\_abstract\_base<wchar\_t>

The virtual table for the std::\_\_ctype\_abstract\_base<wchar\_t> class is described by [Table 16-136](#ID_CLS_45_24300_45_0)

**Table 16-136 Primary vtable for \_\_ctype\_abstract\_base<wchar\_t>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for \_\_ctype\_abstract\_base<wchar\_t> |
| vfunc[0]: | NULL or \_\_ctype\_abstract\_base<wchar\_t>::~\_\_ctype\_abstract\_base() |
| vfunc[1]: | NULL or \_\_ctype\_abstract\_base<wchar\_t>::~\_\_ctype\_abstract\_base() |
| vfunc[2]: | \_\_cxa\_pure\_virtual |
| vfunc[3]: | \_\_cxa\_pure\_virtual |
| vfunc[4]: | \_\_cxa\_pure\_virtual |
| vfunc[5]: | \_\_cxa\_pure\_virtual |
| vfunc[6]: | \_\_cxa\_pure\_virtual |
| vfunc[7]: | \_\_cxa\_pure\_virtual |
| vfunc[8]: | \_\_cxa\_pure\_virtual |
| vfunc[9]: | \_\_cxa\_pure\_virtual |
| vfunc[10]: | \_\_cxa\_pure\_virtual |
| vfunc[11]: | \_\_cxa\_pure\_virtual |
| vfunc[12]: | \_\_cxa\_pure\_virtual |
| vfunc[13]: | \_\_cxa\_pure\_virtual |

#### 16.1.53.2 Interfaces for Class \_\_ctype\_abstract\_base<wchar\_t>

No external methods are defined for libstdcxx - Class std::\_\_ctype\_abstract\_base<wchar\_t> in this part of the specification. See also the relevant architecture specific part of this specification.

An LSB conforming implementation shall provide the generic data interfaces for Class std::\_\_ctype\_abstract\_base<wchar\_t> specified in [Table 16-137](#ID_TBL_45_LIBSTDCXX_45_CLATY_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-137 libstdcxx - Class \_\_ctype\_abstract\_base<wchar\_t> Data Interfaces**

|  |
| --- |
| typeinfo for \_\_ctype\_abstract\_base<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for \_\_ctype\_abstract\_base<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for \_\_ctype\_abstract\_base<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.54 Class ctype<char>**

#### 16.1.54.1 Class data for ctype<char>

The virtual table for the std::ctype<char> class is described by [Table 16-138](#ID_CLS_45_24230_45_0)

**Table 16-138 Primary vtable for ctype<char>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for ctype<char> |
| vfunc[0]: | ctype<char>::~ctype() |
| vfunc[1]: | ctype<char>::~ctype() |
| vfunc[2]: | ctype<char>::do\_toupper(char) const |
| vfunc[3]: | ctype<char>::do\_toupper(char\*, char const\*) const |
| vfunc[4]: | ctype<char>::do\_tolower(char) const |
| vfunc[5]: | ctype<char>::do\_tolower(char\*, char const\*) const |
| vfunc[6]: | ctype<char>::do\_widen(char) const |
| vfunc[7]: | ctype<char>::do\_widen(char const\*, char const\*, char\*) const |
| vfunc[8]: | ctype<char>::do\_narrow(char, char) const |
| vfunc[9]: | ctype<char>::do\_narrow(char const\*, char const\*, char, char\*) const |

#### 16.1.54.2 Interfaces for Class ctype<char>

An LSB conforming implementation shall provide the generic methods for Class std::ctype<char> specified in [Table 16-139](#ID_TBL_45_LIBSTDCXX_45_CLATZ_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-139 libstdcxx - Class ctype<char> Function Interfaces**

|  |
| --- |
| ctype<char>::do\_tolower(char\*, char const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<char>::do\_tolower(char) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<char>::do\_toupper(char\*, char const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<char>::do\_toupper(char) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<char>::do\_widen(char const\*, char const\*, char\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<char>::do\_widen(char) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<char>::do\_narrow(char const\*, char const\*, char, char\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<char>::do\_narrow(char, char) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<char>::classic\_table()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<char>::~ctype()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<char>::~ctype()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<char>::~ctype()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bool has\_facet<ctype<char> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::ctype<char> specified in [Table 16-140](#ID_TBL_45_LIBSTDCXX_45_CLATZ_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-140 libstdcxx - Class ctype<char> Data Interfaces**

|  |
| --- |
| ctype<char>::table\_size(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<char>::id(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for ctype<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for ctype<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for ctype<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.55 Class ctype<wchar\_t>**

#### 16.1.55.1 Class data for ctype<wchar\_t>

The virtual table for the std::ctype<wchar\_t> class is described by [Table 16-141](#ID_CLS_45_24229_45_0)

**Table 16-141 Primary vtable for ctype<wchar\_t>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for ctype<wchar\_t> |
| vfunc[0]: | ctype<wchar\_t>::~ctype() |
| vfunc[1]: | ctype<wchar\_t>::~ctype() |
| vfunc[2]: | ctype<wchar\_t>::do\_is(unsigned short, wchar\_t) const |
| vfunc[3]: | ctype<wchar\_t>::do\_is(wchar\_t const\*, wchar\_t const\*, unsigned short\*) const |
| vfunc[4]: | ctype<wchar\_t>::do\_scan\_is(unsigned short, wchar\_t const\*, wchar\_t const\*) const |
| vfunc[5]: | ctype<wchar\_t>::do\_scan\_not(unsigned short, wchar\_t const\*, wchar\_t const\*) const |
| vfunc[6]: | ctype<wchar\_t>::do\_toupper(wchar\_t) const |
| vfunc[7]: | ctype<wchar\_t>::do\_toupper(wchar\_t\*, wchar\_t const\*) const |
| vfunc[8]: | ctype<wchar\_t>::do\_tolower(wchar\_t) const |
| vfunc[9]: | ctype<wchar\_t>::do\_tolower(wchar\_t\*, wchar\_t const\*) const |
| vfunc[10]: | ctype<wchar\_t>::do\_widen(char) const |
| vfunc[11]: | ctype<wchar\_t>::do\_widen(char const\*, char const\*, wchar\_t\*) const |
| vfunc[12]: | ctype<wchar\_t>::do\_narrow(wchar\_t, char) const |
| vfunc[13]: | ctype<wchar\_t>::do\_narrow(wchar\_t const\*, wchar\_t const\*, char, char\*) const |

The Run Time Type Information for the std::ctype<wchar\_t> class is described by [Table 16-142](#ID_RTTI_45_24008)

**Table 16-142 typeinfo for ctype<wchar\_t>**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for ctype<wchar\_t> |

#### 16.1.55.2 Interfaces for Class ctype<wchar\_t>

An LSB conforming implementation shall provide the generic methods for Class std::ctype<wchar\_t> specified in [Table 16-143](#ID_TBL_45_LIBSTDCXX_45_CLAUA_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-143 libstdcxx - Class ctype<wchar\_t> Function Interfaces**

|  |
| --- |
| ctype<wchar\_t>::do\_scan\_is(unsigned short, wchar\_t const\*, wchar\_t const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<wchar\_t>::do\_tolower(wchar\_t\*, wchar\_t const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<wchar\_t>::do\_tolower(wchar\_t) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<wchar\_t>::do\_toupper(wchar\_t\*, wchar\_t const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<wchar\_t>::do\_toupper(wchar\_t) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<wchar\_t>::do\_scan\_not(unsigned short, wchar\_t const\*, wchar\_t const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<wchar\_t>::\_M\_convert\_to\_wmask(unsigned short) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<wchar\_t>::do\_is(wchar\_t const\*, wchar\_t const\*, unsigned short\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<wchar\_t>::do\_is(unsigned short, wchar\_t) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<wchar\_t>::do\_widen(char const\*, char const\*, wchar\_t\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<wchar\_t>::do\_widen(char) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<wchar\_t>::do\_narrow(wchar\_t const\*, wchar\_t const\*, char, char\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<wchar\_t>::do\_narrow(wchar\_t, char) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<wchar\_t>::\_M\_initialize\_ctype()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<wchar\_t>::~ctype()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<wchar\_t>::~ctype()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<wchar\_t>::~ctype()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::ctype<wchar\_t> specified in [Table 16-144](#ID_TBL_45_LIBSTDCXX_45_CLAUA_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-144 libstdcxx - Class ctype<wchar\_t> Data Interfaces**

|  |
| --- |
| ctype<wchar\_t>::id(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for ctype<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for ctype<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for ctype<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.56 Class ctype\_byname<char>**

#### 16.1.56.1 Class data for ctype\_byname<char>

The virtual table for the std::ctype\_byname<char> class is described by [Table 16-145](#ID_CLS_45_24299_45_0)

**Table 16-145 Primary vtable for ctype\_byname<char>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for ctype\_byname<char> |
| vfunc[0]: | ctype\_byname<char>::~ctype\_byname() |
| vfunc[1]: | ctype\_byname<char>::~ctype\_byname() |
| vfunc[2]: | ctype<char>::do\_toupper(char) const |
| vfunc[3]: | ctype<char>::do\_toupper(char\*, char const\*) const |
| vfunc[4]: | ctype<char>::do\_tolower(char) const |
| vfunc[5]: | ctype<char>::do\_tolower(char\*, char const\*) const |
| vfunc[6]: | ctype<char>::do\_widen(char) const |
| vfunc[7]: | ctype<char>::do\_widen(char const\*, char const\*, char\*) const |
| vfunc[8]: | ctype<char>::do\_narrow(char, char) const |
| vfunc[9]: | ctype<char>::do\_narrow(char const\*, char const\*, char, char\*) const |

The Run Time Type Information for the std::ctype\_byname<char> class is described by [Table 16-146](#ID_RTTI_45_24087)

**Table 16-146 typeinfo for ctype\_byname<char>**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for ctype\_byname<char> |

#### 16.1.56.2 Interfaces for Class ctype\_byname<char>

An LSB conforming implementation shall provide the generic methods for Class std::ctype\_byname<char> specified in [Table 16-147](#ID_TBL_45_LIBSTDCXX_45_CLAUB_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-147 libstdcxx - Class ctype\_byname<char> Function Interfaces**

|  |
| --- |
| ctype\_byname<char>::~ctype\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype\_byname<char>::~ctype\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype\_byname<char>::~ctype\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::ctype\_byname<char> specified in [Table 16-148](#ID_TBL_45_LIBSTDCXX_45_CLAUB_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-148 libstdcxx - Class ctype\_byname<char> Data Interfaces**

|  |
| --- |
| typeinfo for ctype\_byname<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for ctype\_byname<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for ctype\_byname<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.57 Class ctype\_byname<wchar\_t>**

#### 16.1.57.1 Class data for ctype\_byname<wchar\_t>

The virtual table for the std::ctype\_byname<wchar\_t> class is described by [Table 16-149](#ID_CLS_45_24298_45_0)

**Table 16-149 Primary vtable for ctype\_byname<wchar\_t>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for ctype\_byname<wchar\_t> |
| vfunc[0]: | ctype\_byname<wchar\_t>::~ctype\_byname() |
| vfunc[1]: | ctype\_byname<wchar\_t>::~ctype\_byname() |
| vfunc[2]: | ctype<wchar\_t>::do\_is(unsigned short, wchar\_t) const |
| vfunc[3]: | ctype<wchar\_t>::do\_is(wchar\_t const\*, wchar\_t const\*, unsigned short\*) const |
| vfunc[4]: | ctype<wchar\_t>::do\_scan\_is(unsigned short, wchar\_t const\*, wchar\_t const\*) const |
| vfunc[5]: | ctype<wchar\_t>::do\_scan\_not(unsigned short, wchar\_t const\*, wchar\_t const\*) const |
| vfunc[6]: | ctype<wchar\_t>::do\_toupper(wchar\_t) const |
| vfunc[7]: | ctype<wchar\_t>::do\_toupper(wchar\_t\*, wchar\_t const\*) const |
| vfunc[8]: | ctype<wchar\_t>::do\_tolower(wchar\_t) const |
| vfunc[9]: | ctype<wchar\_t>::do\_tolower(wchar\_t\*, wchar\_t const\*) const |
| vfunc[10]: | ctype<wchar\_t>::do\_widen(char) const |
| vfunc[11]: | ctype<wchar\_t>::do\_widen(char const\*, char const\*, wchar\_t\*) const |
| vfunc[12]: | ctype<wchar\_t>::do\_narrow(wchar\_t, char) const |
| vfunc[13]: | ctype<wchar\_t>::do\_narrow(wchar\_t const\*, wchar\_t const\*, char, char\*) const |

The Run Time Type Information for the std::ctype\_byname<wchar\_t> class is described by [Table 16-150](#ID_RTTI_45_24086)

**Table 16-150 typeinfo for ctype\_byname<wchar\_t>**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for ctype\_byname<wchar\_t> |

#### 16.1.57.2 Interfaces for Class ctype\_byname<wchar\_t>

An LSB conforming implementation shall provide the generic methods for Class std::ctype\_byname<wchar\_t> specified in [Table 16-151](#ID_TBL_45_LIBSTDCXX_45_CLAUC_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-151 libstdcxx - Class ctype\_byname<wchar\_t> Function Interfaces**

|  |
| --- |
| ctype\_byname<wchar\_t>::~ctype\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype\_byname<wchar\_t>::~ctype\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype\_byname<wchar\_t>::~ctype\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::ctype\_byname<wchar\_t> specified in [Table 16-152](#ID_TBL_45_LIBSTDCXX_45_CLAUC_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-152 libstdcxx - Class ctype\_byname<wchar\_t> Data Interfaces**

|  |
| --- |
| typeinfo for ctype\_byname<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for ctype\_byname<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for ctype\_byname<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.58 Class basic\_string<char, char\_traits<char>, allocator<char> >**

#### 16.1.58.1 Interfaces for Class basic\_string<char, char\_traits<char>, allocator<char> >

An LSB conforming implementation shall provide the generic methods for Class std::basic\_string<char, std::char\_traits<char>, std::allocator<char> > specified in [Table 16-153](#ID_TBL_45_LIBSTDCXX_45_CLAUD_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-153 libstdcxx - Class basic\_string<char, char\_traits<char>, allocator<char> > Function Interfaces**

|  |
| --- |
| basic\_string<char, char\_traits<char>, allocator<char> >::\_M\_disjunct(char const\*) const(GLIBCXX\_3.4.5) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::get\_allocator() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::end() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::\_Rep::\_M\_is\_leaked() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::\_Rep::\_M\_is\_shared() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::data() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::rend() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::size() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::begin() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::c\_str() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::empty() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::\_M\_rep() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::length() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::rbegin() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::\_M\_data() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::\_M\_iend() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::compare(char const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::compare(basic\_string<char, char\_traits<char>, allocator<char> > const&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::capacity() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::max\_size() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::\_M\_ibegin() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::\_Alloc\_hider::\_Alloc\_hider(char\*, allocator<char> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::\_Alloc\_hider::\_Alloc\_hider(char\*, allocator<char> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::\_M\_leak\_hard()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::\_S\_empty\_rep()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::\_S\_copy\_chars(char\*, \_\_gnu\_cxx::\_\_normal\_iterator<char const\*, basic\_string<char, char\_traits<char>, allocator<char> > >, \_\_gnu\_cxx::\_\_normal\_iterator<char const\*, basic\_string<char, char\_traits<char>, allocator<char> > >)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::\_S\_copy\_chars(char\*, \_\_gnu\_cxx::\_\_normal\_iterator<char\*, basic\_string<char, char\_traits<char>, allocator<char> > >, \_\_gnu\_cxx::\_\_normal\_iterator<char\*, basic\_string<char, char\_traits<char>, allocator<char> > >)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::\_S\_copy\_chars(char\*, char const\*, char const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::\_S\_copy\_chars(char\*, char\*, char\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::end()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::\_Rep::\_M\_destroy(allocator<char> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::\_Rep::\_M\_dispose(allocator<char> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::\_Rep::\_M\_refcopy()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::\_Rep::\_M\_refdata()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::\_Rep::\_S\_empty\_rep()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::\_Rep::\_M\_set\_leaked()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::\_Rep::\_M\_set\_sharable()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::\_Rep::\_M\_grab(allocator<char> const&, allocator<char> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::rend()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::swap(basic\_string<char, char\_traits<char>, allocator<char> >&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::begin()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::clear()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::erase(\_\_gnu\_cxx::\_\_normal\_iterator<char\*, basic\_string<char, char\_traits<char>, allocator<char> > >)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::erase(\_\_gnu\_cxx::\_\_normal\_iterator<char\*, basic\_string<char, char\_traits<char>, allocator<char> > >, \_\_gnu\_cxx::\_\_normal\_iterator<char\*, basic\_string<char, char\_traits<char>, allocator<char> > >)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::append(char const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::append(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::assign(char const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::assign(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::insert(\_\_gnu\_cxx::\_\_normal\_iterator<char\*, basic\_string<char, char\_traits<char>, allocator<char> > >, char)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::rbegin()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::\_M\_data(char\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::\_M\_leak()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::replace(\_\_gnu\_cxx::\_\_normal\_iterator<char\*, basic\_string<char, char\_traits<char>, allocator<char> > >, \_\_gnu\_cxx::\_\_normal\_iterator<char\*, basic\_string<char, char\_traits<char>, allocator<char> > >, \_\_gnu\_cxx::\_\_normal\_iterator<char const\*, basic\_string<char, char\_traits<char>, allocator<char> > >, \_\_gnu\_cxx::\_\_normal\_iterator<char const\*, basic\_string<char, char\_traits<char>, allocator<char> > >)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::replace(\_\_gnu\_cxx::\_\_normal\_iterator<char\*, basic\_string<char, char\_traits<char>, allocator<char> > >, \_\_gnu\_cxx::\_\_normal\_iterator<char\*, basic\_string<char, char\_traits<char>, allocator<char> > >, char const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::replace(\_\_gnu\_cxx::\_\_normal\_iterator<char\*, basic\_string<char, char\_traits<char>, allocator<char> > >, \_\_gnu\_cxx::\_\_normal\_iterator<char\*, basic\_string<char, char\_traits<char>, allocator<char> > >, char const\*, char const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::replace(\_\_gnu\_cxx::\_\_normal\_iterator<char\*, basic\_string<char, char\_traits<char>, allocator<char> > >, \_\_gnu\_cxx::\_\_normal\_iterator<char\*, basic\_string<char, char\_traits<char>, allocator<char> > >, basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::replace(\_\_gnu\_cxx::\_\_normal\_iterator<char\*, basic\_string<char, char\_traits<char>, allocator<char> > >, \_\_gnu\_cxx::\_\_normal\_iterator<char\*, basic\_string<char, char\_traits<char>, allocator<char> > >, char\*, char\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::replace(\_\_gnu\_cxx::\_\_normal\_iterator<char\*, basic\_string<char, char\_traits<char>, allocator<char> > >, \_\_gnu\_cxx::\_\_normal\_iterator<char\*, basic\_string<char, char\_traits<char>, allocator<char> > >, \_\_gnu\_cxx::\_\_normal\_iterator<char\*, basic\_string<char, char\_traits<char>, allocator<char> > >, \_\_gnu\_cxx::\_\_normal\_iterator<char\*, basic\_string<char, char\_traits<char>, allocator<char> > >)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::push\_back(char)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::basic\_string(char const\*, allocator<char> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::basic\_string(allocator<char> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::basic\_string(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::basic\_string()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::basic\_string<\_\_gnu\_cxx::\_\_normal\_iterator<char\*, basic\_string<char, char\_traits<char>, allocator<char> > > >(\_\_gnu\_cxx::\_\_normal\_iterator<char\*, basic\_string<char, char\_traits<char>, allocator<char> > >, \_\_gnu\_cxx::\_\_normal\_iterator<char\*, basic\_string<char, char\_traits<char>, allocator<char> > >, allocator<char> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::basic\_string<char const\*>(char const\*, char const\*, allocator<char> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::basic\_string<char\*>(char\*, char\*, allocator<char> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::basic\_string(char const\*, allocator<char> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::basic\_string(allocator<char> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::basic\_string(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::basic\_string()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::basic\_string<\_\_gnu\_cxx::\_\_normal\_iterator<char\*, basic\_string<char, char\_traits<char>, allocator<char> > > >(\_\_gnu\_cxx::\_\_normal\_iterator<char\*, basic\_string<char, char\_traits<char>, allocator<char> > >, \_\_gnu\_cxx::\_\_normal\_iterator<char\*, basic\_string<char, char\_traits<char>, allocator<char> > >, allocator<char> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::basic\_string<char const\*>(char const\*, char const\*, allocator<char> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::basic\_string<char\*>(char\*, char\*, allocator<char> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::~basic\_string()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::~basic\_string()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::operator=(char const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::operator=(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::operator=(char)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::operator+=(char const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::operator+=(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::operator+=(char)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::basic\_string<char, std::char\_traits<char>, std::allocator<char> > specified in [Table 16-154](#ID_TBL_45_LIBSTDCXX_45_CLAUD_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-154 libstdcxx - Class basic\_string<char, char\_traits<char>, allocator<char> > Data Interfaces**

|  |
| --- |
| basic\_string<char, char\_traits<char>, allocator<char> >::\_Rep::\_S\_max\_size(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::\_Rep::\_S\_terminal(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::\_Rep::\_S\_empty\_rep\_storage(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> >::npos(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.59 Class basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >**

#### 16.1.59.1 Interfaces for Class basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >

An LSB conforming implementation shall provide the generic methods for Class std::basic\_string<wchar\_t, std::char\_traits<wchar\_t>, std::allocator<wchar\_t> > specified in [Table 16-155](#ID_TBL_45_LIBSTDCXX_45_CLAUE_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-155 libstdcxx - Class basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > Function Interfaces**

|  |
| --- |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_M\_disjunct(wchar\_t const\*) const(GLIBCXX\_3.4.5) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::get\_allocator() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::end() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_Rep::\_M\_is\_leaked() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_Rep::\_M\_is\_shared() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::data() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::rend() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::size() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::begin() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::c\_str() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::empty() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_M\_rep() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::length() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::rbegin() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_M\_data() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_M\_iend() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::compare(wchar\_t const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::compare(basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::capacity() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::max\_size() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_M\_ibegin() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_Alloc\_hider::\_Alloc\_hider(wchar\_t\*, allocator<wchar\_t> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_Alloc\_hider::\_Alloc\_hider(wchar\_t\*, allocator<wchar\_t> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_M\_leak\_hard()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_S\_empty\_rep()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_S\_copy\_chars(wchar\_t\*, \_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t const\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > >, \_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t const\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > >)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_S\_copy\_chars(wchar\_t\*, \_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > >, \_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > >)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_S\_copy\_chars(wchar\_t\*, wchar\_t const\*, wchar\_t const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_S\_copy\_chars(wchar\_t\*, wchar\_t\*, wchar\_t\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::end()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_Rep::\_M\_destroy(allocator<wchar\_t> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_Rep::\_M\_dispose(allocator<wchar\_t> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_Rep::\_M\_refcopy()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_Rep::\_M\_refdata()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_Rep::\_S\_empty\_rep()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_Rep::\_M\_set\_leaked()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_Rep::\_M\_set\_sharable()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_Rep::\_M\_grab(allocator<wchar\_t> const&, allocator<wchar\_t> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::rend()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::swap(basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::begin()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::clear()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::erase(\_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > >)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::erase(\_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > >, \_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > >)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::append(wchar\_t const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::append(basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::assign(wchar\_t const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::assign(basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::insert(\_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > >, wchar\_t)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::rbegin()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_M\_data(wchar\_t\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_M\_leak()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::replace(\_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > >, \_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > >, \_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t const\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > >, \_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t const\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > >)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::replace(\_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > >, \_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > >, wchar\_t const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::replace(\_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > >, \_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > >, wchar\_t const\*, wchar\_t const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::replace(\_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > >, \_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > >, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::replace(\_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > >, \_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > >, wchar\_t\*, wchar\_t\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::replace(\_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > >, \_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > >, \_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > >, \_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > >)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::push\_back(wchar\_t)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_string(wchar\_t const\*, allocator<wchar\_t> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_string(allocator<wchar\_t> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_string(basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_string()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_string<\_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > > >(\_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > >, \_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > >, allocator<wchar\_t> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_string<wchar\_t const\*>(wchar\_t const\*, wchar\_t const\*, allocator<wchar\_t> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_string<wchar\_t\*>(wchar\_t\*, wchar\_t\*, allocator<wchar\_t> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_string(wchar\_t const\*, allocator<wchar\_t> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_string(allocator<wchar\_t> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_string(basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_string()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_string<\_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > > >(\_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > >, \_\_gnu\_cxx::\_\_normal\_iterator<wchar\_t\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > >, allocator<wchar\_t> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_string<wchar\_t const\*>(wchar\_t const\*, wchar\_t const\*, allocator<wchar\_t> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_string<wchar\_t\*>(wchar\_t\*, wchar\_t\*, allocator<wchar\_t> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::~basic\_string()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::~basic\_string()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::operator=(wchar\_t const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::operator=(basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::operator=(wchar\_t)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::operator+=(wchar\_t const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::operator+=(basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::operator+=(wchar\_t)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> > operator+<char, char\_traits<char>, allocator<char> >(char const\*, basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> > operator+<char, char\_traits<char>, allocator<char> >(basic\_string<char, char\_traits<char>, allocator<char> > const&, basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<char, char\_traits<char>, allocator<char> > operator+<char, char\_traits<char>, allocator<char> >(char, basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > operator+<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >(wchar\_t const\*, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > operator+<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >(basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > operator+<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >(wchar\_t, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::basic\_string<wchar\_t, std::char\_traits<wchar\_t>, std::allocator<wchar\_t> > specified in [Table 16-156](#ID_TBL_45_LIBSTDCXX_45_CLAUE_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-156 libstdcxx - Class basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > Data Interfaces**

|  |
| --- |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_Rep::\_S\_max\_size(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_Rep::\_S\_terminal(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_Rep::\_S\_empty\_rep\_storage(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::npos(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.60 Class basic\_stringstream<char, char\_traits<char>, allocator<char> >**

#### 16.1.60.1 Class data for basic\_stringstream<char, char\_traits<char>, allocator<char> >

The virtual table for the std::basic\_stringstream<char, std::char\_traits<char>, std::allocator<char> > class is described in the relevant architecture specific part of this specification.

The VTT for the std::basic\_stringstream<char, std::char\_traits<char>, std::allocator<char> > class is described by [Table 16-157](#ID_VTT_45_24742)

**Table 16-157 VTT for basic\_stringstream<char, char\_traits<char>, allocator<char> >**

|  |  |
| --- | --- |
| VTT Name | \_ZTTSt18basic\_stringstreamIcSt11char\_traitsIcESaIcEE |
| Number of Entries | 10 |

#### 16.1.60.2 Interfaces for Class basic\_stringstream<char, char\_traits<char>, allocator<char> >

An LSB conforming implementation shall provide the generic methods for Class std::basic\_stringstream<char, std::char\_traits<char>, std::allocator<char> > specified in [Table 16-158](#ID_TBL_45_LIBSTDCXX_45_CLAUF_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-158 libstdcxx - Class basic\_stringstream<char, char\_traits<char>, allocator<char> > Function Interfaces**

|  |
| --- |
| basic\_stringstream<char, char\_traits<char>, allocator<char> >::str() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringstream<char, char\_traits<char>, allocator<char> >::rdbuf() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringstream<char, char\_traits<char>, allocator<char> >::str(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringstream<char, char\_traits<char>, allocator<char> >::basic\_stringstream(basic\_string<char, char\_traits<char>, allocator<char> > const&, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringstream<char, char\_traits<char>, allocator<char> >::basic\_stringstream(\_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringstream<char, char\_traits<char>, allocator<char> >::basic\_stringstream(basic\_string<char, char\_traits<char>, allocator<char> > const&, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringstream<char, char\_traits<char>, allocator<char> >::basic\_stringstream(\_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringstream<char, char\_traits<char>, allocator<char> >::~basic\_stringstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringstream<char, char\_traits<char>, allocator<char> >::~basic\_stringstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringstream<char, char\_traits<char>, allocator<char> >::~basic\_stringstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::basic\_stringstream<char, std::char\_traits<char>, std::allocator<char> > specified in [Table 16-159](#ID_TBL_45_LIBSTDCXX_45_CLAUF_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-159 libstdcxx - Class basic\_stringstream<char, char\_traits<char>, allocator<char> > Data Interfaces**

|  |
| --- |
| typeinfo for basic\_stringstream<char, char\_traits<char>, allocator<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for basic\_stringstream<char, char\_traits<char>, allocator<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| VTT for basic\_stringstream<char, char\_traits<char>, allocator<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for basic\_stringstream<char, char\_traits<char>, allocator<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.61 Class basic\_stringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >**

#### 16.1.61.1 Class data for basic\_stringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >

The virtual table for the std::basic\_stringstream<wchar\_t, std::char\_traits<wchar\_t>, std::allocator<wchar\_t> > class is described in the relevant architecture specific part of this specification.

The VTT for the std::basic\_stringstream<wchar\_t, std::char\_traits<wchar\_t>, std::allocator<wchar\_t> > class is described by [Table 16-160](#ID_VTT_45_24743)

**Table 16-160 VTT for basic\_stringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >**

|  |  |
| --- | --- |
| VTT Name | \_ZTTSt18basic\_stringstreamIwSt11char\_traitsIwESaIwEE |
| Number of Entries | 10 |

#### 16.1.61.2 Interfaces for Class basic\_stringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >

An LSB conforming implementation shall provide the generic methods for Class std::basic\_stringstream<wchar\_t, std::char\_traits<wchar\_t>, std::allocator<wchar\_t> > specified in [Table 16-161](#ID_TBL_45_LIBSTDCXX_45_CLAUG_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-161 libstdcxx - Class basic\_stringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > Function Interfaces**

|  |
| --- |
| basic\_stringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::str() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::rdbuf() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::str(basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_stringstream(basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_stringstream(\_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_stringstream(basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_stringstream(\_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::~basic\_stringstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::~basic\_stringstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::~basic\_stringstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::basic\_stringstream<wchar\_t, std::char\_traits<wchar\_t>, std::allocator<wchar\_t> > specified in [Table 16-162](#ID_TBL_45_LIBSTDCXX_45_CLAUG_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-162 libstdcxx - Class basic\_stringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > Data Interfaces**

|  |
| --- |
| typeinfo for basic\_stringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for basic\_stringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| VTT for basic\_stringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for basic\_stringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.62 Class basic\_istringstream<char, char\_traits<char>, allocator<char> >**

#### 16.1.62.1 Class data for basic\_istringstream<char, char\_traits<char>, allocator<char> >

The virtual table for the std::basic\_istringstream<char, std::char\_traits<char>, std::allocator<char> > class is described in the relevant architecture specific part of this specification.

The VTT for the std::basic\_istringstream<char, std::char\_traits<char>, std::allocator<char> > class is described by [Table 16-163](#ID_VTT_45_24394)

**Table 16-163 VTT for basic\_istringstream<char, char\_traits<char>, allocator<char> >**

|  |  |
| --- | --- |
| VTT Name | \_ZTTSt19basic\_istringstreamIcSt11char\_traitsIcESaIcEE |
| Number of Entries | 4 |

#### 16.1.62.2 Interfaces for Class basic\_istringstream<char, char\_traits<char>, allocator<char> >

An LSB conforming implementation shall provide the generic methods for Class std::basic\_istringstream<char, std::char\_traits<char>, std::allocator<char> > specified in [Table 16-164](#ID_TBL_45_LIBSTDCXX_45_CLAUH_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-164 libstdcxx - Class basic\_istringstream<char, char\_traits<char>, allocator<char> > Function Interfaces**

|  |
| --- |
| basic\_istringstream<char, char\_traits<char>, allocator<char> >::str() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istringstream<char, char\_traits<char>, allocator<char> >::rdbuf() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istringstream<char, char\_traits<char>, allocator<char> >::str(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istringstream<char, char\_traits<char>, allocator<char> >::basic\_istringstream(basic\_string<char, char\_traits<char>, allocator<char> > const&, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istringstream<char, char\_traits<char>, allocator<char> >::basic\_istringstream(\_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istringstream<char, char\_traits<char>, allocator<char> >::basic\_istringstream(basic\_string<char, char\_traits<char>, allocator<char> > const&, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istringstream<char, char\_traits<char>, allocator<char> >::basic\_istringstream(\_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istringstream<char, char\_traits<char>, allocator<char> >::~basic\_istringstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istringstream<char, char\_traits<char>, allocator<char> >::~basic\_istringstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istringstream<char, char\_traits<char>, allocator<char> >::~basic\_istringstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::basic\_istringstream<char, std::char\_traits<char>, std::allocator<char> > specified in [Table 16-165](#ID_TBL_45_LIBSTDCXX_45_CLAUH_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-165 libstdcxx - Class basic\_istringstream<char, char\_traits<char>, allocator<char> > Data Interfaces**

|  |
| --- |
| typeinfo for basic\_istringstream<char, char\_traits<char>, allocator<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for basic\_istringstream<char, char\_traits<char>, allocator<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| VTT for basic\_istringstream<char, char\_traits<char>, allocator<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for basic\_istringstream<char, char\_traits<char>, allocator<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.63 Class basic\_istringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >**

#### 16.1.63.1 Class data for basic\_istringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >

The virtual table for the std::basic\_istringstream<wchar\_t, std::char\_traits<wchar\_t>, std::allocator<wchar\_t> > class is described in the relevant architecture specific part of this specification.

The VTT for the std::basic\_istringstream<wchar\_t, std::char\_traits<wchar\_t>, std::allocator<wchar\_t> > class is described by [Table 16-166](#ID_VTT_45_24395)

**Table 16-166 VTT for basic\_istringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >**

|  |  |
| --- | --- |
| VTT Name | \_ZTTSt19basic\_istringstreamIwSt11char\_traitsIwESaIwEE |
| Number of Entries | 4 |

#### 16.1.63.2 Interfaces for Class basic\_istringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >

An LSB conforming implementation shall provide the generic methods for Class std::basic\_istringstream<wchar\_t, std::char\_traits<wchar\_t>, std::allocator<wchar\_t> > specified in [Table 16-167](#ID_TBL_45_LIBSTDCXX_45_CLAUI_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-167 libstdcxx - Class basic\_istringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > Function Interfaces**

|  |
| --- |
| basic\_istringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::str() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::rdbuf() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::str(basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_istringstream(basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_istringstream(\_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_istringstream(basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_istringstream(\_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::~basic\_istringstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::~basic\_istringstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::~basic\_istringstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::basic\_istringstream<wchar\_t, std::char\_traits<wchar\_t>, std::allocator<wchar\_t> > specified in [Table 16-168](#ID_TBL_45_LIBSTDCXX_45_CLAUI_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-168 libstdcxx - Class basic\_istringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > Data Interfaces**

|  |
| --- |
| typeinfo for basic\_istringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for basic\_istringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| VTT for basic\_istringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for basic\_istringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.64 Class basic\_ostringstream<char, char\_traits<char>, allocator<char> >**

#### 16.1.64.1 Class data for basic\_ostringstream<char, char\_traits<char>, allocator<char> >

The virtual table for the std::basic\_ostringstream<char, std::char\_traits<char>, std::allocator<char> > class is described in the relevant architecture specific part of this specification.

The VTT for the std::basic\_ostringstream<char, std::char\_traits<char>, std::allocator<char> > class is described by [Table 16-169](#ID_VTT_45_24396)

**Table 16-169 VTT for basic\_ostringstream<char, char\_traits<char>, allocator<char> >**

|  |  |
| --- | --- |
| VTT Name | \_ZTTSt19basic\_ostringstreamIcSt11char\_traitsIcESaIcEE |
| Number of Entries | 4 |

#### 16.1.64.2 Interfaces for Class basic\_ostringstream<char, char\_traits<char>, allocator<char> >

An LSB conforming implementation shall provide the generic methods for Class std::basic\_ostringstream<char, std::char\_traits<char>, std::allocator<char> > specified in [Table 16-170](#ID_TBL_45_LIBSTDCXX_45_CLAUJ_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-170 libstdcxx - Class basic\_ostringstream<char, char\_traits<char>, allocator<char> > Function Interfaces**

|  |
| --- |
| basic\_ostringstream<char, char\_traits<char>, allocator<char> >::str() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostringstream<char, char\_traits<char>, allocator<char> >::rdbuf() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostringstream<char, char\_traits<char>, allocator<char> >::str(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostringstream<char, char\_traits<char>, allocator<char> >::basic\_ostringstream(basic\_string<char, char\_traits<char>, allocator<char> > const&, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostringstream<char, char\_traits<char>, allocator<char> >::basic\_ostringstream(\_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostringstream<char, char\_traits<char>, allocator<char> >::basic\_ostringstream(basic\_string<char, char\_traits<char>, allocator<char> > const&, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostringstream<char, char\_traits<char>, allocator<char> >::basic\_ostringstream(\_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostringstream<char, char\_traits<char>, allocator<char> >::~basic\_ostringstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostringstream<char, char\_traits<char>, allocator<char> >::~basic\_ostringstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostringstream<char, char\_traits<char>, allocator<char> >::~basic\_ostringstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::basic\_ostringstream<char, std::char\_traits<char>, std::allocator<char> > specified in [Table 16-171](#ID_TBL_45_LIBSTDCXX_45_CLAUJ_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-171 libstdcxx - Class basic\_ostringstream<char, char\_traits<char>, allocator<char> > Data Interfaces**

|  |
| --- |
| typeinfo for basic\_ostringstream<char, char\_traits<char>, allocator<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for basic\_ostringstream<char, char\_traits<char>, allocator<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| VTT for basic\_ostringstream<char, char\_traits<char>, allocator<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for basic\_ostringstream<char, char\_traits<char>, allocator<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.65 Class basic\_ostringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >**

#### 16.1.65.1 Class data for basic\_ostringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >

The virtual table for the std::basic\_ostringstream<wchar\_t, std::char\_traits<wchar\_t>, std::allocator<wchar\_t> > class is described in the relevant architecture specific part of this specification.

The VTT for the std::basic\_ostringstream<wchar\_t, std::char\_traits<wchar\_t>, std::allocator<wchar\_t> > class is described by [Table 16-172](#ID_VTT_45_24397)

**Table 16-172 VTT for basic\_ostringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >**

|  |  |
| --- | --- |
| VTT Name | \_ZTTSt19basic\_ostringstreamIwSt11char\_traitsIwESaIwEE |
| Number of Entries | 4 |

#### 16.1.65.2 Interfaces for Class basic\_ostringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >

An LSB conforming implementation shall provide the generic methods for Class std::basic\_ostringstream<wchar\_t, std::char\_traits<wchar\_t>, std::allocator<wchar\_t> > specified in [Table 16-173](#ID_TBL_45_LIBSTDCXX_45_CLAUK_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-173 libstdcxx - Class basic\_ostringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > Function Interfaces**

|  |
| --- |
| basic\_ostringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::str() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::rdbuf() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::str(basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_ostringstream(basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_ostringstream(\_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_ostringstream(basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_ostringstream(\_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::~basic\_ostringstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::~basic\_ostringstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::~basic\_ostringstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::basic\_ostringstream<wchar\_t, std::char\_traits<wchar\_t>, std::allocator<wchar\_t> > specified in [Table 16-174](#ID_TBL_45_LIBSTDCXX_45_CLAUK_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-174 libstdcxx - Class basic\_ostringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > Data Interfaces**

|  |
| --- |
| typeinfo for basic\_ostringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for basic\_ostringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| VTT for basic\_ostringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for basic\_ostringstream<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.66 Class basic\_stringbuf<char, char\_traits<char>, allocator<char> >**

#### 16.1.66.1 Class data for basic\_stringbuf<char, char\_traits<char>, allocator<char> >

The virtual table for the std::basic\_stringbuf<char, std::char\_traits<char>, std::allocator<char> > class is described by [Table 16-175](#ID_CLS_45_24311_45_0)

**Table 16-175 Primary vtable for basic\_stringbuf<char, char\_traits<char>, allocator<char> >**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for basic\_stringbuf<char, char\_traits<char>, allocator<char> > |
| vfunc[0]: | basic\_stringbuf<char, char\_traits<char>, allocator<char> >::~basic\_stringbuf() |
| vfunc[1]: | basic\_stringbuf<char, char\_traits<char>, allocator<char> >::~basic\_stringbuf() |
| vfunc[2]: | basic\_streambuf<char, char\_traits<char> >::imbue(locale const&) |
| vfunc[3]: | See architecture specific part. |
| vfunc[4]: | See architecture specific part. |
| vfunc[5]: | basic\_stringbuf<char, char\_traits<char>, allocator<char> >::seekpos(fpos<\_\_mbstate\_t>, \_Ios\_Openmode) |
| vfunc[6]: | basic\_streambuf<char, char\_traits<char> >::sync() |
| vfunc[7]: | basic\_streambuf<char, char\_traits<char> >::showmanyc() |
| vfunc[8]: | See architecture specific part. |
| vfunc[9]: | basic\_stringbuf<char, char\_traits<char>, allocator<char> >::underflow() |
| vfunc[10]: | basic\_streambuf<char, char\_traits<char> >::uflow() |
| vfunc[11]: | basic\_stringbuf<char, char\_traits<char>, allocator<char> >::pbackfail(int) |
| vfunc[12]: | See architecture specific part. |
| vfunc[13]: | basic\_stringbuf<char, char\_traits<char>, allocator<char> >::overflow(int) |

The Run Time Type Information for the std::basic\_stringbuf<char, std::char\_traits<char>, std::allocator<char> > class is described by [Table 16-176](#ID_RTTI_45_24089)

**Table 16-176 typeinfo for basic\_stringbuf<char, char\_traits<char>, allocator<char> >**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for basic\_stringbuf<char, char\_traits<char>, allocator<char> > |

#### 16.1.66.2 Interfaces for Class basic\_stringbuf<char, char\_traits<char>, allocator<char> >

An LSB conforming implementation shall provide the generic methods for Class std::basic\_stringbuf<char, std::char\_traits<char>, std::allocator<char> > specified in [Table 16-177](#ID_TBL_45_LIBSTDCXX_45_CLAUL_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-177 libstdcxx - Class basic\_stringbuf<char, char\_traits<char>, allocator<char> > Function Interfaces**

|  |
| --- |
| basic\_stringbuf<char, char\_traits<char>, allocator<char> >::str() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringbuf<char, char\_traits<char>, allocator<char> >::\_M\_update\_egptr()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringbuf<char, char\_traits<char>, allocator<char> >::\_M\_stringbuf\_init(\_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringbuf<char, char\_traits<char>, allocator<char> >::str(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringbuf<char, char\_traits<char>, allocator<char> >::seekpos(fpos<\_\_mbstate\_t>, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringbuf<char, char\_traits<char>, allocator<char> >::overflow(int)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringbuf<char, char\_traits<char>, allocator<char> >::pbackfail(int)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringbuf<char, char\_traits<char>, allocator<char> >::showmanyc()(GLIBCXX\_3.4.6) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringbuf<char, char\_traits<char>, allocator<char> >::underflow()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringbuf<char, char\_traits<char>, allocator<char> >::basic\_stringbuf(basic\_string<char, char\_traits<char>, allocator<char> > const&, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringbuf<char, char\_traits<char>, allocator<char> >::basic\_stringbuf(\_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringbuf<char, char\_traits<char>, allocator<char> >::basic\_stringbuf(basic\_string<char, char\_traits<char>, allocator<char> > const&, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringbuf<char, char\_traits<char>, allocator<char> >::basic\_stringbuf(\_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringbuf<char, char\_traits<char>, allocator<char> >::~basic\_stringbuf()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringbuf<char, char\_traits<char>, allocator<char> >::~basic\_stringbuf()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::basic\_stringbuf<char, std::char\_traits<char>, std::allocator<char> > specified in [Table 16-178](#ID_TBL_45_LIBSTDCXX_45_CLAUL_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-178 libstdcxx - Class basic\_stringbuf<char, char\_traits<char>, allocator<char> > Data Interfaces**

|  |
| --- |
| typeinfo for basic\_stringbuf<char, char\_traits<char>, allocator<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for basic\_stringbuf<char, char\_traits<char>, allocator<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for basic\_stringbuf<char, char\_traits<char>, allocator<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.67 Class basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >**

#### 16.1.67.1 Class data for basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >

The virtual table for the std::basic\_stringbuf<wchar\_t, std::char\_traits<wchar\_t>, std::allocator<wchar\_t> > class is described by [Table 16-179](#ID_CLS_45_24307_45_0)

**Table 16-179 Primary vtable for basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > |
| vfunc[0]: | basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::~basic\_stringbuf() |
| vfunc[1]: | basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::~basic\_stringbuf() |
| vfunc[2]: | basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::imbue(locale const&) |
| vfunc[3]: | See architecture specific part. |
| vfunc[4]: | See architecture specific part. |
| vfunc[5]: | basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::seekpos(fpos<\_\_mbstate\_t>, \_Ios\_Openmode) |
| vfunc[6]: | basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::sync() |
| vfunc[7]: | basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::showmanyc() |
| vfunc[8]: | See architecture specific part. |
| vfunc[9]: | basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::underflow() |
| vfunc[10]: | basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::uflow() |
| vfunc[11]: | basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::pbackfail(unsigned int) |
| vfunc[12]: | See architecture specific part. |
| vfunc[13]: | basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::overflow(unsigned int) |

The Run Time Type Information for the std::basic\_stringbuf<wchar\_t, std::char\_traits<wchar\_t>, std::allocator<wchar\_t> > class is described by [Table 16-180](#ID_RTTI_45_24088)

**Table 16-180 typeinfo for basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > |

#### 16.1.67.2 Interfaces for Class basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >

An LSB conforming implementation shall provide the generic methods for Class std::basic\_stringbuf<wchar\_t, std::char\_traits<wchar\_t>, std::allocator<wchar\_t> > specified in [Table 16-181](#ID_TBL_45_LIBSTDCXX_45_CLAUM_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-181 libstdcxx - Class basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > Function Interfaces**

|  |
| --- |
| basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::str() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_M\_update\_egptr()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::\_M\_stringbuf\_init(\_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::str(basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::seekpos(fpos<\_\_mbstate\_t>, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::overflow(unsigned int)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::pbackfail(unsigned int)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::showmanyc()(GLIBCXX\_3.4.6) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::underflow()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_stringbuf(basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_stringbuf(\_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_stringbuf(basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::basic\_stringbuf(\_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::~basic\_stringbuf()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >::~basic\_stringbuf()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::basic\_stringbuf<wchar\_t, std::char\_traits<wchar\_t>, std::allocator<wchar\_t> > specified in [Table 16-182](#ID_TBL_45_LIBSTDCXX_45_CLAUM_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-182 libstdcxx - Class basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > Data Interfaces**

|  |
| --- |
| typeinfo for basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for basic\_stringbuf<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.68 Class basic\_iostream<char, char\_traits<char> >**

#### 16.1.68.1 Class data for basic\_iostream<char, char\_traits<char> >

The virtual table for the std::basic\_iostream<char, std::char\_traits<char> > class is described in the relevant architecture specific part of this specification.

The VTT for the std::basic\_iostream<char, std::char\_traits<char> > class is described by [Table 16-183](#ID_VTT_45_24660)

**Table 16-183 VTT for basic\_iostream<char, char\_traits<char> >**

|  |  |
| --- | --- |
| VTT Name | \_ZTTSd |
| Number of Entries | 7 |

#### 16.1.68.2 Interfaces for Class basic\_iostream<char, char\_traits<char> >

An LSB conforming implementation shall provide the generic methods for Class std::basic\_iostream<char, std::char\_traits<char> > specified in [Table 16-184](#ID_TBL_45_LIBSTDCXX_45_CLAUN_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-184 libstdcxx - Class basic\_iostream<char, char\_traits<char> > Function Interfaces**

|  |
| --- |
| basic\_iostream<char, char\_traits<char> >::basic\_iostream(basic\_streambuf<char, char\_traits<char> >\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_iostream<char, char\_traits<char> >::basic\_iostream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_iostream<char, char\_traits<char> >::basic\_iostream(basic\_streambuf<char, char\_traits<char> >\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_iostream<char, char\_traits<char> >::basic\_iostream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_iostream<char, char\_traits<char> >::~basic\_iostream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_iostream<char, char\_traits<char> >::~basic\_iostream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_iostream<char, char\_traits<char> >::~basic\_iostream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >& operator>><char\_traits<char> >(basic\_istream<char, char\_traits<char> >&, signed char\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::basic\_iostream<char, std::char\_traits<char> > specified in [Table 16-185](#ID_TBL_45_LIBSTDCXX_45_CLAUN_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-185 libstdcxx - Class basic\_iostream<char, char\_traits<char> > Data Interfaces**

|  |
| --- |
| typeinfo for basic\_iostream<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for basic\_iostream<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| VTT for basic\_iostream<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for basic\_iostream<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.69 Class basic\_iostream<wchar\_t, char\_traits<wchar\_t> >**

#### 16.1.69.1 Class data for basic\_iostream<wchar\_t, char\_traits<wchar\_t> >

The virtual table for the std::basic\_iostream<wchar\_t, std::char\_traits<wchar\_t> > class is described in the relevant architecture specific part of this specification.

The VTT for the std::basic\_iostream<wchar\_t, std::char\_traits<wchar\_t> > class is described by [Table 16-186](#ID_VTT_45_24661)

**Table 16-186 VTT for basic\_iostream<wchar\_t, char\_traits<wchar\_t> >**

|  |  |
| --- | --- |
| VTT Name | \_ZTTSt14basic\_iostreamIwSt11char\_traitsIwEE |
| Number of Entries | 7 |

#### 16.1.69.2 Interfaces for Class basic\_iostream<wchar\_t, char\_traits<wchar\_t> >

An LSB conforming implementation shall provide the generic methods for Class std::basic\_iostream<wchar\_t, std::char\_traits<wchar\_t> > specified in [Table 16-187](#ID_TBL_45_LIBSTDCXX_45_CLAUO_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-187 libstdcxx - Class basic\_iostream<wchar\_t, char\_traits<wchar\_t> > Function Interfaces**

|  |
| --- |
| basic\_iostream<wchar\_t, char\_traits<wchar\_t> >::basic\_iostream(basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_iostream<wchar\_t, char\_traits<wchar\_t> >::basic\_iostream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_iostream<wchar\_t, char\_traits<wchar\_t> >::basic\_iostream(basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_iostream<wchar\_t, char\_traits<wchar\_t> >::basic\_iostream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_iostream<wchar\_t, char\_traits<wchar\_t> >::~basic\_iostream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_iostream<wchar\_t, char\_traits<wchar\_t> >::~basic\_iostream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_iostream<wchar\_t, char\_traits<wchar\_t> >::~basic\_iostream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::basic\_iostream<wchar\_t, std::char\_traits<wchar\_t> > specified in [Table 16-188](#ID_TBL_45_LIBSTDCXX_45_CLAUO_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-188 libstdcxx - Class basic\_iostream<wchar\_t, char\_traits<wchar\_t> > Data Interfaces**

|  |
| --- |
| typeinfo for basic\_iostream<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for basic\_iostream<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| VTT for basic\_iostream<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for basic\_iostream<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.70 Class basic\_istream<char, char\_traits<char> >**

#### 16.1.70.1 Class data for basic\_istream<char, char\_traits<char> >

The virtual table for the std::basic\_istream<char, std::char\_traits<char> > class is described in the relevant architecture specific part of this specification.

The VTT for the std::basic\_istream<char, std::char\_traits<char> > class is described by [Table 16-189](#ID_VTT_45_24999)

**Table 16-189 VTT for basic\_istream<char, char\_traits<char> >**

|  |  |
| --- | --- |
| VTT Name | \_ZTTSi |
| Number of Entries | 2 |

#### 16.1.70.2 Interfaces for Class basic\_istream<char, char\_traits<char> >

An LSB conforming implementation shall provide the generic methods for Class std::basic\_istream<char, std::char\_traits<char> > specified in [Table 16-190](#ID_TBL_45_LIBSTDCXX_45_CLAUP_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-190 libstdcxx - Class basic\_istream<char, char\_traits<char> > Function Interfaces**

|  |
| --- |
| basic\_istream<char, char\_traits<char> >::gcount() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::sentry::operator bool() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::get(basic\_streambuf<char, char\_traits<char> >&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::get(basic\_streambuf<char, char\_traits<char> >&, char)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::get(char&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::get()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::peek()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::sync()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::seekg(fpos<\_\_mbstate\_t>)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::tellg()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::unget()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::ignore()(GLIBCXX\_3.4.5) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::sentry::sentry(basic\_istream<char, char\_traits<char> >&, bool)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::sentry::sentry(basic\_istream<char, char\_traits<char> >&, bool)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::putback(char)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::basic\_istream(basic\_streambuf<char, char\_traits<char> >\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::basic\_istream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::basic\_istream(basic\_streambuf<char, char\_traits<char> >\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::basic\_istream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::~basic\_istream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::~basic\_istream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::~basic\_istream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::operator>>(basic\_istream<char, char\_traits<char> >& (\*)(basic\_istream<char, char\_traits<char> >&))(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::operator>>(ios\_base& (\*)(ios\_base&))(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::operator>>(basic\_ios<char, char\_traits<char> >& (\*)(basic\_ios<char, char\_traits<char> >&))(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::operator>>(basic\_streambuf<char, char\_traits<char> >\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::operator>>(void\*&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::operator>>(bool&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::operator>>(double&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::operator>>(long double&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::operator>>(float&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::operator>>(int&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::operator>>(unsigned int&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::operator>>(long&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::operator>>(unsigned long&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::operator>>(short&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::operator>>(unsigned short&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::operator>>(long long&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >::operator>>(unsigned long long&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >& ws<char, char\_traits<char> >(basic\_istream<char, char\_traits<char> >&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >& getline<char, char\_traits<char>, allocator<char> >(basic\_istream<char, char\_traits<char> >&, basic\_string<char, char\_traits<char>, allocator<char> >&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >& getline<char, char\_traits<char>, allocator<char> >(basic\_istream<char, char\_traits<char> >&, basic\_string<char, char\_traits<char>, allocator<char> >&, char)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >& operator>><char\_traits<char> >(basic\_istream<char, char\_traits<char> >&, unsigned char\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >& operator>><char\_traits<char> >(basic\_istream<char, char\_traits<char> >&, signed char&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >& operator>><char\_traits<char> >(basic\_istream<char, char\_traits<char> >&, unsigned char&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >& operator>><char, char\_traits<char> >(basic\_istream<char, char\_traits<char> >&, char\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >& operator>><char, char\_traits<char> >(basic\_istream<char, char\_traits<char> >&, char&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >& operator>><char, char\_traits<char> >(basic\_istream<char, char\_traits<char> >&, \_Setiosflags)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >& operator>><char, char\_traits<char> >(basic\_istream<char, char\_traits<char> >&, \_Setprecision)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >& operator>><char, char\_traits<char> >(basic\_istream<char, char\_traits<char> >&, \_Resetiosflags)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >& operator>><char, char\_traits<char> >(basic\_istream<char, char\_traits<char> >&, \_Setw)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >& operator>><char, char\_traits<char> >(basic\_istream<char, char\_traits<char> >&, \_Setbase)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >& operator>><char, char\_traits<char> >(basic\_istream<char, char\_traits<char> >&, \_Setfill<char>)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >& operator>><char, char\_traits<char>, allocator<char> >(basic\_istream<char, char\_traits<char> >&, basic\_string<char, char\_traits<char>, allocator<char> >&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >& operator>><double, char, char\_traits<char> >(basic\_istream<char, char\_traits<char> >&, complex<double>&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >& operator>><long double, char, char\_traits<char> >(basic\_istream<char, char\_traits<char> >&, complex<long double>&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<char, char\_traits<char> >& operator>><float, char, char\_traits<char> >(basic\_istream<char, char\_traits<char> >&, complex<float>&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::basic\_istream<char, std::char\_traits<char> > specified in [Table 16-191](#ID_TBL_45_LIBSTDCXX_45_CLAUP_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-191 libstdcxx - Class basic\_istream<char, char\_traits<char> > Data Interfaces**

|  |
| --- |
| typeinfo for basic\_istream<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for basic\_istream<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| VTT for basic\_istream<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for basic\_istream<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.71 Class basic\_istream<wchar\_t, char\_traits<wchar\_t> >**

#### 16.1.71.1 Class data for basic\_istream<wchar\_t, char\_traits<wchar\_t> >

The virtual table for the std::basic\_istream<wchar\_t, std::char\_traits<wchar\_t> > class is described in the relevant architecture specific part of this specification.

The VTT for the std::basic\_istream<wchar\_t, std::char\_traits<wchar\_t> > class is described by [Table 16-192](#ID_VTT_45_25001)

**Table 16-192 VTT for basic\_istream<wchar\_t, char\_traits<wchar\_t> >**

|  |  |
| --- | --- |
| VTT Name | \_ZTTSt13basic\_istreamIwSt11char\_traitsIwEE |
| Number of Entries | 2 |

#### 16.1.71.2 Interfaces for Class basic\_istream<wchar\_t, char\_traits<wchar\_t> >

An LSB conforming implementation shall provide the generic methods for Class std::basic\_istream<wchar\_t, std::char\_traits<wchar\_t> > specified in [Table 16-193](#ID_TBL_45_LIBSTDCXX_45_CLAUQ_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-193 libstdcxx - Class basic\_istream<wchar\_t, char\_traits<wchar\_t> > Function Interfaces**

|  |
| --- |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::gcount() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::sentry::operator bool() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::get(basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::get(basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >&, wchar\_t)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::get(wchar\_t&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::get()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::peek()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::sync()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::seekg(fpos<\_\_mbstate\_t>)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::tellg()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::unget()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::ignore()(GLIBCXX\_3.4.5) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::sentry::sentry(basic\_istream<wchar\_t, char\_traits<wchar\_t> >&, bool)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::sentry::sentry(basic\_istream<wchar\_t, char\_traits<wchar\_t> >&, bool)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::putback(wchar\_t)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::basic\_istream(basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::basic\_istream(basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::~basic\_istream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::~basic\_istream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::~basic\_istream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::operator>>(basic\_istream<wchar\_t, char\_traits<wchar\_t> >& (\*)(basic\_istream<wchar\_t, char\_traits<wchar\_t> >&))(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::operator>>(ios\_base& (\*)(ios\_base&))(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::operator>>(basic\_ios<wchar\_t, char\_traits<wchar\_t> >& (\*)(basic\_ios<wchar\_t, char\_traits<wchar\_t> >&))(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::operator>>(basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::operator>>(void\*&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::operator>>(bool&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::operator>>(double&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::operator>>(long double&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::operator>>(float&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::operator>>(int&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::operator>>(unsigned int&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::operator>>(long&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::operator>>(unsigned long&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::operator>>(short&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::operator>>(unsigned short&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::operator>>(long long&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::operator>>(unsigned long long&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >& ws<wchar\_t, char\_traits<wchar\_t> >(basic\_istream<wchar\_t, char\_traits<wchar\_t> >&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >& getline<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >(basic\_istream<wchar\_t, char\_traits<wchar\_t> >&, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >& getline<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >(basic\_istream<wchar\_t, char\_traits<wchar\_t> >&, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >&, wchar\_t)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >& operator>><double, wchar\_t, char\_traits<wchar\_t> >(basic\_istream<wchar\_t, char\_traits<wchar\_t> >&, complex<double>&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >& operator>><long double, wchar\_t, char\_traits<wchar\_t> >(basic\_istream<wchar\_t, char\_traits<wchar\_t> >&, complex<long double>&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >& operator>><float, wchar\_t, char\_traits<wchar\_t> >(basic\_istream<wchar\_t, char\_traits<wchar\_t> >&, complex<float>&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >& operator>><wchar\_t, char\_traits<wchar\_t> >(basic\_istream<wchar\_t, char\_traits<wchar\_t> >&, wchar\_t\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >& operator>><wchar\_t, char\_traits<wchar\_t> >(basic\_istream<wchar\_t, char\_traits<wchar\_t> >&, wchar\_t&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >& operator>><wchar\_t, char\_traits<wchar\_t> >(basic\_istream<wchar\_t, char\_traits<wchar\_t> >&, \_Setiosflags)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >& operator>><wchar\_t, char\_traits<wchar\_t> >(basic\_istream<wchar\_t, char\_traits<wchar\_t> >&, \_Setprecision)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >& operator>><wchar\_t, char\_traits<wchar\_t> >(basic\_istream<wchar\_t, char\_traits<wchar\_t> >&, \_Resetiosflags)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >& operator>><wchar\_t, char\_traits<wchar\_t> >(basic\_istream<wchar\_t, char\_traits<wchar\_t> >&, \_Setw)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >& operator>><wchar\_t, char\_traits<wchar\_t> >(basic\_istream<wchar\_t, char\_traits<wchar\_t> >&, \_Setbase)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >& operator>><wchar\_t, char\_traits<wchar\_t> >(basic\_istream<wchar\_t, char\_traits<wchar\_t> >&, \_Setfill<wchar\_t>)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >& operator>><wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >(basic\_istream<wchar\_t, char\_traits<wchar\_t> >&, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::basic\_istream<wchar\_t, std::char\_traits<wchar\_t> > specified in [Table 16-194](#ID_TBL_45_LIBSTDCXX_45_CLAUQ_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-194 libstdcxx - Class basic\_istream<wchar\_t, char\_traits<wchar\_t> > Data Interfaces**

|  |
| --- |
| typeinfo for basic\_istream<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for basic\_istream<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| VTT for basic\_istream<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for basic\_istream<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.72 Class istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >**

#### 16.1.72.1 Interfaces for Class istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >

An LSB conforming implementation shall provide the generic methods for Class std::istreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > specified in [Table 16-195](#ID_TBL_45_LIBSTDCXX_45_CLAUR_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-195 libstdcxx - Class istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > Function Interfaces**

|  |
| --- |
| istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >::operator++()(GLIBCXX\_3.4.5) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.73 Class istreambuf\_iterator<char, char\_traits<char> >**

#### 16.1.73.1 Interfaces for Class istreambuf\_iterator<char, char\_traits<char> >

An LSB conforming implementation shall provide the generic methods for Class std::istreambuf\_iterator<char, std::char\_traits<char> > specified in [Table 16-196](#ID_TBL_45_LIBSTDCXX_45_CLAUS_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-196 libstdcxx - Class istreambuf\_iterator<char, char\_traits<char> > Function Interfaces**

|  |
| --- |
| istreambuf\_iterator<char, char\_traits<char> >::operator++()(GLIBCXX\_3.4.5) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.74 Class basic\_ostream<char, char\_traits<char> >**

#### 16.1.74.1 Class data for basic\_ostream<char, char\_traits<char> >

The virtual table for the std::basic\_ostream<char, std::char\_traits<char> > class is described in the relevant architecture specific part of this specification.

The VTT for the std::basic\_ostream<char, std::char\_traits<char> > class is described by [Table 16-197](#ID_VTT_45_25000)

**Table 16-197 VTT for basic\_ostream<char, char\_traits<char> >**

|  |  |
| --- | --- |
| VTT Name | \_ZTTSo |
| Number of Entries | 2 |

#### 16.1.74.2 Interfaces for Class basic\_ostream<char, char\_traits<char> >

An LSB conforming implementation shall provide the generic methods for Class std::basic\_ostream<char, std::char\_traits<char> > specified in [Table 16-198](#ID_TBL_45_LIBSTDCXX_45_CLAUT_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-198 libstdcxx - Class basic\_ostream<char, char\_traits<char> > Function Interfaces**

|  |
| --- |
| basic\_ostream<char, char\_traits<char> >::sentry::operator bool() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::put(char)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::flush()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::seekp(fpos<\_\_mbstate\_t>)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::tellp()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::sentry::sentry(basic\_ostream<char, char\_traits<char> >&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::sentry::sentry(basic\_ostream<char, char\_traits<char> >&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::sentry::~sentry()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::sentry::~sentry()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::basic\_ostream(basic\_streambuf<char, char\_traits<char> >\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::basic\_ostream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::basic\_ostream(basic\_streambuf<char, char\_traits<char> >\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::basic\_ostream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::~basic\_ostream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::~basic\_ostream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::~basic\_ostream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::operator<<(basic\_ostream<char, char\_traits<char> >& (\*)(basic\_ostream<char, char\_traits<char> >&))(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::operator<<(ios\_base& (\*)(ios\_base&))(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::operator<<(basic\_ios<char, char\_traits<char> >& (\*)(basic\_ios<char, char\_traits<char> >&))(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::operator<<(void const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::operator<<(basic\_streambuf<char, char\_traits<char> >\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::operator<<(bool)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::operator<<(double)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::operator<<(long double)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::operator<<(float)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::operator<<(int)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::operator<<(unsigned int)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::operator<<(long)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::operator<<(unsigned long)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::operator<<(short)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::operator<<(unsigned short)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::operator<<(long long)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >::operator<<(unsigned long long)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >& endl<char, char\_traits<char> >(basic\_ostream<char, char\_traits<char> >&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >& ends<char, char\_traits<char> >(basic\_ostream<char, char\_traits<char> >&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >& flush<char, char\_traits<char> >(basic\_ostream<char, char\_traits<char> >&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >& operator<< <char\_traits<char> >(basic\_ostream<char, char\_traits<char> >&, signed char const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >& operator<< <char\_traits<char> >(basic\_ostream<char, char\_traits<char> >&, char const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >& operator<< <char\_traits<char> >(basic\_ostream<char, char\_traits<char> >&, unsigned char const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >& operator<< <char\_traits<char> >(basic\_ostream<char, char\_traits<char> >&, signed char)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >& operator<< <char\_traits<char> >(basic\_ostream<char, char\_traits<char> >&, char)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >& operator<< <char\_traits<char> >(basic\_ostream<char, char\_traits<char> >&, unsigned char)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >& operator<< <char, char\_traits<char> >(basic\_ostream<char, char\_traits<char> >&, \_Setiosflags)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >& operator<< <char, char\_traits<char> >(basic\_ostream<char, char\_traits<char> >&, \_Setprecision)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >& operator<< <char, char\_traits<char> >(basic\_ostream<char, char\_traits<char> >&, \_Resetiosflags)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >& operator<< <char, char\_traits<char> >(basic\_ostream<char, char\_traits<char> >&, \_Setw)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >& operator<< <char, char\_traits<char> >(basic\_ostream<char, char\_traits<char> >&, \_Setbase)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >& operator<< <char, char\_traits<char> >(basic\_ostream<char, char\_traits<char> >&, \_Setfill<char>)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >& operator<< <char, char\_traits<char>, allocator<char> >(basic\_ostream<char, char\_traits<char> >&, basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >& operator<< <double, char, char\_traits<char> >(basic\_ostream<char, char\_traits<char> >&, complex<double> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >& operator<< <long double, char, char\_traits<char> >(basic\_ostream<char, char\_traits<char> >&, complex<long double> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<char, char\_traits<char> >& operator<< <float, char, char\_traits<char> >(basic\_ostream<char, char\_traits<char> >&, complex<float> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::basic\_ostream<char, std::char\_traits<char> > specified in [Table 16-199](#ID_TBL_45_LIBSTDCXX_45_CLAUT_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-199 libstdcxx - Class basic\_ostream<char, char\_traits<char> > Data Interfaces**

|  |
| --- |
| typeinfo for basic\_ostream<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for basic\_ostream<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| VTT for basic\_ostream<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for basic\_ostream<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.75 Class basic\_ostream<wchar\_t, char\_traits<wchar\_t> >**

#### 16.1.75.1 Class data for basic\_ostream<wchar\_t, char\_traits<wchar\_t> >

The virtual table for the std::basic\_ostream<wchar\_t, std::char\_traits<wchar\_t> > class is described in the relevant architecture specific part of this specification.

The VTT for the std::basic\_ostream<wchar\_t, std::char\_traits<wchar\_t> > class is described by [Table 16-200](#ID_VTT_45_25002)

**Table 16-200 VTT for basic\_ostream<wchar\_t, char\_traits<wchar\_t> >**

|  |  |
| --- | --- |
| VTT Name | \_ZTTSt13basic\_ostreamIwSt11char\_traitsIwEE |
| Number of Entries | 2 |

#### 16.1.75.2 Interfaces for Class basic\_ostream<wchar\_t, char\_traits<wchar\_t> >

An LSB conforming implementation shall provide the generic methods for Class std::basic\_ostream<wchar\_t, std::char\_traits<wchar\_t> > specified in [Table 16-201](#ID_TBL_45_LIBSTDCXX_45_CLAUU_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-201 libstdcxx - Class basic\_ostream<wchar\_t, char\_traits<wchar\_t> > Function Interfaces**

|  |
| --- |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::sentry::operator bool() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::put(wchar\_t)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::flush()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::seekp(fpos<\_\_mbstate\_t>)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::tellp()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::sentry::sentry(basic\_ostream<wchar\_t, char\_traits<wchar\_t> >&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::sentry::sentry(basic\_ostream<wchar\_t, char\_traits<wchar\_t> >&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::sentry::~sentry()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::sentry::~sentry()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::basic\_ostream(basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::basic\_ostream(basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::~basic\_ostream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::~basic\_ostream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::~basic\_ostream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::operator<<(basic\_ostream<wchar\_t, char\_traits<wchar\_t> >& (\*)(basic\_ostream<wchar\_t, char\_traits<wchar\_t> >&))(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::operator<<(ios\_base& (\*)(ios\_base&))(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::operator<<(basic\_ios<wchar\_t, char\_traits<wchar\_t> >& (\*)(basic\_ios<wchar\_t, char\_traits<wchar\_t> >&))(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::operator<<(void const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::operator<<(basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::operator<<(bool)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::operator<<(double)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::operator<<(long double)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::operator<<(float)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::operator<<(int)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::operator<<(unsigned int)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::operator<<(long)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::operator<<(unsigned long)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::operator<<(short)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::operator<<(unsigned short)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::operator<<(long long)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::operator<<(unsigned long long)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >& endl<wchar\_t, char\_traits<wchar\_t> >(basic\_ostream<wchar\_t, char\_traits<wchar\_t> >&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >& ends<wchar\_t, char\_traits<wchar\_t> >(basic\_ostream<wchar\_t, char\_traits<wchar\_t> >&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >& flush<wchar\_t, char\_traits<wchar\_t> >(basic\_ostream<wchar\_t, char\_traits<wchar\_t> >&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >& operator<< <double, wchar\_t, char\_traits<wchar\_t> >(basic\_ostream<wchar\_t, char\_traits<wchar\_t> >&, complex<double> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >& operator<< <long double, wchar\_t, char\_traits<wchar\_t> >(basic\_ostream<wchar\_t, char\_traits<wchar\_t> >&, complex<long double> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >& operator<< <float, wchar\_t, char\_traits<wchar\_t> >(basic\_ostream<wchar\_t, char\_traits<wchar\_t> >&, complex<float> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >& operator<< <wchar\_t, char\_traits<wchar\_t> >(basic\_ostream<wchar\_t, char\_traits<wchar\_t> >&, wchar\_t const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >& operator<< <wchar\_t, char\_traits<wchar\_t> >(basic\_ostream<wchar\_t, char\_traits<wchar\_t> >&, char const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >& operator<< <wchar\_t, char\_traits<wchar\_t> >(basic\_ostream<wchar\_t, char\_traits<wchar\_t> >&, wchar\_t)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >& operator<< <wchar\_t, char\_traits<wchar\_t> >(basic\_ostream<wchar\_t, char\_traits<wchar\_t> >&, \_Setiosflags)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >& operator<< <wchar\_t, char\_traits<wchar\_t> >(basic\_ostream<wchar\_t, char\_traits<wchar\_t> >&, \_Setprecision)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >& operator<< <wchar\_t, char\_traits<wchar\_t> >(basic\_ostream<wchar\_t, char\_traits<wchar\_t> >&, \_Resetiosflags)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >& operator<< <wchar\_t, char\_traits<wchar\_t> >(basic\_ostream<wchar\_t, char\_traits<wchar\_t> >&, \_Setw)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >& operator<< <wchar\_t, char\_traits<wchar\_t> >(basic\_ostream<wchar\_t, char\_traits<wchar\_t> >&, \_Setbase)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >& operator<< <wchar\_t, char\_traits<wchar\_t> >(basic\_ostream<wchar\_t, char\_traits<wchar\_t> >&, \_Setfill<wchar\_t>)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >& operator<< <wchar\_t, char\_traits<wchar\_t> >(basic\_ostream<wchar\_t, char\_traits<wchar\_t> >&, char)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >& operator<< <wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >(basic\_ostream<wchar\_t, char\_traits<wchar\_t> >&, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::basic\_ostream<wchar\_t, std::char\_traits<wchar\_t> > specified in [Table 16-202](#ID_TBL_45_LIBSTDCXX_45_CLAUU_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-202 libstdcxx - Class basic\_ostream<wchar\_t, char\_traits<wchar\_t> > Data Interfaces**

|  |
| --- |
| typeinfo for basic\_ostream<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for basic\_ostream<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| VTT for basic\_ostream<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for basic\_ostream<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.76 Class basic\_fstream<char, char\_traits<char> >**

#### 16.1.76.1 Class data for basic\_fstream<char, char\_traits<char> >

The virtual table for the std::basic\_fstream<char, std::char\_traits<char> > class is described in the relevant architecture specific part of this specification.

The VTT for the std::basic\_fstream<char, std::char\_traits<char> > class is described by [Table 16-203](#ID_VTT_45_24740)

**Table 16-203 VTT for basic\_fstream<char, char\_traits<char> >**

|  |  |
| --- | --- |
| VTT Name | \_ZTTSt13basic\_fstreamIcSt11char\_traitsIcEE |
| Number of Entries | 10 |

#### 16.1.76.2 Interfaces for Class basic\_fstream<char, char\_traits<char> >

An LSB conforming implementation shall provide the generic methods for Class std::basic\_fstream<char, std::char\_traits<char> > specified in [Table 16-204](#ID_TBL_45_LIBSTDCXX_45_CLAUV_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-204 libstdcxx - Class basic\_fstream<char, char\_traits<char> > Function Interfaces**

|  |
| --- |
| basic\_fstream<char, char\_traits<char> >::rdbuf() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_fstream<char, char\_traits<char> >::is\_open() const(GLIBCXX\_3.4.5) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_fstream<char, char\_traits<char> >::open(char const\*, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_fstream<char, char\_traits<char> >::close()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_fstream<char, char\_traits<char> >::is\_open()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_fstream<char, char\_traits<char> >::basic\_fstream(char const\*, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_fstream<char, char\_traits<char> >::basic\_fstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_fstream<char, char\_traits<char> >::basic\_fstream(char const\*, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_fstream<char, char\_traits<char> >::basic\_fstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_fstream<char, char\_traits<char> >::~basic\_fstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_fstream<char, char\_traits<char> >::~basic\_fstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_fstream<char, char\_traits<char> >::~basic\_fstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::basic\_fstream<char, std::char\_traits<char> > specified in [Table 16-205](#ID_TBL_45_LIBSTDCXX_45_CLAUV_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-205 libstdcxx - Class basic\_fstream<char, char\_traits<char> > Data Interfaces**

|  |
| --- |
| typeinfo for basic\_fstream<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for basic\_fstream<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| VTT for basic\_fstream<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for basic\_fstream<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.77 Class basic\_fstream<wchar\_t, char\_traits<wchar\_t> >**

#### 16.1.77.1 Class data for basic\_fstream<wchar\_t, char\_traits<wchar\_t> >

The virtual table for the std::basic\_fstream<wchar\_t, std::char\_traits<wchar\_t> > class is described in the relevant architecture specific part of this specification.

The VTT for the std::basic\_fstream<wchar\_t, std::char\_traits<wchar\_t> > class is described by [Table 16-206](#ID_VTT_45_24741)

**Table 16-206 VTT for basic\_fstream<wchar\_t, char\_traits<wchar\_t> >**

|  |  |
| --- | --- |
| VTT Name | \_ZTTSt13basic\_fstreamIwSt11char\_traitsIwEE |
| Number of Entries | 10 |

#### 16.1.77.2 Interfaces for Class basic\_fstream<wchar\_t, char\_traits<wchar\_t> >

An LSB conforming implementation shall provide the generic methods for Class std::basic\_fstream<wchar\_t, std::char\_traits<wchar\_t> > specified in [Table 16-207](#ID_TBL_45_LIBSTDCXX_45_CLAUW_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-207 libstdcxx - Class basic\_fstream<wchar\_t, char\_traits<wchar\_t> > Function Interfaces**

|  |
| --- |
| basic\_fstream<wchar\_t, char\_traits<wchar\_t> >::rdbuf() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_fstream<wchar\_t, char\_traits<wchar\_t> >::is\_open() const(GLIBCXX\_3.4.5) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_fstream<wchar\_t, char\_traits<wchar\_t> >::open(char const\*, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_fstream<wchar\_t, char\_traits<wchar\_t> >::close()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_fstream<wchar\_t, char\_traits<wchar\_t> >::is\_open()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_fstream<wchar\_t, char\_traits<wchar\_t> >::basic\_fstream(char const\*, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_fstream<wchar\_t, char\_traits<wchar\_t> >::basic\_fstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_fstream<wchar\_t, char\_traits<wchar\_t> >::basic\_fstream(char const\*, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_fstream<wchar\_t, char\_traits<wchar\_t> >::basic\_fstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_fstream<wchar\_t, char\_traits<wchar\_t> >::~basic\_fstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_fstream<wchar\_t, char\_traits<wchar\_t> >::~basic\_fstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_fstream<wchar\_t, char\_traits<wchar\_t> >::~basic\_fstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::basic\_fstream<wchar\_t, std::char\_traits<wchar\_t> > specified in [Table 16-208](#ID_TBL_45_LIBSTDCXX_45_CLAUW_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-208 libstdcxx - Class basic\_fstream<wchar\_t, char\_traits<wchar\_t> > Data Interfaces**

|  |
| --- |
| typeinfo for basic\_fstream<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for basic\_fstream<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| VTT for basic\_fstream<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for basic\_fstream<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.78 Class basic\_ifstream<char, char\_traits<char> >**

#### 16.1.78.1 Class data for basic\_ifstream<char, char\_traits<char> >

The virtual table for the std::basic\_ifstream<char, std::char\_traits<char> > class is described in the relevant architecture specific part of this specification.

The VTT for the std::basic\_ifstream<char, std::char\_traits<char> > class is described by [Table 16-209](#ID_VTT_45_24390)

**Table 16-209 VTT for basic\_ifstream<char, char\_traits<char> >**

|  |  |
| --- | --- |
| VTT Name | \_ZTTSt14basic\_ifstreamIcSt11char\_traitsIcEE |
| Number of Entries | 4 |

#### 16.1.78.2 Interfaces for Class basic\_ifstream<char, char\_traits<char> >

An LSB conforming implementation shall provide the generic methods for Class std::basic\_ifstream<char, std::char\_traits<char> > specified in [Table 16-210](#ID_TBL_45_LIBSTDCXX_45_CLAUX_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-210 libstdcxx - Class basic\_ifstream<char, char\_traits<char> > Function Interfaces**

|  |
| --- |
| basic\_ifstream<char, char\_traits<char> >::rdbuf() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ifstream<char, char\_traits<char> >::is\_open() const(GLIBCXX\_3.4.5) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ifstream<char, char\_traits<char> >::open(char const\*, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ifstream<char, char\_traits<char> >::close()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ifstream<char, char\_traits<char> >::is\_open()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ifstream<char, char\_traits<char> >::basic\_ifstream(char const\*, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ifstream<char, char\_traits<char> >::basic\_ifstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ifstream<char, char\_traits<char> >::basic\_ifstream(char const\*, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ifstream<char, char\_traits<char> >::basic\_ifstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ifstream<char, char\_traits<char> >::~basic\_ifstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ifstream<char, char\_traits<char> >::~basic\_ifstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ifstream<char, char\_traits<char> >::~basic\_ifstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::basic\_ifstream<char, std::char\_traits<char> > specified in [Table 16-211](#ID_TBL_45_LIBSTDCXX_45_CLAUX_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-211 libstdcxx - Class basic\_ifstream<char, char\_traits<char> > Data Interfaces**

|  |
| --- |
| typeinfo for basic\_ifstream<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for basic\_ifstream<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| VTT for basic\_ifstream<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for basic\_ifstream<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.79 Class basic\_ifstream<wchar\_t, char\_traits<wchar\_t> >**

#### 16.1.79.1 Class data for basic\_ifstream<wchar\_t, char\_traits<wchar\_t> >

The virtual table for the std::basic\_ifstream<wchar\_t, std::char\_traits<wchar\_t> > class is described in the relevant architecture specific part of this specification.

The VTT for the std::basic\_ifstream<wchar\_t, std::char\_traits<wchar\_t> > class is described by [Table 16-212](#ID_VTT_45_24391)

**Table 16-212 VTT for basic\_ifstream<wchar\_t, char\_traits<wchar\_t> >**

|  |  |
| --- | --- |
| VTT Name | \_ZTTSt14basic\_ifstreamIwSt11char\_traitsIwEE |
| Number of Entries | 4 |

#### 16.1.79.2 Interfaces for Class basic\_ifstream<wchar\_t, char\_traits<wchar\_t> >

An LSB conforming implementation shall provide the generic methods for Class std::basic\_ifstream<wchar\_t, std::char\_traits<wchar\_t> > specified in [Table 16-213](#ID_TBL_45_LIBSTDCXX_45_CLAUY_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-213 libstdcxx - Class basic\_ifstream<wchar\_t, char\_traits<wchar\_t> > Function Interfaces**

|  |
| --- |
| basic\_ifstream<wchar\_t, char\_traits<wchar\_t> >::rdbuf() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ifstream<wchar\_t, char\_traits<wchar\_t> >::is\_open() const(GLIBCXX\_3.4.5) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ifstream<wchar\_t, char\_traits<wchar\_t> >::open(char const\*, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ifstream<wchar\_t, char\_traits<wchar\_t> >::close()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ifstream<wchar\_t, char\_traits<wchar\_t> >::is\_open()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ifstream<wchar\_t, char\_traits<wchar\_t> >::basic\_ifstream(char const\*, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ifstream<wchar\_t, char\_traits<wchar\_t> >::basic\_ifstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ifstream<wchar\_t, char\_traits<wchar\_t> >::basic\_ifstream(char const\*, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ifstream<wchar\_t, char\_traits<wchar\_t> >::basic\_ifstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ifstream<wchar\_t, char\_traits<wchar\_t> >::~basic\_ifstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ifstream<wchar\_t, char\_traits<wchar\_t> >::~basic\_ifstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ifstream<wchar\_t, char\_traits<wchar\_t> >::~basic\_ifstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::basic\_ifstream<wchar\_t, std::char\_traits<wchar\_t> > specified in [Table 16-214](#ID_TBL_45_LIBSTDCXX_45_CLAUY_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-214 libstdcxx - Class basic\_ifstream<wchar\_t, char\_traits<wchar\_t> > Data Interfaces**

|  |
| --- |
| typeinfo for basic\_ifstream<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo for basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for basic\_ifstream<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| VTT for basic\_ifstream<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for basic\_ifstream<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.80 Class basic\_ofstream<char, char\_traits<char> >**

#### 16.1.80.1 Class data for basic\_ofstream<char, char\_traits<char> >

The virtual table for the std::basic\_ofstream<char, std::char\_traits<char> > class is described in the relevant architecture specific part of this specification.

The VTT for the std::basic\_ofstream<char, std::char\_traits<char> > class is described by [Table 16-215](#ID_VTT_45_24392)

**Table 16-215 VTT for basic\_ofstream<char, char\_traits<char> >**

|  |  |
| --- | --- |
| VTT Name | \_ZTTSt14basic\_ofstreamIcSt11char\_traitsIcEE |
| Number of Entries | 4 |

#### 16.1.80.2 Interfaces for Class basic\_ofstream<char, char\_traits<char> >

An LSB conforming implementation shall provide the generic methods for Class std::basic\_ofstream<char, std::char\_traits<char> > specified in [Table 16-216](#ID_TBL_45_LIBSTDCXX_45_CLAUZ_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-216 libstdcxx - Class basic\_ofstream<char, char\_traits<char> > Function Interfaces**

|  |
| --- |
| basic\_ofstream<char, char\_traits<char> >::rdbuf() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ofstream<char, char\_traits<char> >::is\_open() const(GLIBCXX\_3.4.5) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ofstream<char, char\_traits<char> >::open(char const\*, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ofstream<char, char\_traits<char> >::close()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ofstream<char, char\_traits<char> >::is\_open()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ofstream<char, char\_traits<char> >::basic\_ofstream(char const\*, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ofstream<char, char\_traits<char> >::basic\_ofstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ofstream<char, char\_traits<char> >::basic\_ofstream(char const\*, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ofstream<char, char\_traits<char> >::basic\_ofstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ofstream<char, char\_traits<char> >::~basic\_ofstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ofstream<char, char\_traits<char> >::~basic\_ofstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ofstream<char, char\_traits<char> >::~basic\_ofstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::basic\_ofstream<char, std::char\_traits<char> > specified in [Table 16-217](#ID_TBL_45_LIBSTDCXX_45_CLAUZ_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-217 libstdcxx - Class basic\_ofstream<char, char\_traits<char> > Data Interfaces**

|  |
| --- |
| typeinfo for basic\_ofstream<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for basic\_ofstream<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| VTT for basic\_ofstream<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for basic\_ofstream<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.81 Class basic\_ofstream<wchar\_t, char\_traits<wchar\_t> >**

#### 16.1.81.1 Class data for basic\_ofstream<wchar\_t, char\_traits<wchar\_t> >

The virtual table for the std::basic\_ofstream<wchar\_t, std::char\_traits<wchar\_t> > class is described in the relevant architecture specific part of this specification.

The VTT for the std::basic\_ofstream<wchar\_t, std::char\_traits<wchar\_t> > class is described by [Table 16-218](#ID_VTT_45_24393)

**Table 16-218 VTT for basic\_ofstream<wchar\_t, char\_traits<wchar\_t> >**

|  |  |
| --- | --- |
| VTT Name | \_ZTTSt14basic\_ofstreamIwSt11char\_traitsIwEE |
| Number of Entries | 4 |

#### 16.1.81.2 Interfaces for Class basic\_ofstream<wchar\_t, char\_traits<wchar\_t> >

An LSB conforming implementation shall provide the generic methods for Class std::basic\_ofstream<wchar\_t, std::char\_traits<wchar\_t> > specified in [Table 16-219](#ID_TBL_45_LIBSTDCXX_45_CLAVA_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-219 libstdcxx - Class basic\_ofstream<wchar\_t, char\_traits<wchar\_t> > Function Interfaces**

|  |
| --- |
| basic\_ofstream<wchar\_t, char\_traits<wchar\_t> >::rdbuf() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ofstream<wchar\_t, char\_traits<wchar\_t> >::is\_open() const(GLIBCXX\_3.4.5) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ofstream<wchar\_t, char\_traits<wchar\_t> >::open(char const\*, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ofstream<wchar\_t, char\_traits<wchar\_t> >::close()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ofstream<wchar\_t, char\_traits<wchar\_t> >::is\_open()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ofstream<wchar\_t, char\_traits<wchar\_t> >::basic\_ofstream(char const\*, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ofstream<wchar\_t, char\_traits<wchar\_t> >::basic\_ofstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ofstream<wchar\_t, char\_traits<wchar\_t> >::basic\_ofstream(char const\*, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ofstream<wchar\_t, char\_traits<wchar\_t> >::basic\_ofstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ofstream<wchar\_t, char\_traits<wchar\_t> >::~basic\_ofstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ofstream<wchar\_t, char\_traits<wchar\_t> >::~basic\_ofstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ofstream<wchar\_t, char\_traits<wchar\_t> >::~basic\_ofstream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::basic\_ofstream<wchar\_t, std::char\_traits<wchar\_t> > specified in [Table 16-220](#ID_TBL_45_LIBSTDCXX_45_CLAVA_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-220 libstdcxx - Class basic\_ofstream<wchar\_t, char\_traits<wchar\_t> > Data Interfaces**

|  |
| --- |
| typeinfo for basic\_ofstream<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for basic\_ofstream<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| VTT for basic\_ofstream<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for basic\_ofstream<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.82 Class basic\_streambuf<char, char\_traits<char> >**

#### 16.1.82.1 Class data for basic\_streambuf<char, char\_traits<char> >

The virtual table for the std::basic\_streambuf<char, std::char\_traits<char> > class is described by [Table 16-221](#ID_CLS_45_24315_45_0)

**Table 16-221 Primary vtable for basic\_streambuf<char, char\_traits<char> >**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for basic\_streambuf<char, char\_traits<char> > |
| vfunc[0]: | basic\_streambuf<char, char\_traits<char> >::~basic\_streambuf() |
| vfunc[1]: | basic\_streambuf<char, char\_traits<char> >::~basic\_streambuf() |
| vfunc[2]: | basic\_streambuf<char, char\_traits<char> >::imbue(locale const&) |
| vfunc[3]: | See architecture specific part. |
| vfunc[4]: | See architecture specific part. |
| vfunc[5]: | basic\_streambuf<char, char\_traits<char> >::seekpos(fpos<\_\_mbstate\_t>, \_Ios\_Openmode) |
| vfunc[6]: | basic\_streambuf<char, char\_traits<char> >::sync() |
| vfunc[7]: | basic\_streambuf<char, char\_traits<char> >::showmanyc() |
| vfunc[8]: | See architecture specific part. |
| vfunc[9]: | basic\_streambuf<char, char\_traits<char> >::underflow() |
| vfunc[10]: | basic\_streambuf<char, char\_traits<char> >::uflow() |
| vfunc[11]: | basic\_streambuf<char, char\_traits<char> >::pbackfail(int) |
| vfunc[12]: | See architecture specific part. |
| vfunc[13]: | basic\_streambuf<char, char\_traits<char> >::overflow(int) |

The Run Time Type Information for the std::basic\_streambuf<char, std::char\_traits<char> > class is described by [Table 16-222](#ID_RTTI_45_23990)

**Table 16-222 typeinfo for basic\_streambuf<char, char\_traits<char> >**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_class\_type\_info |
| Name | typeinfo name for basic\_streambuf<char, char\_traits<char> > |

#### 16.1.82.2 Interfaces for Class basic\_streambuf<char, char\_traits<char> >

An LSB conforming implementation shall provide the generic methods for Class std::basic\_streambuf<char, std::char\_traits<char> > specified in [Table 16-223](#ID_TBL_45_LIBSTDCXX_45_CLAVB_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-223 libstdcxx - Class basic\_streambuf<char, char\_traits<char> > Function Interfaces**

|  |
| --- |
| basic\_streambuf<char, char\_traits<char> >::gptr() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::pptr() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::eback() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::egptr() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::epptr() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::pbase() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::getloc() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::pubseekpos(fpos<\_\_mbstate\_t>, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::setg(char\*, char\*, char\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::setp(char\*, char\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::sync()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::gbump(int)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::imbue(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::pbump(int)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::sgetc()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::sputc(char)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::uflow()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::sbumpc()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::snextc()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::pubsync()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::seekpos(fpos<\_\_mbstate\_t>, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::sungetc()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::in\_avail()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::overflow(int)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::pubimbue(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::pbackfail(int)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::showmanyc()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::sputbackc(char)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::underflow()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::basic\_streambuf(basic\_streambuf<char, char\_traits<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::basic\_streambuf()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::basic\_streambuf(basic\_streambuf<char, char\_traits<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::basic\_streambuf()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::~basic\_streambuf()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::~basic\_streambuf()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::~basic\_streambuf()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<char, char\_traits<char> >::operator=(basic\_streambuf<char, char\_traits<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::basic\_streambuf<char, std::char\_traits<char> > specified in [Table 16-224](#ID_TBL_45_LIBSTDCXX_45_CLAVB_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-224 libstdcxx - Class basic\_streambuf<char, char\_traits<char> > Data Interfaces**

|  |
| --- |
| typeinfo for basic\_streambuf<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for basic\_streambuf<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for basic\_streambuf<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.83 Class basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >**

#### 16.1.83.1 Class data for basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >

The virtual table for the std::basic\_streambuf<wchar\_t, std::char\_traits<wchar\_t> > class is described by [Table 16-225](#ID_CLS_45_24314_45_0)

**Table 16-225 Primary vtable for basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for basic\_streambuf<wchar\_t, char\_traits<wchar\_t> > |
| vfunc[0]: | basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::~basic\_streambuf() |
| vfunc[1]: | basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::~basic\_streambuf() |
| vfunc[2]: | basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::imbue(locale const&) |
| vfunc[3]: | See architecture specific part. |
| vfunc[4]: | See architecture specific part. |
| vfunc[5]: | basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::seekpos(fpos<\_\_mbstate\_t>, \_Ios\_Openmode) |
| vfunc[6]: | basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::sync() |
| vfunc[7]: | basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::showmanyc() |
| vfunc[8]: | See architecture specific part. |
| vfunc[9]: | basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::underflow() |
| vfunc[10]: | basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::uflow() |
| vfunc[11]: | basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::pbackfail(unsigned int) |
| vfunc[12]: | See architecture specific part. |
| vfunc[13]: | basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::overflow(unsigned int) |

The Run Time Type Information for the std::basic\_streambuf<wchar\_t, std::char\_traits<wchar\_t> > class is described by [Table 16-226](#ID_RTTI_45_23991)

**Table 16-226 typeinfo for basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_class\_type\_info |
| Name | typeinfo name for basic\_streambuf<wchar\_t, char\_traits<wchar\_t> > |

#### 16.1.83.2 Interfaces for Class basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >

An LSB conforming implementation shall provide the generic methods for Class std::basic\_streambuf<wchar\_t, std::char\_traits<wchar\_t> > specified in [Table 16-227](#ID_TBL_45_LIBSTDCXX_45_CLAVC_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-227 libstdcxx - Class basic\_streambuf<wchar\_t, char\_traits<wchar\_t> > Function Interfaces**

|  |
| --- |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::gptr() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::pptr() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::eback() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::egptr() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::epptr() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::pbase() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::getloc() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::pubseekpos(fpos<\_\_mbstate\_t>, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::setg(wchar\_t\*, wchar\_t\*, wchar\_t\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::setp(wchar\_t\*, wchar\_t\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::sync()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::gbump(int)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::imbue(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::pbump(int)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::sgetc()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::sputc(wchar\_t)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::uflow()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::sbumpc()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::snextc()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::pubsync()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::seekpos(fpos<\_\_mbstate\_t>, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::sungetc()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::in\_avail()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::overflow(unsigned int)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::pubimbue(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::pbackfail(unsigned int)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::showmanyc()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::sputbackc(wchar\_t)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::underflow()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::basic\_streambuf(basic\_streambuf<wchar\_t, char\_traits<wchar\_t> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::basic\_streambuf()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::basic\_streambuf(basic\_streambuf<wchar\_t, char\_traits<wchar\_t> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::basic\_streambuf()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::~basic\_streambuf()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::~basic\_streambuf()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::~basic\_streambuf()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::operator=(basic\_streambuf<wchar\_t, char\_traits<wchar\_t> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.84 Class basic\_filebuf<char, char\_traits<char> >**

#### 16.1.84.1 Class data for basic\_filebuf<char, char\_traits<char> >

The virtual table for the std::basic\_filebuf<char, std::char\_traits<char> > class is described by [Table 16-228](#ID_CLS_45_24240_45_0)

**Table 16-228 Primary vtable for basic\_filebuf<char, char\_traits<char> >**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for basic\_filebuf<char, char\_traits<char> > |
| vfunc[0]: | basic\_filebuf<char, char\_traits<char> >::~basic\_filebuf() |
| vfunc[1]: | basic\_filebuf<char, char\_traits<char> >::~basic\_filebuf() |
| vfunc[2]: | basic\_filebuf<char, char\_traits<char> >::imbue(locale const&) |
| vfunc[3]: | See architecture specific part. |
| vfunc[4]: | See architecture specific part. |
| vfunc[5]: | basic\_filebuf<char, char\_traits<char> >::seekpos(fpos<\_\_mbstate\_t>, \_Ios\_Openmode) |
| vfunc[6]: | basic\_filebuf<char, char\_traits<char> >::sync() |
| vfunc[7]: | basic\_filebuf<char, char\_traits<char> >::showmanyc() |
| vfunc[8]: | See architecture specific part. |
| vfunc[9]: | basic\_filebuf<char, char\_traits<char> >::underflow() |
| vfunc[10]: | basic\_streambuf<char, char\_traits<char> >::uflow() |
| vfunc[11]: | basic\_filebuf<char, char\_traits<char> >::pbackfail(int) |
| vfunc[12]: | See architecture specific part. |
| vfunc[13]: | basic\_filebuf<char, char\_traits<char> >::overflow(int) |

The Run Time Type Information for the std::basic\_filebuf<char, std::char\_traits<char> > class is described by [Table 16-229](#ID_RTTI_45_23988)

**Table 16-229 typeinfo for basic\_filebuf<char, char\_traits<char> >**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for basic\_filebuf<char, char\_traits<char> > |

#### 16.1.84.2 Interfaces for Class basic\_filebuf<char, char\_traits<char> >

An LSB conforming implementation shall provide the generic methods for Class std::basic\_filebuf<char, std::char\_traits<char> > specified in [Table 16-230](#ID_TBL_45_LIBSTDCXX_45_CLAVD_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-230 libstdcxx - Class basic\_filebuf<char, char\_traits<char> > Function Interfaces**

|  |
| --- |
| basic\_filebuf<char, char\_traits<char> >::is\_open() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<char, char\_traits<char> >::\_M\_create\_pback()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<char, char\_traits<char> >::\_M\_destroy\_pback()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<char, char\_traits<char> >::\_M\_terminate\_output()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<char, char\_traits<char> >::\_M\_destroy\_internal\_buffer()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<char, char\_traits<char> >::\_M\_allocate\_internal\_buffer()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<char, char\_traits<char> >::open(char const\*, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<char, char\_traits<char> >::sync()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<char, char\_traits<char> >::close()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<char, char\_traits<char> >::imbue(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<char, char\_traits<char> >::seekpos(fpos<\_\_mbstate\_t>, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<char, char\_traits<char> >::overflow(int)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<char, char\_traits<char> >::pbackfail(int)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<char, char\_traits<char> >::showmanyc()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<char, char\_traits<char> >::underflow()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<char, char\_traits<char> >::basic\_filebuf()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<char, char\_traits<char> >::basic\_filebuf()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<char, char\_traits<char> >::~basic\_filebuf()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<char, char\_traits<char> >::~basic\_filebuf()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<char, char\_traits<char> >::~basic\_filebuf()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::basic\_filebuf<char, std::char\_traits<char> > specified in [Table 16-231](#ID_TBL_45_LIBSTDCXX_45_CLAVD_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-231 libstdcxx - Class basic\_filebuf<char, char\_traits<char> > Data Interfaces**

|  |
| --- |
| typeinfo for basic\_filebuf<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for basic\_filebuf<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for basic\_filebuf<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.85 Class basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >**

#### 16.1.85.1 Class data for basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >

The virtual table for the std::basic\_filebuf<wchar\_t, std::char\_traits<wchar\_t> > class is described by [Table 16-232](#ID_CLS_45_24239_45_0)

**Table 16-232 Primary vtable for basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for basic\_filebuf<wchar\_t, char\_traits<wchar\_t> > |
| vfunc[0]: | basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >::~basic\_filebuf() |
| vfunc[1]: | basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >::~basic\_filebuf() |
| vfunc[2]: | basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >::imbue(locale const&) |
| vfunc[3]: | See architecture specific part. |
| vfunc[4]: | See architecture specific part. |
| vfunc[5]: | basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >::seekpos(fpos<\_\_mbstate\_t>, \_Ios\_Openmode) |
| vfunc[6]: | basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >::sync() |
| vfunc[7]: | basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >::showmanyc() |
| vfunc[8]: | See architecture specific part. |
| vfunc[9]: | basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >::underflow() |
| vfunc[10]: | basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >::uflow() |
| vfunc[11]: | basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >::pbackfail(unsigned int) |
| vfunc[12]: | See architecture specific part. |
| vfunc[13]: | basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >::overflow(unsigned int) |

The Run Time Type Information for the std::basic\_filebuf<wchar\_t, std::char\_traits<wchar\_t> > class is described by [Table 16-233](#ID_RTTI_45_23989)

**Table 16-233 typeinfo for basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for basic\_filebuf<wchar\_t, char\_traits<wchar\_t> > |

#### 16.1.85.2 Interfaces for Class basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >

An LSB conforming implementation shall provide the generic methods for Class std::basic\_filebuf<wchar\_t, std::char\_traits<wchar\_t> > specified in [Table 16-234](#ID_TBL_45_LIBSTDCXX_45_CLAVE_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-234 libstdcxx - Class basic\_filebuf<wchar\_t, char\_traits<wchar\_t> > Function Interfaces**

|  |
| --- |
| basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >::is\_open() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >::\_M\_create\_pback()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >::\_M\_destroy\_pback()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >::\_M\_terminate\_output()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >::\_M\_destroy\_internal\_buffer()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >::\_M\_allocate\_internal\_buffer()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >::open(char const\*, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >::sync()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >::close()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >::imbue(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >::seekpos(fpos<\_\_mbstate\_t>, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >::overflow(unsigned int)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >::pbackfail(unsigned int)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >::showmanyc()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >::underflow()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >::basic\_filebuf()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >::basic\_filebuf()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >::~basic\_filebuf()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >::~basic\_filebuf()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >::~basic\_filebuf()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::basic\_istream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_istream<wchar\_t, char\_traits<wchar\_t> >::basic\_istream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::basic\_ostream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ostream<wchar\_t, char\_traits<wchar\_t> >::basic\_ostream()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::basic\_filebuf<wchar\_t, std::char\_traits<wchar\_t> > specified in [Table 16-235](#ID_TBL_45_LIBSTDCXX_45_CLAVE_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-235 libstdcxx - Class basic\_filebuf<wchar\_t, char\_traits<wchar\_t> > Data Interfaces**

|  |
| --- |
| typeinfo for basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for basic\_filebuf<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.86 Class ios\_base**

#### 16.1.86.1 Class data for ios\_base

The virtual table for the std::ios\_base class is described by [Table 16-236](#ID_CLS_45_29891_45_0)

**Table 16-236 Primary vtable for ios\_base**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for ios\_base |
| vfunc[0]: | ios\_base::~ios\_base() |
| vfunc[1]: | ios\_base::~ios\_base() |

The Run Time Type Information for the std::ios\_base class is described by [Table 16-237](#ID_RTTI_45_24001)

**Table 16-237 typeinfo for ios\_base**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_class\_type\_info |
| Name | typeinfo name for ios\_base |

#### 16.1.86.2 Interfaces for Class ios\_base

An LSB conforming implementation shall provide the generic methods for Class std::ios\_base specified in [Table 16-238](#ID_TBL_45_LIBSTDCXX_45_CLAVF_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-238 libstdcxx - Class ios\_base Function Interfaces**

|  |
| --- |
| ios\_base::\_M\_grow\_words(int, bool)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::sync\_with\_stdio(bool)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::\_M\_call\_callbacks(ios\_base::event)(GLIBCXX\_3.4.6) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::register\_callback(void (\*)(ios\_base::event, ios\_base&, int), int)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::\_M\_dispose\_callbacks()(GLIBCXX\_3.4.6) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::Init::Init()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::Init::Init()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::Init::~Init()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::Init::~Init()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::imbue(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::xalloc()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::\_M\_init()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::failure::failure(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::failure::failure(basic\_string<char, char\_traits<char>, allocator<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::failure::~failure()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::failure::~failure()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::failure::~failure()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::ios\_base()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::ios\_base()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::~ios\_base()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::~ios\_base()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::~ios\_base()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::ios\_base specified in [Table 16-239](#ID_TBL_45_LIBSTDCXX_45_CLAVF_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-239 libstdcxx - Class ios\_base Data Interfaces**

|  |
| --- |
| ios\_base::floatfield(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::scientific(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::adjustfield(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::in(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::app(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::ate(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::beg(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::cur(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::dec(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::end(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::hex(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::oct(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::out(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::left(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::fixed(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::right(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::trunc(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::badbit(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::binary(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::eofbit(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::skipws(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::failbit(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::goodbit(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::showpos(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::unitbuf(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::internal(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::showbase(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::basefield(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::boolalpha(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::showpoint(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ios\_base::uppercase(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for ios\_base(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for ios\_base(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for ios\_base(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.87 Class basic\_ios<char, char\_traits<char> >**

#### 16.1.87.1 Class data for basic\_ios<char, char\_traits<char> >

The virtual table for the std::basic\_ios<char, std::char\_traits<char> > class is described by [Table 16-240](#ID_CLS_45_24251_45_0)

**Table 16-240 Primary vtable for basic\_ios<char, char\_traits<char> >**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for basic\_ios<char, char\_traits<char> > |
| vfunc[0]: | basic\_ios<char, char\_traits<char> >::~basic\_ios() |
| vfunc[1]: | basic\_ios<char, char\_traits<char> >::~basic\_ios() |

#### 16.1.87.2 Interfaces for Class basic\_ios<char, char\_traits<char> >

An LSB conforming implementation shall provide the generic methods for Class std::basic\_ios<char, std::char\_traits<char> > specified in [Table 16-241](#ID_TBL_45_LIBSTDCXX_45_CLAVG_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-241 libstdcxx - Class basic\_ios<char, char\_traits<char> > Function Interfaces**

|  |
| --- |
| basic\_ios<char, char\_traits<char> >::exceptions() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<char, char\_traits<char> >::bad() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<char, char\_traits<char> >::eof() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<char, char\_traits<char> >::tie() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<char, char\_traits<char> >::fail() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<char, char\_traits<char> >::fill() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<char, char\_traits<char> >::good() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<char, char\_traits<char> >::rdbuf() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<char, char\_traits<char> >::widen(char) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<char, char\_traits<char> >::narrow(char, char) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<char, char\_traits<char> >::rdstate() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<char, char\_traits<char> >::operator void\*() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<char, char\_traits<char> >::operator!() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<char, char\_traits<char> >::exceptions(\_Ios\_Iostate)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<char, char\_traits<char> >::\_M\_setstate(\_Ios\_Iostate)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<char, char\_traits<char> >::tie(basic\_ostream<char, char\_traits<char> >\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<char, char\_traits<char> >::fill(char)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<char, char\_traits<char> >::init(basic\_streambuf<char, char\_traits<char> >\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<char, char\_traits<char> >::clear(\_Ios\_Iostate)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<char, char\_traits<char> >::imbue(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<char, char\_traits<char> >::rdbuf(basic\_streambuf<char, char\_traits<char> >\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<char, char\_traits<char> >::copyfmt(basic\_ios<char, char\_traits<char> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<char, char\_traits<char> >::setstate(\_Ios\_Iostate)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<char, char\_traits<char> >::basic\_ios(basic\_streambuf<char, char\_traits<char> >\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<char, char\_traits<char> >::basic\_ios()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<char, char\_traits<char> >::basic\_ios(basic\_streambuf<char, char\_traits<char> >\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<char, char\_traits<char> >::basic\_ios()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<char, char\_traits<char> >::~basic\_ios()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<char, char\_traits<char> >::~basic\_ios()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<char, char\_traits<char> >::~basic\_ios()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::basic\_ios<char, std::char\_traits<char> > specified in [Table 16-242](#ID_TBL_45_LIBSTDCXX_45_CLAVG_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-242 libstdcxx - Class basic\_ios<char, char\_traits<char> > Data Interfaces**

|  |
| --- |
| typeinfo for basic\_ios<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for basic\_ios<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for basic\_ios<char, char\_traits<char> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.88 Class basic\_ios<wchar\_t, char\_traits<wchar\_t> >**

#### 16.1.88.1 Class data for basic\_ios<wchar\_t, char\_traits<wchar\_t> >

The virtual table for the std::basic\_ios<wchar\_t, std::char\_traits<wchar\_t> > class is described by [Table 16-243](#ID_CLS_45_24250_45_0)

**Table 16-243 Primary vtable for basic\_ios<wchar\_t, char\_traits<wchar\_t> >**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for basic\_ios<wchar\_t, char\_traits<wchar\_t> > |
| vfunc[0]: | basic\_ios<wchar\_t, char\_traits<wchar\_t> >::~basic\_ios() |
| vfunc[1]: | basic\_ios<wchar\_t, char\_traits<wchar\_t> >::~basic\_ios() |

The Run Time Type Information for the std::basic\_ios<wchar\_t, std::char\_traits<wchar\_t> > class is described by [Table 16-244](#ID_RTTI_45_24004)

**Table 16-244 typeinfo for basic\_ios<wchar\_t, char\_traits<wchar\_t> >**

|  |  |  |
| --- | --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |  |
| Name | typeinfo name for basic\_ios<wchar\_t, char\_traits<wchar\_t> > |  |
| flags: | 8 |  |
| basetype: | typeinfo for ios\_base | 1026 |

#### 16.1.88.2 Interfaces for Class basic\_ios<wchar\_t, char\_traits<wchar\_t> >

An LSB conforming implementation shall provide the generic methods for Class std::basic\_ios<wchar\_t, std::char\_traits<wchar\_t> > specified in [Table 16-245](#ID_TBL_45_LIBSTDCXX_45_CLAVH_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-245 libstdcxx - Class basic\_ios<wchar\_t, char\_traits<wchar\_t> > Function Interfaces**

|  |
| --- |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::exceptions() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::bad() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::eof() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::tie() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::fail() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::fill() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::good() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::rdbuf() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::widen(char) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::narrow(wchar\_t, char) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::rdstate() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::operator void\*() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::operator!() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::exceptions(\_Ios\_Iostate)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::\_M\_setstate(\_Ios\_Iostate)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::tie(basic\_ostream<wchar\_t, char\_traits<wchar\_t> >\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::fill(wchar\_t)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::init(basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::clear(\_Ios\_Iostate)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::imbue(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::rdbuf(basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::copyfmt(basic\_ios<wchar\_t, char\_traits<wchar\_t> > const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::setstate(\_Ios\_Iostate)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::basic\_ios(basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::basic\_ios()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::basic\_ios(basic\_streambuf<wchar\_t, char\_traits<wchar\_t> >\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::basic\_ios()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::~basic\_ios()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::~basic\_ios()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| basic\_ios<wchar\_t, char\_traits<wchar\_t> >::~basic\_ios()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::basic\_ios<wchar\_t, std::char\_traits<wchar\_t> > specified in [Table 16-246](#ID_TBL_45_LIBSTDCXX_45_CLAVH_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-246 libstdcxx - Class basic\_ios<wchar\_t, char\_traits<wchar\_t> > Data Interfaces**

|  |
| --- |
| typeinfo for basic\_ios<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for basic\_ios<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for basic\_ios<wchar\_t, char\_traits<wchar\_t> >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.89 Class ios\_base::failure**

#### 16.1.89.1 Class data for ios\_base::failure

The virtual table for the std::ios\_base::failure class is described by [Table 16-247](#ID_CLS_45_24252_45_0)

**Table 16-247 Primary vtable for ios\_base::failure**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for ios\_base::failure |
| vfunc[0]: | ios\_base::failure::~failure() |
| vfunc[1]: | ios\_base::failure::~failure() |
| vfunc[2]: | ios\_base::failure::what() const |

The Run Time Type Information for the std::ios\_base::failure class is described by [Table 16-248](#ID_RTTI_45_24059)

**Table 16-248 typeinfo for ios\_base::failure**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for ios\_base::failure |

#### 16.1.89.2 Interfaces for Class ios\_base::failure

An LSB conforming implementation shall provide the generic methods for Class std::ios\_base::failure specified in [Table 16-249](#ID_TBL_45_LIBSTDCXX_45_CLAVI_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-249 libstdcxx - Class ios\_base::failure Function Interfaces**

|  |
| --- |
| ios\_base::failure::what() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::ios\_base::failure specified in [Table 16-250](#ID_TBL_45_LIBSTDCXX_45_CLAVI_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-250 libstdcxx - Class ios\_base::failure Data Interfaces**

|  |
| --- |
| typeinfo for ios\_base::failure(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for ios\_base::failure(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for ios\_base::failure(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.90 Class \_\_timepunct<char>**

#### 16.1.90.1 Class data for \_\_timepunct<char>

The virtual table for the std::\_\_timepunct<char> class is described by [Table 16-251](#ID_CLS_45_24291_45_0)

**Table 16-251 Primary vtable for \_\_timepunct<char>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for \_\_timepunct<char> |
| vfunc[0]: | \_\_timepunct<char>::~\_\_timepunct() |
| vfunc[1]: | \_\_timepunct<char>::~\_\_timepunct() |

The Run Time Type Information for the std::\_\_timepunct<char> class is described by [Table 16-252](#ID_RTTI_45_24085)

**Table 16-252 typeinfo for \_\_timepunct<char>**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for \_\_timepunct<char> |

#### 16.1.90.2 Interfaces for Class \_\_timepunct<char>

An LSB conforming implementation shall provide the generic methods for Class std::\_\_timepunct<char> specified in [Table 16-253](#ID_TBL_45_LIBSTDCXX_45_CLAVJ_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-253 libstdcxx - Class \_\_timepunct<char> Function Interfaces**

|  |
| --- |
| \_\_timepunct<char>::\_M\_am\_pm\_format(char const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_timepunct<char>::\_M\_date\_formats(char const\*\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_timepunct<char>::\_M\_time\_formats(char const\*\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_timepunct<char>::\_M\_days\_abbreviated(char const\*\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_timepunct<char>::\_M\_date\_time\_formats(char const\*\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_timepunct<char>::\_M\_months\_abbreviated(char const\*\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_timepunct<char>::\_M\_days(char const\*\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_timepunct<char>::\_M\_am\_pm(char const\*\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_timepunct<char>::\_M\_months(char const\*\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_timepunct<wchar\_t>::\_M\_am\_pm\_format(wchar\_t const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_timepunct<char>::\_M\_initialize\_timepunct(\_\_locale\_struct\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_timepunct<char>::~\_\_timepunct()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_timepunct<char>::~\_\_timepunct()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_timepunct<char>::~\_\_timepunct()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bool has\_facet<\_\_timepunct<char> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::\_\_timepunct<char> specified in [Table 16-254](#ID_TBL_45_LIBSTDCXX_45_CLAVJ_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-254 libstdcxx - Class \_\_timepunct<char> Data Interfaces**

|  |
| --- |
| guard variable for \_\_timepunct<char>::id(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_timepunct<char>::id(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for \_\_timepunct<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for \_\_timepunct<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for \_\_timepunct<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.91 Class \_\_timepunct<wchar\_t>**

#### 16.1.91.1 Class data for \_\_timepunct<wchar\_t>

The virtual table for the std::\_\_timepunct<wchar\_t> class is described by [Table 16-255](#ID_CLS_45_24290_45_0)

**Table 16-255 Primary vtable for \_\_timepunct<wchar\_t>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for \_\_timepunct<wchar\_t> |
| vfunc[0]: | \_\_timepunct<wchar\_t>::~\_\_timepunct() |
| vfunc[1]: | \_\_timepunct<wchar\_t>::~\_\_timepunct() |

The Run Time Type Information for the std::\_\_timepunct<wchar\_t> class is described by [Table 16-256](#ID_RTTI_45_24084)

**Table 16-256 typeinfo for \_\_timepunct<wchar\_t>**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for \_\_timepunct<wchar\_t> |

#### 16.1.91.2 Interfaces for Class \_\_timepunct<wchar\_t>

An LSB conforming implementation shall provide the generic methods for Class std::\_\_timepunct<wchar\_t> specified in [Table 16-257](#ID_TBL_45_LIBSTDCXX_45_CLAVK_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-257 libstdcxx - Class \_\_timepunct<wchar\_t> Function Interfaces**

|  |
| --- |
| \_\_timepunct<wchar\_t>::\_M\_date\_formats(wchar\_t const\*\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_timepunct<wchar\_t>::\_M\_time\_formats(wchar\_t const\*\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_timepunct<wchar\_t>::\_M\_days\_abbreviated(wchar\_t const\*\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_timepunct<wchar\_t>::\_M\_date\_time\_formats(wchar\_t const\*\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_timepunct<wchar\_t>::\_M\_months\_abbreviated(wchar\_t const\*\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_timepunct<wchar\_t>::\_M\_days(wchar\_t const\*\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_timepunct<wchar\_t>::\_M\_am\_pm(wchar\_t const\*\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_timepunct<wchar\_t>::\_M\_months(wchar\_t const\*\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_timepunct<wchar\_t>::\_M\_initialize\_timepunct(\_\_locale\_struct\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_timepunct<wchar\_t>::~\_\_timepunct()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_timepunct<wchar\_t>::~\_\_timepunct()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_timepunct<wchar\_t>::~\_\_timepunct()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bool has\_facet<\_\_timepunct<wchar\_t> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::\_\_timepunct<wchar\_t> specified in [Table 16-258](#ID_TBL_45_LIBSTDCXX_45_CLAVK_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-258 libstdcxx - Class \_\_timepunct<wchar\_t> Data Interfaces**

|  |
| --- |
| guard variable for \_\_timepunct<wchar\_t>::id(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| \_\_timepunct<wchar\_t>::id(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for \_\_timepunct<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for \_\_timepunct<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for \_\_timepunct<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.92 Class messages\_base**

#### 16.1.92.1 Class data for messages\_base

The Run Time Type Information for the std::messages\_base class is described by [Table 16-259](#ID_RTTI_45_24011)

**Table 16-259 typeinfo for messages\_base**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_class\_type\_info |
| Name | typeinfo name for messages\_base |

#### 16.1.92.2 Interfaces for Class messages\_base

No external methods are defined for libstdcxx - Class std::messages\_base in this part of the specification. See also the relevant architecture specific part of this specification.

An LSB conforming implementation shall provide the generic data interfaces for Class std::messages\_base specified in [Table 16-260](#ID_TBL_45_LIBSTDCXX_45_CLAVL_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-260 libstdcxx - Class messages\_base Data Interfaces**

|  |
| --- |
| typeinfo for messages\_base(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for messages\_base(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.93 Class messages<char>**

#### 16.1.93.1 Class data for messages<char>

The virtual table for the std::messages<char> class is described by [Table 16-261](#ID_CLS_45_24285_45_0)

**Table 16-261 Primary vtable for messages<char>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for messages<char> |
| vfunc[0]: | messages<char>::~messages() |
| vfunc[1]: | messages<char>::~messages() |
| vfunc[2]: | messages<char>::do\_open(basic\_string<char, char\_traits<char>, allocator<char> > const&, locale const&) const |
| vfunc[3]: | messages<char>::do\_get(int, int, int, basic\_string<char, char\_traits<char>, allocator<char> > const&) const |
| vfunc[4]: | messages<char>::do\_close(int) const |

#### 16.1.93.2 Interfaces for Class messages<char>

An LSB conforming implementation shall provide the generic methods for Class std::messages<char> specified in [Table 16-262](#ID_TBL_45_LIBSTDCXX_45_CLAVM_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-262 libstdcxx - Class messages<char> Function Interfaces**

|  |
| --- |
| messages<char>::\_M\_convert\_to\_char(basic\_string<char, char\_traits<char>, allocator<char> > const&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| messages<char>::\_M\_convert\_from\_char(char\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| messages<char>::get(int, int, int, basic\_string<char, char\_traits<char>, allocator<char> > const&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| messages<char>::open(basic\_string<char, char\_traits<char>, allocator<char> > const&, locale const&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| messages<char>::open(basic\_string<char, char\_traits<char>, allocator<char> > const&, locale const&, char const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| messages<char>::close(int) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| messages<char>::do\_get(int, int, int, basic\_string<char, char\_traits<char>, allocator<char> > const&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| messages<char>::do\_open(basic\_string<char, char\_traits<char>, allocator<char> > const&, locale const&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| messages<char>::do\_close(int) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| messages<char>::~messages()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| messages<char>::~messages()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| messages<char>::~messages()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::messages<char> specified in [Table 16-263](#ID_TBL_45_LIBSTDCXX_45_CLAVM_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-263 libstdcxx - Class messages<char> Data Interfaces**

|  |
| --- |
| guard variable for messages<char>::id(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| messages<char>::id(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for messages<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for messages<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for messages<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.94 Class messages<wchar\_t>**

#### 16.1.94.1 Class data for messages<wchar\_t>

The virtual table for the std::messages<wchar\_t> class is described by [Table 16-264](#ID_CLS_45_24284_45_0)

**Table 16-264 Primary vtable for messages<wchar\_t>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for messages<wchar\_t> |
| vfunc[0]: | messages<wchar\_t>::~messages() |
| vfunc[1]: | messages<wchar\_t>::~messages() |
| vfunc[2]: | messages<wchar\_t>::do\_open(basic\_string<char, char\_traits<char>, allocator<char> > const&, locale const&) const |
| vfunc[3]: | messages<wchar\_t>::do\_get(int, int, int, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&) const |
| vfunc[4]: | messages<wchar\_t>::do\_close(int) const |

#### 16.1.94.2 Interfaces for Class messages<wchar\_t>

An LSB conforming implementation shall provide the generic methods for Class std::messages<wchar\_t> specified in [Table 16-265](#ID_TBL_45_LIBSTDCXX_45_CLAVN_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-265 libstdcxx - Class messages<wchar\_t> Function Interfaces**

|  |
| --- |
| messages<wchar\_t>::\_M\_convert\_to\_char(basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| messages<wchar\_t>::\_M\_convert\_from\_char(char\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| messages<wchar\_t>::get(int, int, int, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| messages<wchar\_t>::open(basic\_string<char, char\_traits<char>, allocator<char> > const&, locale const&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| messages<wchar\_t>::open(basic\_string<char, char\_traits<char>, allocator<char> > const&, locale const&, char const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| messages<wchar\_t>::close(int) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| messages<wchar\_t>::do\_get(int, int, int, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| messages<wchar\_t>::do\_open(basic\_string<char, char\_traits<char>, allocator<char> > const&, locale const&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| messages<wchar\_t>::do\_close(int) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| messages<wchar\_t>::~messages()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| messages<wchar\_t>::~messages()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| messages<wchar\_t>::~messages()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::messages<wchar\_t> specified in [Table 16-266](#ID_TBL_45_LIBSTDCXX_45_CLAVN_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-266 libstdcxx - Class messages<wchar\_t> Data Interfaces**

|  |
| --- |
| guard variable for messages<wchar\_t>::id(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| messages<wchar\_t>::id(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for messages<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for messages<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for messages<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.95 Class messages\_byname<char>**

#### 16.1.95.1 Class data for messages\_byname<char>

The virtual table for the std::messages\_byname<char> class is described by [Table 16-267](#ID_CLS_45_24272_45_0)

**Table 16-267 Primary vtable for messages\_byname<char>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for messages\_byname<char> |
| vfunc[0]: | messages\_byname<char>::~messages\_byname() |
| vfunc[1]: | messages\_byname<char>::~messages\_byname() |
| vfunc[2]: | messages<char>::do\_open(basic\_string<char, char\_traits<char>, allocator<char> > const&, locale const&) const |
| vfunc[3]: | messages<char>::do\_get(int, int, int, basic\_string<char, char\_traits<char>, allocator<char> > const&) const |
| vfunc[4]: | messages<char>::do\_close(int) const |

The Run Time Type Information for the std::messages\_byname<char> class is described by [Table 16-268](#ID_RTTI_45_24074)

**Table 16-268 typeinfo for messages\_byname<char>**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for messages\_byname<char> |

#### 16.1.95.2 Interfaces for Class messages\_byname<char>

An LSB conforming implementation shall provide the generic methods for Class std::messages\_byname<char> specified in [Table 16-269](#ID_TBL_45_LIBSTDCXX_45_CLAVO_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-269 libstdcxx - Class messages\_byname<char> Function Interfaces**

|  |
| --- |
| messages\_byname<char>::~messages\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| messages\_byname<char>::~messages\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| messages\_byname<char>::~messages\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::messages\_byname<char> specified in [Table 16-270](#ID_TBL_45_LIBSTDCXX_45_CLAVO_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-270 libstdcxx - Class messages\_byname<char> Data Interfaces**

|  |
| --- |
| typeinfo for messages\_byname<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for messages\_byname<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for messages\_byname<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.96 Class messages\_byname<wchar\_t>**

#### 16.1.96.1 Class data for messages\_byname<wchar\_t>

The virtual table for the std::messages\_byname<wchar\_t> class is described by [Table 16-271](#ID_CLS_45_24258_45_0)

**Table 16-271 Primary vtable for messages\_byname<wchar\_t>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for messages\_byname<wchar\_t> |
| vfunc[0]: | messages\_byname<wchar\_t>::~messages\_byname() |
| vfunc[1]: | messages\_byname<wchar\_t>::~messages\_byname() |
| vfunc[2]: | messages<wchar\_t>::do\_open(basic\_string<char, char\_traits<char>, allocator<char> > const&, locale const&) const |
| vfunc[3]: | messages<wchar\_t>::do\_get(int, int, int, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&) const |
| vfunc[4]: | messages<wchar\_t>::do\_close(int) const |

The Run Time Type Information for the std::messages\_byname<wchar\_t> class is described by [Table 16-272](#ID_RTTI_45_24062)

**Table 16-272 typeinfo for messages\_byname<wchar\_t>**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for messages\_byname<wchar\_t> |

#### 16.1.96.2 Interfaces for Class messages\_byname<wchar\_t>

An LSB conforming implementation shall provide the generic methods for Class std::messages\_byname<wchar\_t> specified in [Table 16-273](#ID_TBL_45_LIBSTDCXX_45_CLAVP_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-273 libstdcxx - Class messages\_byname<wchar\_t> Function Interfaces**

|  |
| --- |
| messages\_byname<wchar\_t>::~messages\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| messages\_byname<wchar\_t>::~messages\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| messages\_byname<wchar\_t>::~messages\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::messages\_byname<wchar\_t> specified in [Table 16-274](#ID_TBL_45_LIBSTDCXX_45_CLAVP_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-274 libstdcxx - Class messages\_byname<wchar\_t> Data Interfaces**

|  |
| --- |
| typeinfo for messages\_byname<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for messages\_byname<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for messages\_byname<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.97 Class numpunct<char>**

#### 16.1.97.1 Class data for numpunct<char>

The virtual table for the std::numpunct<char> class is described by [Table 16-275](#ID_CLS_45_24295_45_0)

**Table 16-275 Primary vtable for numpunct<char>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for numpunct<char> |
| vfunc[0]: | numpunct<char>::~numpunct() |
| vfunc[1]: | numpunct<char>::~numpunct() |
| vfunc[2]: | numpunct<char>::do\_decimal\_point() const |
| vfunc[3]: | numpunct<char>::do\_thousands\_sep() const |
| vfunc[4]: | numpunct<char>::do\_grouping() const |
| vfunc[5]: | numpunct<char>::do\_truename() const |
| vfunc[6]: | numpunct<char>::do\_falsename() const |

The Run Time Type Information for the std::numpunct<char> class is described by [Table 16-276](#ID_RTTI_45_24014)

**Table 16-276 typeinfo for numpunct<char>**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for numpunct<char> |

#### 16.1.97.2 Interfaces for Class numpunct<char>

An LSB conforming implementation shall provide the generic methods for Class std::numpunct<char> specified in [Table 16-277](#ID_TBL_45_LIBSTDCXX_45_CLAVQ_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-277 libstdcxx - Class numpunct<char> Function Interfaces**

|  |
| --- |
| numpunct<char>::do\_grouping() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct<char>::do\_truename() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct<char>::do\_falsename() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct<char>::decimal\_point() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct<char>::thousands\_sep() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct<char>::do\_decimal\_point() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct<char>::do\_thousands\_sep() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct<char>::grouping() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct<char>::truename() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct<char>::falsename() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct<char>::\_M\_initialize\_numpunct(\_\_locale\_struct\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct<char>::~numpunct()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct<char>::~numpunct()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct<char>::~numpunct()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::numpunct<char> specified in [Table 16-278](#ID_TBL_45_LIBSTDCXX_45_CLAVQ_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-278 libstdcxx - Class numpunct<char> Data Interfaces**

|  |
| --- |
| guard variable for numpunct<char>::id(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| numpunct<char>::id(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for numpunct<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for numpunct<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for numpunct<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.98 Class numpunct<wchar\_t>**

#### 16.1.98.1 Class data for numpunct<wchar\_t>

The virtual table for the std::numpunct<wchar\_t> class is described by [Table 16-279](#ID_CLS_45_24294_45_0)

**Table 16-279 Primary vtable for numpunct<wchar\_t>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for numpunct<wchar\_t> |
| vfunc[0]: | numpunct<wchar\_t>::~numpunct() |
| vfunc[1]: | numpunct<wchar\_t>::~numpunct() |
| vfunc[2]: | numpunct<wchar\_t>::do\_decimal\_point() const |
| vfunc[3]: | numpunct<wchar\_t>::do\_thousands\_sep() const |
| vfunc[4]: | numpunct<wchar\_t>::do\_grouping() const |
| vfunc[5]: | numpunct<wchar\_t>::do\_truename() const |
| vfunc[6]: | numpunct<wchar\_t>::do\_falsename() const |

The Run Time Type Information for the std::numpunct<wchar\_t> class is described by [Table 16-280](#ID_RTTI_45_24024)

**Table 16-280 typeinfo for numpunct<wchar\_t>**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for numpunct<wchar\_t> |

#### 16.1.98.2 Interfaces for Class numpunct<wchar\_t>

An LSB conforming implementation shall provide the generic methods for Class std::numpunct<wchar\_t> specified in [Table 16-281](#ID_TBL_45_LIBSTDCXX_45_CLAVR_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-281 libstdcxx - Class numpunct<wchar\_t> Function Interfaces**

|  |
| --- |
| numpunct<wchar\_t>::do\_grouping() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct<wchar\_t>::do\_truename() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct<wchar\_t>::do\_falsename() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct<wchar\_t>::decimal\_point() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct<wchar\_t>::thousands\_sep() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct<wchar\_t>::do\_decimal\_point() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct<wchar\_t>::do\_thousands\_sep() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct<wchar\_t>::grouping() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct<wchar\_t>::truename() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct<wchar\_t>::falsename() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct<wchar\_t>::\_M\_initialize\_numpunct(\_\_locale\_struct\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct<wchar\_t>::~numpunct()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct<wchar\_t>::~numpunct()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct<wchar\_t>::~numpunct()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::numpunct<wchar\_t> specified in [Table 16-282](#ID_TBL_45_LIBSTDCXX_45_CLAVR_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-282 libstdcxx - Class numpunct<wchar\_t> Data Interfaces**

|  |
| --- |
| guard variable for numpunct<wchar\_t>::id(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| numpunct<wchar\_t>::id(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for numpunct<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for numpunct<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for numpunct<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.99 Class numpunct\_byname<char>**

#### 16.1.99.1 Class data for numpunct\_byname<char>

The virtual table for the std::numpunct\_byname<char> class is described by [Table 16-283](#ID_CLS_45_24279_45_0)

**Table 16-283 Primary vtable for numpunct\_byname<char>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for numpunct\_byname<char> |
| vfunc[0]: | numpunct\_byname<char>::~numpunct\_byname() |
| vfunc[1]: | numpunct\_byname<char>::~numpunct\_byname() |
| vfunc[2]: | numpunct<char>::do\_decimal\_point() const |
| vfunc[3]: | numpunct<char>::do\_thousands\_sep() const |
| vfunc[4]: | numpunct<char>::do\_grouping() const |
| vfunc[5]: | numpunct<char>::do\_truename() const |
| vfunc[6]: | numpunct<char>::do\_falsename() const |

The Run Time Type Information for the std::numpunct\_byname<char> class is described by [Table 16-284](#ID_RTTI_45_24079)

**Table 16-284 typeinfo for numpunct\_byname<char>**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for numpunct\_byname<char> |

#### 16.1.99.2 Interfaces for Class numpunct\_byname<char>

An LSB conforming implementation shall provide the generic methods for Class std::numpunct\_byname<char> specified in [Table 16-285](#ID_TBL_45_LIBSTDCXX_45_CLAVS_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-285 libstdcxx - Class numpunct\_byname<char> Function Interfaces**

|  |
| --- |
| numpunct\_byname<char>::~numpunct\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct\_byname<char>::~numpunct\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct\_byname<char>::~numpunct\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::numpunct\_byname<char> specified in [Table 16-286](#ID_TBL_45_LIBSTDCXX_45_CLAVS_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-286 libstdcxx - Class numpunct\_byname<char> Data Interfaces**

|  |
| --- |
| typeinfo for numpunct\_byname<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for numpunct\_byname<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for numpunct\_byname<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.100 Class numpunct\_byname<wchar\_t>**

#### 16.1.100.1 Class data for numpunct\_byname<wchar\_t>

The virtual table for the std::numpunct\_byname<wchar\_t> class is described by [Table 16-287](#ID_CLS_45_24265_45_0)

**Table 16-287 Primary vtable for numpunct\_byname<wchar\_t>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for numpunct\_byname<wchar\_t> |
| vfunc[0]: | numpunct\_byname<wchar\_t>::~numpunct\_byname() |
| vfunc[1]: | numpunct\_byname<wchar\_t>::~numpunct\_byname() |
| vfunc[2]: | numpunct<wchar\_t>::do\_decimal\_point() const |
| vfunc[3]: | numpunct<wchar\_t>::do\_thousands\_sep() const |
| vfunc[4]: | numpunct<wchar\_t>::do\_grouping() const |
| vfunc[5]: | numpunct<wchar\_t>::do\_truename() const |
| vfunc[6]: | numpunct<wchar\_t>::do\_falsename() const |

The Run Time Type Information for the std::numpunct\_byname<wchar\_t> class is described by [Table 16-288](#ID_RTTI_45_24067)

**Table 16-288 typeinfo for numpunct\_byname<wchar\_t>**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for numpunct\_byname<wchar\_t> |

#### 16.1.100.2 Interfaces for Class numpunct\_byname<wchar\_t>

An LSB conforming implementation shall provide the generic methods for Class std::numpunct\_byname<wchar\_t> specified in [Table 16-289](#ID_TBL_45_LIBSTDCXX_45_CLAVT_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-289 libstdcxx - Class numpunct\_byname<wchar\_t> Function Interfaces**

|  |
| --- |
| numpunct\_byname<wchar\_t>::~numpunct\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct\_byname<wchar\_t>::~numpunct\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct\_byname<wchar\_t>::~numpunct\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::numpunct\_byname<wchar\_t> specified in [Table 16-290](#ID_TBL_45_LIBSTDCXX_45_CLAVT_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-290 libstdcxx - Class numpunct\_byname<wchar\_t> Data Interfaces**

|  |
| --- |
| typeinfo for numpunct\_byname<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for numpunct\_byname<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for numpunct\_byname<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.101 Class \_\_codecvt\_abstract\_base<char, char, \_\_mbstate\_t>**

#### 16.1.101.1 Class data for \_\_codecvt\_abstract\_base<char, char, \_\_mbstate\_t>

The virtual table for the std::\_\_codecvt\_abstract\_base<char, char, \_\_mbstate\_t> class is described by [Table 16-291](#ID_CLS_45_24297_45_0)

**Table 16-291 Primary vtable for \_\_codecvt\_abstract\_base<char, char, \_\_mbstate\_t>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for \_\_codecvt\_abstract\_base<char, char, \_\_mbstate\_t> |
| vfunc[0]: | NULL or \_\_codecvt\_abstract\_base<char, char, \_\_mbstate\_t>::~\_\_codecvt\_abstract\_base() |
| vfunc[1]: | NULL or \_\_codecvt\_abstract\_base<char, char, \_\_mbstate\_t>::~\_\_codecvt\_abstract\_base() |
| vfunc[2]: | \_\_cxa\_pure\_virtual |
| vfunc[3]: | \_\_cxa\_pure\_virtual |
| vfunc[4]: | \_\_cxa\_pure\_virtual |
| vfunc[5]: | \_\_cxa\_pure\_virtual |
| vfunc[6]: | \_\_cxa\_pure\_virtual |
| vfunc[7]: | \_\_cxa\_pure\_virtual |
| vfunc[8]: | \_\_cxa\_pure\_virtual |

#### 16.1.101.2 Interfaces for Class \_\_codecvt\_abstract\_base<char, char, \_\_mbstate\_t>

No external methods are defined for libstdcxx - Class std::\_\_codecvt\_abstract\_base<char, char, \_\_mbstate\_t> in this part of the specification. See also the relevant architecture specific part of this specification.

An LSB conforming implementation shall provide the generic data interfaces for Class std::\_\_codecvt\_abstract\_base<char, char, \_\_mbstate\_t> specified in [Table 16-292](#ID_TBL_45_LIBSTDCXX_45_CLAVU_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-292 libstdcxx - Class \_\_codecvt\_abstract\_base<char, char, \_\_mbstate\_t> Data Interfaces**

|  |
| --- |
| typeinfo for \_\_codecvt\_abstract\_base<char, char, \_\_mbstate\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for \_\_codecvt\_abstract\_base<char, char, \_\_mbstate\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for \_\_codecvt\_abstract\_base<char, char, \_\_mbstate\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.102 Class \_\_codecvt\_abstract\_base<wchar\_t, char, \_\_mbstate\_t>**

#### 16.1.102.1 Class data for \_\_codecvt\_abstract\_base<wchar\_t, char, \_\_mbstate\_t>

The virtual table for the std::\_\_codecvt\_abstract\_base<wchar\_t, char, \_\_mbstate\_t> class is described by [Table 16-293](#ID_CLS_45_24296_45_0)

**Table 16-293 Primary vtable for \_\_codecvt\_abstract\_base<wchar\_t, char, \_\_mbstate\_t>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for \_\_codecvt\_abstract\_base<wchar\_t, char, \_\_mbstate\_t> |
| vfunc[0]: | NULL or \_\_codecvt\_abstract\_base<wchar\_t, char, \_\_mbstate\_t>::~\_\_codecvt\_abstract\_base() |
| vfunc[1]: | NULL or \_\_codecvt\_abstract\_base<wchar\_t, char, \_\_mbstate\_t>::~\_\_codecvt\_abstract\_base() |
| vfunc[2]: | \_\_cxa\_pure\_virtual |
| vfunc[3]: | \_\_cxa\_pure\_virtual |
| vfunc[4]: | \_\_cxa\_pure\_virtual |
| vfunc[5]: | \_\_cxa\_pure\_virtual |
| vfunc[6]: | \_\_cxa\_pure\_virtual |
| vfunc[7]: | \_\_cxa\_pure\_virtual |
| vfunc[8]: | \_\_cxa\_pure\_virtual |

#### 16.1.102.2 Interfaces for Class \_\_codecvt\_abstract\_base<wchar\_t, char, \_\_mbstate\_t>

No external methods are defined for libstdcxx - Class std::\_\_codecvt\_abstract\_base<wchar\_t, char, \_\_mbstate\_t> in this part of the specification. See also the relevant architecture specific part of this specification.

An LSB conforming implementation shall provide the generic data interfaces for Class std::\_\_codecvt\_abstract\_base<wchar\_t, char, \_\_mbstate\_t> specified in [Table 16-294](#ID_TBL_45_LIBSTDCXX_45_CLAVV_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-294 libstdcxx - Class \_\_codecvt\_abstract\_base<wchar\_t, char, \_\_mbstate\_t> Data Interfaces**

|  |
| --- |
| typeinfo for \_\_codecvt\_abstract\_base<wchar\_t, char, \_\_mbstate\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for \_\_codecvt\_abstract\_base<wchar\_t, char, \_\_mbstate\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for \_\_codecvt\_abstract\_base<wchar\_t, char, \_\_mbstate\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.103 Class codecvt\_base**

#### 16.1.103.1 Class data for codecvt\_base

The Run Time Type Information for the std::codecvt\_base class is described by [Table 16-295](#ID_RTTI_45_24009)

**Table 16-295 typeinfo for codecvt\_base**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_class\_type\_info |
| Name | typeinfo name for codecvt\_base |

#### 16.1.103.2 Interfaces for Class codecvt\_base

No external methods are defined for libstdcxx - Class std::codecvt\_base in this part of the specification. See also the relevant architecture specific part of this specification.

An LSB conforming implementation shall provide the generic data interfaces for Class std::codecvt\_base specified in [Table 16-296](#ID_TBL_45_LIBSTDCXX_45_CLAVW_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-296 libstdcxx - Class codecvt\_base Data Interfaces**

|  |
| --- |
| typeinfo for codecvt\_base(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for codecvt\_base(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.104 Class codecvt<char, char, \_\_mbstate\_t>**

#### 16.1.104.1 Class data for codecvt<char, char, \_\_mbstate\_t>

The virtual table for the std::codecvt<char, char, \_\_mbstate\_t> class is described by [Table 16-297](#ID_CLS_45_24228_45_0)

**Table 16-297 Primary vtable for codecvt<char, char, \_\_mbstate\_t>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for codecvt<char, char, \_\_mbstate\_t> |
| vfunc[0]: | codecvt<char, char, \_\_mbstate\_t>::~codecvt() |
| vfunc[1]: | codecvt<char, char, \_\_mbstate\_t>::~codecvt() |
| vfunc[2]: | codecvt<char, char, \_\_mbstate\_t>::do\_out(\_\_mbstate\_t&, char const\*, char const\*, char const\*&, char\*, char\*, char\*&) const |
| vfunc[3]: | codecvt<char, char, \_\_mbstate\_t>::do\_unshift(\_\_mbstate\_t&, char\*, char\*, char\*&) const |
| vfunc[4]: | codecvt<char, char, \_\_mbstate\_t>::do\_in(\_\_mbstate\_t&, char const\*, char const\*, char const\*&, char\*, char\*, char\*&) const |
| vfunc[5]: | codecvt<char, char, \_\_mbstate\_t>::do\_encoding() const |
| vfunc[6]: | codecvt<char, char, \_\_mbstate\_t>::do\_always\_noconv() const |
| vfunc[7]: | See architecture specific part. |
| vfunc[8]: | codecvt<char, char, \_\_mbstate\_t>::do\_max\_length() const |

The Run Time Type Information for the std::codecvt<char, char, \_\_mbstate\_t> class is described by [Table 16-298](#ID_RTTI_45_24020)

**Table 16-298 typeinfo for codecvt<char, char, \_\_mbstate\_t>**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for codecvt<char, char, \_\_mbstate\_t> |

#### 16.1.104.2 Interfaces for Class codecvt<char, char, \_\_mbstate\_t>

An LSB conforming implementation shall provide the generic methods for Class std::codecvt<char, char, \_\_mbstate\_t> specified in [Table 16-299](#ID_TBL_45_LIBSTDCXX_45_CLAVX_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-299 libstdcxx - Class codecvt<char, char, \_\_mbstate\_t> Function Interfaces**

|  |
| --- |
| codecvt<char, char, \_\_mbstate\_t>::do\_unshift(\_\_mbstate\_t&, char\*, char\*, char\*&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| codecvt<char, char, \_\_mbstate\_t>::do\_encoding() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| codecvt<char, char, \_\_mbstate\_t>::do\_max\_length() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| codecvt<char, char, \_\_mbstate\_t>::do\_always\_noconv() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| codecvt<char, char, \_\_mbstate\_t>::do\_in(\_\_mbstate\_t&, char const\*, char const\*, char const\*&, char\*, char\*, char\*&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| codecvt<char, char, \_\_mbstate\_t>::do\_out(\_\_mbstate\_t&, char const\*, char const\*, char const\*&, char\*, char\*, char\*&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| codecvt<char, char, \_\_mbstate\_t>::~codecvt()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| codecvt<char, char, \_\_mbstate\_t>::~codecvt()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| codecvt<char, char, \_\_mbstate\_t>::~codecvt()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::codecvt<char, char, \_\_mbstate\_t> specified in [Table 16-300](#ID_TBL_45_LIBSTDCXX_45_CLAVX_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-300 libstdcxx - Class codecvt<char, char, \_\_mbstate\_t> Data Interfaces**

|  |
| --- |
| codecvt<char, char, \_\_mbstate\_t>::id(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for codecvt<char, char, \_\_mbstate\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for codecvt<char, char, \_\_mbstate\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for codecvt<char, char, \_\_mbstate\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.105 Class codecvt<wchar\_t, char, \_\_mbstate\_t>**

#### 16.1.105.1 Class data for codecvt<wchar\_t, char, \_\_mbstate\_t>

The virtual table for the std::codecvt<wchar\_t, char, \_\_mbstate\_t> class is described by [Table 16-301](#ID_CLS_45_24227_45_0)

**Table 16-301 Primary vtable for codecvt<wchar\_t, char, \_\_mbstate\_t>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for codecvt<wchar\_t, char, \_\_mbstate\_t> |
| vfunc[0]: | codecvt<wchar\_t, char, \_\_mbstate\_t>::~codecvt() |
| vfunc[1]: | codecvt<wchar\_t, char, \_\_mbstate\_t>::~codecvt() |
| vfunc[2]: | codecvt<wchar\_t, char, \_\_mbstate\_t>::do\_out(\_\_mbstate\_t&, wchar\_t const\*, wchar\_t const\*, wchar\_t const\*&, char\*, char\*, char\*&) const |
| vfunc[3]: | codecvt<wchar\_t, char, \_\_mbstate\_t>::do\_unshift(\_\_mbstate\_t&, char\*, char\*, char\*&) const |
| vfunc[4]: | codecvt<wchar\_t, char, \_\_mbstate\_t>::do\_in(\_\_mbstate\_t&, char const\*, char const\*, char const\*&, wchar\_t\*, wchar\_t\*, wchar\_t\*&) const |
| vfunc[5]: | codecvt<wchar\_t, char, \_\_mbstate\_t>::do\_encoding() const |
| vfunc[6]: | codecvt<wchar\_t, char, \_\_mbstate\_t>::do\_always\_noconv() const |
| vfunc[7]: | See architecture specific part. |
| vfunc[8]: | codecvt<wchar\_t, char, \_\_mbstate\_t>::do\_max\_length() const |

The Run Time Type Information for the std::codecvt<wchar\_t, char, \_\_mbstate\_t> class is described by [Table 16-302](#ID_RTTI_45_24028)

**Table 16-302 typeinfo for codecvt<wchar\_t, char, \_\_mbstate\_t>**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for codecvt<wchar\_t, char, \_\_mbstate\_t> |

#### 16.1.105.2 Interfaces for Class codecvt<wchar\_t, char, \_\_mbstate\_t>

An LSB conforming implementation shall provide the generic methods for Class std::codecvt<wchar\_t, char, \_\_mbstate\_t> specified in [Table 16-303](#ID_TBL_45_LIBSTDCXX_45_CLAVY_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-303 libstdcxx - Class codecvt<wchar\_t, char, \_\_mbstate\_t> Function Interfaces**

|  |
| --- |
| codecvt<wchar\_t, char, \_\_mbstate\_t>::do\_unshift(\_\_mbstate\_t&, char\*, char\*, char\*&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| codecvt<wchar\_t, char, \_\_mbstate\_t>::do\_encoding() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| codecvt<wchar\_t, char, \_\_mbstate\_t>::do\_max\_length() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| codecvt<wchar\_t, char, \_\_mbstate\_t>::do\_always\_noconv() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| codecvt<wchar\_t, char, \_\_mbstate\_t>::do\_in(\_\_mbstate\_t&, char const\*, char const\*, char const\*&, wchar\_t\*, wchar\_t\*, wchar\_t\*&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| codecvt<wchar\_t, char, \_\_mbstate\_t>::do\_out(\_\_mbstate\_t&, wchar\_t const\*, wchar\_t const\*, wchar\_t const\*&, char\*, char\*, char\*&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| codecvt<wchar\_t, char, \_\_mbstate\_t>::~codecvt()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| codecvt<wchar\_t, char, \_\_mbstate\_t>::~codecvt()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| codecvt<wchar\_t, char, \_\_mbstate\_t>::~codecvt()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::codecvt<wchar\_t, char, \_\_mbstate\_t> specified in [Table 16-304](#ID_TBL_45_LIBSTDCXX_45_CLAVY_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-304 libstdcxx - Class codecvt<wchar\_t, char, \_\_mbstate\_t> Data Interfaces**

|  |
| --- |
| codecvt<wchar\_t, char, \_\_mbstate\_t>::id(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for codecvt<wchar\_t, char, \_\_mbstate\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for codecvt<wchar\_t, char, \_\_mbstate\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for codecvt<wchar\_t, char, \_\_mbstate\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.106 Class codecvt\_byname<char, char, \_\_mbstate\_t>**

#### 16.1.106.1 Class data for codecvt\_byname<char, char, \_\_mbstate\_t>

The virtual table for the std::codecvt\_byname<char, char, \_\_mbstate\_t> class is described by [Table 16-305](#ID_CLS_45_24271_45_0)

**Table 16-305 Primary vtable for codecvt\_byname<char, char, \_\_mbstate\_t>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for codecvt\_byname<char, char, \_\_mbstate\_t> |
| vfunc[0]: | codecvt\_byname<char, char, \_\_mbstate\_t>::~codecvt\_byname() |
| vfunc[1]: | codecvt\_byname<char, char, \_\_mbstate\_t>::~codecvt\_byname() |
| vfunc[2]: | codecvt<char, char, \_\_mbstate\_t>::do\_out(\_\_mbstate\_t&, char const\*, char const\*, char const\*&, char\*, char\*, char\*&) const |
| vfunc[3]: | codecvt<char, char, \_\_mbstate\_t>::do\_unshift(\_\_mbstate\_t&, char\*, char\*, char\*&) const |
| vfunc[4]: | codecvt<char, char, \_\_mbstate\_t>::do\_in(\_\_mbstate\_t&, char const\*, char const\*, char const\*&, char\*, char\*, char\*&) const |
| vfunc[5]: | codecvt<char, char, \_\_mbstate\_t>::do\_encoding() const |
| vfunc[6]: | codecvt<char, char, \_\_mbstate\_t>::do\_always\_noconv() const |
| vfunc[7]: | See architecture specific part. |
| vfunc[8]: | codecvt<char, char, \_\_mbstate\_t>::do\_max\_length() const |

The Run Time Type Information for the std::codecvt\_byname<char, char, \_\_mbstate\_t> class is described by [Table 16-306](#ID_RTTI_45_24073)

**Table 16-306 typeinfo for codecvt\_byname<char, char, \_\_mbstate\_t>**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for codecvt\_byname<char, char, \_\_mbstate\_t> |

#### 16.1.106.2 Interfaces for Class codecvt\_byname<char, char, \_\_mbstate\_t>

An LSB conforming implementation shall provide the generic methods for Class std::codecvt\_byname<char, char, \_\_mbstate\_t> specified in [Table 16-307](#ID_TBL_45_LIBSTDCXX_45_CLAVZ_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-307 libstdcxx - Class codecvt\_byname<char, char, \_\_mbstate\_t> Function Interfaces**

|  |
| --- |
| codecvt\_byname<char, char, \_\_mbstate\_t>::~codecvt\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| codecvt\_byname<char, char, \_\_mbstate\_t>::~codecvt\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| codecvt\_byname<char, char, \_\_mbstate\_t>::~codecvt\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::codecvt\_byname<char, char, \_\_mbstate\_t> specified in [Table 16-308](#ID_TBL_45_LIBSTDCXX_45_CLAVZ_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-308 libstdcxx - Class codecvt\_byname<char, char, \_\_mbstate\_t> Data Interfaces**

|  |
| --- |
| typeinfo for codecvt\_byname<char, char, \_\_mbstate\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for codecvt\_byname<char, char, \_\_mbstate\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for codecvt\_byname<char, char, \_\_mbstate\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.107 Class codecvt\_byname<wchar\_t, char, \_\_mbstate\_t>**

#### 16.1.107.1 Class data for codecvt\_byname<wchar\_t, char, \_\_mbstate\_t>

The virtual table for the std::codecvt\_byname<wchar\_t, char, \_\_mbstate\_t> class is described by [Table 16-309](#ID_CLS_45_24257_45_0)

**Table 16-309 Primary vtable for codecvt\_byname<wchar\_t, char, \_\_mbstate\_t>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for codecvt\_byname<wchar\_t, char, \_\_mbstate\_t> |
| vfunc[0]: | codecvt\_byname<wchar\_t, char, \_\_mbstate\_t>::~codecvt\_byname() |
| vfunc[1]: | codecvt\_byname<wchar\_t, char, \_\_mbstate\_t>::~codecvt\_byname() |
| vfunc[2]: | codecvt<wchar\_t, char, \_\_mbstate\_t>::do\_out(\_\_mbstate\_t&, wchar\_t const\*, wchar\_t const\*, wchar\_t const\*&, char\*, char\*, char\*&) const |
| vfunc[3]: | codecvt<wchar\_t, char, \_\_mbstate\_t>::do\_unshift(\_\_mbstate\_t&, char\*, char\*, char\*&) const |
| vfunc[4]: | codecvt<wchar\_t, char, \_\_mbstate\_t>::do\_in(\_\_mbstate\_t&, char const\*, char const\*, char const\*&, wchar\_t\*, wchar\_t\*, wchar\_t\*&) const |
| vfunc[5]: | codecvt<wchar\_t, char, \_\_mbstate\_t>::do\_encoding() const |
| vfunc[6]: | codecvt<wchar\_t, char, \_\_mbstate\_t>::do\_always\_noconv() const |
| vfunc[7]: | See architecture specific part. |
| vfunc[8]: | codecvt<wchar\_t, char, \_\_mbstate\_t>::do\_max\_length() const |

The Run Time Type Information for the std::codecvt\_byname<wchar\_t, char, \_\_mbstate\_t> class is described by [Table 16-310](#ID_RTTI_45_24061)

**Table 16-310 typeinfo for codecvt\_byname<wchar\_t, char, \_\_mbstate\_t>**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for codecvt\_byname<wchar\_t, char, \_\_mbstate\_t> |

#### 16.1.107.2 Interfaces for Class codecvt\_byname<wchar\_t, char, \_\_mbstate\_t>

An LSB conforming implementation shall provide the generic methods for Class std::codecvt\_byname<wchar\_t, char, \_\_mbstate\_t> specified in [Table 16-311](#ID_TBL_45_LIBSTDCXX_45_CLAWA_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-311 libstdcxx - Class codecvt\_byname<wchar\_t, char, \_\_mbstate\_t> Function Interfaces**

|  |
| --- |
| codecvt\_byname<wchar\_t, char, \_\_mbstate\_t>::~codecvt\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| codecvt\_byname<wchar\_t, char, \_\_mbstate\_t>::~codecvt\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| codecvt\_byname<wchar\_t, char, \_\_mbstate\_t>::~codecvt\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::codecvt\_byname<wchar\_t, char, \_\_mbstate\_t> specified in [Table 16-312](#ID_TBL_45_LIBSTDCXX_45_CLAWA_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-312 libstdcxx - Class codecvt\_byname<wchar\_t, char, \_\_mbstate\_t> Data Interfaces**

|  |
| --- |
| typeinfo for codecvt\_byname<wchar\_t, char, \_\_mbstate\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for codecvt\_byname<wchar\_t, char, \_\_mbstate\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for codecvt\_byname<wchar\_t, char, \_\_mbstate\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.108 Class collate<char>**

#### 16.1.108.1 Class data for collate<char>

The virtual table for the std::collate<char> class is described by [Table 16-313](#ID_CLS_45_24293_45_0)

**Table 16-313 Primary vtable for collate<char>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for collate<char> |
| vfunc[0]: | collate<char>::~collate() |
| vfunc[1]: | collate<char>::~collate() |
| vfunc[2]: | collate<char>::do\_compare(char const\*, char const\*, char const\*, char const\*) const |
| vfunc[3]: | collate<char>::do\_transform(char const\*, char const\*) const |
| vfunc[4]: | collate<char>::do\_hash(char const\*, char const\*) const |

The Run Time Type Information for the std::collate<char> class is described by [Table 16-314](#ID_RTTI_45_24021)

**Table 16-314 typeinfo for collate<char>**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for collate<char> |

#### 16.1.108.2 Interfaces for Class collate<char>

An LSB conforming implementation shall provide the generic methods for Class std::collate<char> specified in [Table 16-315](#ID_TBL_45_LIBSTDCXX_45_CLAWB_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-315 libstdcxx - Class collate<char> Function Interfaces**

|  |
| --- |
| collate<char>::\_M\_compare(char const\*, char const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| collate<char>::do\_compare(char const\*, char const\*, char const\*, char const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| collate<char>::do\_transform(char const\*, char const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| collate<char>::hash(char const\*, char const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| collate<char>::compare(char const\*, char const\*, char const\*, char const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| collate<char>::do\_hash(char const\*, char const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| collate<char>::transform(char const\*, char const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| collate<char>::~collate()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| collate<char>::~collate()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| collate<char>::~collate()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::collate<char> specified in [Table 16-316](#ID_TBL_45_LIBSTDCXX_45_CLAWB_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-316 libstdcxx - Class collate<char> Data Interfaces**

|  |
| --- |
| guard variable for collate<char>::id(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| collate<char>::id(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for collate<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for collate<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for collate<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.109 Class collate<wchar\_t>**

#### 16.1.109.1 Class data for collate<wchar\_t>

The virtual table for the std::collate<wchar\_t> class is described by [Table 16-317](#ID_CLS_45_24292_45_0)

**Table 16-317 Primary vtable for collate<wchar\_t>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for collate<wchar\_t> |
| vfunc[0]: | collate<wchar\_t>::~collate() |
| vfunc[1]: | collate<wchar\_t>::~collate() |
| vfunc[2]: | collate<wchar\_t>::do\_compare(wchar\_t const\*, wchar\_t const\*, wchar\_t const\*, wchar\_t const\*) const |
| vfunc[3]: | collate<wchar\_t>::do\_transform(wchar\_t const\*, wchar\_t const\*) const |
| vfunc[4]: | collate<wchar\_t>::do\_hash(wchar\_t const\*, wchar\_t const\*) const |

The Run Time Type Information for the std::collate<wchar\_t> class is described by [Table 16-318](#ID_RTTI_45_24029)

**Table 16-318 typeinfo for collate<wchar\_t>**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for collate<wchar\_t> |

#### 16.1.109.2 Interfaces for Class collate<wchar\_t>

An LSB conforming implementation shall provide the generic methods for Class std::collate<wchar\_t> specified in [Table 16-319](#ID_TBL_45_LIBSTDCXX_45_CLAWC_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-319 libstdcxx - Class collate<wchar\_t> Function Interfaces**

|  |
| --- |
| collate<wchar\_t>::\_M\_compare(wchar\_t const\*, wchar\_t const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| collate<wchar\_t>::do\_compare(wchar\_t const\*, wchar\_t const\*, wchar\_t const\*, wchar\_t const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| collate<wchar\_t>::do\_transform(wchar\_t const\*, wchar\_t const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| collate<wchar\_t>::hash(wchar\_t const\*, wchar\_t const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| collate<wchar\_t>::compare(wchar\_t const\*, wchar\_t const\*, wchar\_t const\*, wchar\_t const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| collate<wchar\_t>::do\_hash(wchar\_t const\*, wchar\_t const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| collate<wchar\_t>::transform(wchar\_t const\*, wchar\_t const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| collate<wchar\_t>::~collate()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| collate<wchar\_t>::~collate()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| collate<wchar\_t>::~collate()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::collate<wchar\_t> specified in [Table 16-320](#ID_TBL_45_LIBSTDCXX_45_CLAWC_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-320 libstdcxx - Class collate<wchar\_t> Data Interfaces**

|  |
| --- |
| guard variable for collate<wchar\_t>::id(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| collate<wchar\_t>::id(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for collate<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for collate<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for collate<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.110 Class collate\_byname<char>**

#### 16.1.110.1 Class data for collate\_byname<char>

The virtual table for the std::collate\_byname<char> class is described by [Table 16-321](#ID_CLS_45_24270_45_0)

**Table 16-321 Primary vtable for collate\_byname<char>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for collate\_byname<char> |
| vfunc[0]: | collate\_byname<char>::~collate\_byname() |
| vfunc[1]: | collate\_byname<char>::~collate\_byname() |
| vfunc[2]: | collate<char>::do\_compare(char const\*, char const\*, char const\*, char const\*) const |
| vfunc[3]: | collate<char>::do\_transform(char const\*, char const\*) const |
| vfunc[4]: | collate<char>::do\_hash(char const\*, char const\*) const |

The Run Time Type Information for the std::collate\_byname<char> class is described by [Table 16-322](#ID_RTTI_45_24072)

**Table 16-322 typeinfo for collate\_byname<char>**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for collate\_byname<char> |

#### 16.1.110.2 Interfaces for Class collate\_byname<char>

An LSB conforming implementation shall provide the generic methods for Class std::collate\_byname<char> specified in [Table 16-323](#ID_TBL_45_LIBSTDCXX_45_CLAWD_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-323 libstdcxx - Class collate\_byname<char> Function Interfaces**

|  |
| --- |
| collate\_byname<char>::~collate\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| collate\_byname<char>::~collate\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| collate\_byname<char>::~collate\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::collate\_byname<char> specified in [Table 16-324](#ID_TBL_45_LIBSTDCXX_45_CLAWD_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-324 libstdcxx - Class collate\_byname<char> Data Interfaces**

|  |
| --- |
| typeinfo for collate\_byname<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for collate\_byname<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for collate\_byname<char>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.111 Class collate\_byname<wchar\_t>**

#### 16.1.111.1 Class data for collate\_byname<wchar\_t>

The virtual table for the std::collate\_byname<wchar\_t> class is described by [Table 16-325](#ID_CLS_45_24256_45_0)

**Table 16-325 Primary vtable for collate\_byname<wchar\_t>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for collate\_byname<wchar\_t> |
| vfunc[0]: | collate\_byname<wchar\_t>::~collate\_byname() |
| vfunc[1]: | collate\_byname<wchar\_t>::~collate\_byname() |
| vfunc[2]: | collate<wchar\_t>::do\_compare(wchar\_t const\*, wchar\_t const\*, wchar\_t const\*, wchar\_t const\*) const |
| vfunc[3]: | collate<wchar\_t>::do\_transform(wchar\_t const\*, wchar\_t const\*) const |
| vfunc[4]: | collate<wchar\_t>::do\_hash(wchar\_t const\*, wchar\_t const\*) const |

The Run Time Type Information for the std::collate\_byname<wchar\_t> class is described by [Table 16-326](#ID_RTTI_45_24060)

**Table 16-326 typeinfo for collate\_byname<wchar\_t>**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for collate\_byname<wchar\_t> |

#### 16.1.111.2 Interfaces for Class collate\_byname<wchar\_t>

An LSB conforming implementation shall provide the generic methods for Class std::collate\_byname<wchar\_t> specified in [Table 16-327](#ID_TBL_45_LIBSTDCXX_45_CLAWE_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-327 libstdcxx - Class collate\_byname<wchar\_t> Function Interfaces**

|  |
| --- |
| collate\_byname<wchar\_t>::~collate\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| collate\_byname<wchar\_t>::~collate\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| collate\_byname<wchar\_t>::~collate\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::collate\_byname<wchar\_t> specified in [Table 16-328](#ID_TBL_45_LIBSTDCXX_45_CLAWE_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-328 libstdcxx - Class collate\_byname<wchar\_t> Data Interfaces**

|  |
| --- |
| typeinfo for collate\_byname<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for collate\_byname<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for collate\_byname<wchar\_t>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.112 Class time\_base**

#### 16.1.112.1 Class data for time\_base

The Run Time Type Information for the std::time\_base class is described by [Table 16-329](#ID_RTTI_45_24016)

**Table 16-329 typeinfo for time\_base**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_class\_type\_info |
| Name | typeinfo name for time\_base |

#### 16.1.112.2 Interfaces for Class time\_base

No external methods are defined for libstdcxx - Class std::time\_base in this part of the specification. See also the relevant architecture specific part of this specification.

An LSB conforming implementation shall provide the generic data interfaces for Class std::time\_base specified in [Table 16-330](#ID_TBL_45_LIBSTDCXX_45_CLAWF_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-330 libstdcxx - Class time\_base Data Interfaces**

|  |
| --- |
| typeinfo for time\_base(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for time\_base(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.113 Class time\_get\_byname<char, istreambuf\_iterator<char, char\_traits<char> > >**

#### 16.1.113.1 Class data for time\_get\_byname<char, istreambuf\_iterator<char, char\_traits<char> > >

The virtual table for the std::time\_get\_byname<char, std::istreambuf\_iterator<char, std::char\_traits<char> > > class is described by [Table 16-331](#ID_CLS_45_24273_45_0)

**Table 16-331 Primary vtable for time\_get\_byname<char, istreambuf\_iterator<char, char\_traits<char> > >**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for time\_get\_byname<char, istreambuf\_iterator<char, char\_traits<char> > > |
| vfunc[0]: | time\_get\_byname<char, istreambuf\_iterator<char, char\_traits<char> > >::~time\_get\_byname() |
| vfunc[1]: | time\_get\_byname<char, istreambuf\_iterator<char, char\_traits<char> > >::~time\_get\_byname() |
| vfunc[2]: | time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_date\_order() const |
| vfunc[3]: | time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get\_time(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, tm\*) const |
| vfunc[4]: | time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get\_date(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, tm\*) const |
| vfunc[5]: | time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get\_weekday(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, tm\*) const |
| vfunc[6]: | time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get\_monthname(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, tm\*) const |
| vfunc[7]: | time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get\_year(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, tm\*) const |

The Run Time Type Information for the std::time\_get\_byname<char, std::istreambuf\_iterator<char, std::char\_traits<char> > > class is described by [Table 16-332](#ID_RTTI_45_24075)

**Table 16-332 typeinfo for time\_get\_byname<char, istreambuf\_iterator<char, char\_traits<char> > >**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for time\_get\_byname<char, istreambuf\_iterator<char, char\_traits<char> > > |

#### 16.1.113.2 Interfaces for Class time\_get\_byname<char, istreambuf\_iterator<char, char\_traits<char> > >

An LSB conforming implementation shall provide the generic methods for Class std::time\_get\_byname<char, std::istreambuf\_iterator<char, std::char\_traits<char> > > specified in [Table 16-333](#ID_TBL_45_LIBSTDCXX_45_CLAWG_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-333 libstdcxx - Class time\_get\_byname<char, istreambuf\_iterator<char, char\_traits<char> > > Function Interfaces**

|  |
| --- |
| time\_get\_byname<char, istreambuf\_iterator<char, char\_traits<char> > >::~time\_get\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get\_byname<char, istreambuf\_iterator<char, char\_traits<char> > >::~time\_get\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get\_byname<char, istreambuf\_iterator<char, char\_traits<char> > >::~time\_get\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::time\_get\_byname<char, std::istreambuf\_iterator<char, std::char\_traits<char> > > specified in [Table 16-334](#ID_TBL_45_LIBSTDCXX_45_CLAWG_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-334 libstdcxx - Class time\_get\_byname<char, istreambuf\_iterator<char, char\_traits<char> > > Data Interfaces**

|  |
| --- |
| typeinfo for time\_get\_byname<char, istreambuf\_iterator<char, char\_traits<char> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for time\_get\_byname<char, istreambuf\_iterator<char, char\_traits<char> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for time\_get\_byname<char, istreambuf\_iterator<char, char\_traits<char> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.114 Class time\_get\_byname<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >**

#### 16.1.114.1 Class data for time\_get\_byname<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >

The virtual table for the std::time\_get\_byname<wchar\_t, std::istreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > class is described by [Table 16-335](#ID_CLS_45_24259_45_0)

**Table 16-335 Primary vtable for time\_get\_byname<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for time\_get\_byname<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > |
| vfunc[0]: | time\_get\_byname<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~time\_get\_byname() |
| vfunc[1]: | time\_get\_byname<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~time\_get\_byname() |
| vfunc[2]: | time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_date\_order() const |
| vfunc[3]: | time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get\_time(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, tm\*) const |
| vfunc[4]: | time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get\_date(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, tm\*) const |
| vfunc[5]: | time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get\_weekday(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, tm\*) const |
| vfunc[6]: | time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get\_monthname(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, tm\*) const |
| vfunc[7]: | time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get\_year(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, tm\*) const |

The Run Time Type Information for the std::time\_get\_byname<wchar\_t, std::istreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > class is described by [Table 16-336](#ID_RTTI_45_24063)

**Table 16-336 typeinfo for time\_get\_byname<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for time\_get\_byname<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > |

#### 16.1.114.2 Interfaces for Class time\_get\_byname<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >

An LSB conforming implementation shall provide the generic methods for Class std::time\_get\_byname<wchar\_t, std::istreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > specified in [Table 16-337](#ID_TBL_45_LIBSTDCXX_45_CLAWH_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-337 libstdcxx - Class time\_get\_byname<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > Function Interfaces**

|  |
| --- |
| time\_get\_byname<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~time\_get\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get\_byname<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~time\_get\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get\_byname<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~time\_get\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::time\_get\_byname<wchar\_t, std::istreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > specified in [Table 16-338](#ID_TBL_45_LIBSTDCXX_45_CLAWH_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-338 libstdcxx - Class time\_get\_byname<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > Data Interfaces**

|  |
| --- |
| typeinfo for time\_get\_byname<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for time\_get\_byname<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for time\_get\_byname<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.115 Class time\_put\_byname<char, ostreambuf\_iterator<char, char\_traits<char> > >**

#### 16.1.115.1 Class data for time\_put\_byname<char, ostreambuf\_iterator<char, char\_traits<char> > >

The virtual table for the std::time\_put\_byname<char, std::ostreambuf\_iterator<char, std::char\_traits<char> > > class is described by [Table 16-339](#ID_CLS_45_24275_45_0)

**Table 16-339 Primary vtable for time\_put\_byname<char, ostreambuf\_iterator<char, char\_traits<char> > >**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for time\_put\_byname<char, ostreambuf\_iterator<char, char\_traits<char> > > |
| vfunc[0]: | time\_put\_byname<char, ostreambuf\_iterator<char, char\_traits<char> > >::~time\_put\_byname() |
| vfunc[1]: | time\_put\_byname<char, ostreambuf\_iterator<char, char\_traits<char> > >::~time\_put\_byname() |
| vfunc[2]: | time\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::do\_put(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, tm const\*, char, char) const |

The Run Time Type Information for the std::time\_put\_byname<char, std::ostreambuf\_iterator<char, std::char\_traits<char> > > class is described by [Table 16-340](#ID_RTTI_45_24076)

**Table 16-340 typeinfo for time\_put\_byname<char, ostreambuf\_iterator<char, char\_traits<char> > >**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for time\_put\_byname<char, ostreambuf\_iterator<char, char\_traits<char> > > |

#### 16.1.115.2 Interfaces for Class time\_put\_byname<char, ostreambuf\_iterator<char, char\_traits<char> > >

An LSB conforming implementation shall provide the generic methods for Class std::time\_put\_byname<char, std::ostreambuf\_iterator<char, std::char\_traits<char> > > specified in [Table 16-341](#ID_TBL_45_LIBSTDCXX_45_CLAWI_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-341 libstdcxx - Class time\_put\_byname<char, ostreambuf\_iterator<char, char\_traits<char> > > Function Interfaces**

|  |
| --- |
| time\_put\_byname<char, ostreambuf\_iterator<char, char\_traits<char> > >::~time\_put\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_put\_byname<char, ostreambuf\_iterator<char, char\_traits<char> > >::~time\_put\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_put\_byname<char, ostreambuf\_iterator<char, char\_traits<char> > >::~time\_put\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::time\_put\_byname<char, std::ostreambuf\_iterator<char, std::char\_traits<char> > > specified in [Table 16-342](#ID_TBL_45_LIBSTDCXX_45_CLAWI_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-342 libstdcxx - Class time\_put\_byname<char, ostreambuf\_iterator<char, char\_traits<char> > > Data Interfaces**

|  |
| --- |
| typeinfo for time\_put\_byname<char, ostreambuf\_iterator<char, char\_traits<char> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for time\_put\_byname<char, ostreambuf\_iterator<char, char\_traits<char> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for time\_put\_byname<char, ostreambuf\_iterator<char, char\_traits<char> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.116 Class time\_put\_byname<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >**

#### 16.1.116.1 Class data for time\_put\_byname<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >

The virtual table for the std::time\_put\_byname<wchar\_t, std::ostreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > class is described by [Table 16-343](#ID_CLS_45_24261_45_0)

**Table 16-343 Primary vtable for time\_put\_byname<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for time\_put\_byname<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > |
| vfunc[0]: | time\_put\_byname<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~time\_put\_byname() |
| vfunc[1]: | time\_put\_byname<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~time\_put\_byname() |
| vfunc[2]: | time\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, tm const\*, char, char) const |

The Run Time Type Information for the std::time\_put\_byname<wchar\_t, std::ostreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > class is described by [Table 16-344](#ID_RTTI_45_24064)

**Table 16-344 typeinfo for time\_put\_byname<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for time\_put\_byname<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > |

#### 16.1.116.2 Interfaces for Class time\_put\_byname<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >

An LSB conforming implementation shall provide the generic methods for Class std::time\_put\_byname<wchar\_t, std::ostreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > specified in [Table 16-345](#ID_TBL_45_LIBSTDCXX_45_CLAWJ_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-345 libstdcxx - Class time\_put\_byname<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > Function Interfaces**

|  |
| --- |
| time\_put\_byname<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~time\_put\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_put\_byname<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~time\_put\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_put\_byname<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~time\_put\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::time\_put\_byname<wchar\_t, std::ostreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > specified in [Table 16-346](#ID_TBL_45_LIBSTDCXX_45_CLAWJ_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-346 libstdcxx - Class time\_put\_byname<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > Data Interfaces**

|  |
| --- |
| typeinfo for time\_put\_byname<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for time\_put\_byname<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for time\_put\_byname<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.117 Class time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >**

#### 16.1.117.1 Class data for time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >

The virtual table for the std::time\_get<char, std::istreambuf\_iterator<char, std::char\_traits<char> > > class is described by [Table 16-347](#ID_CLS_45_24274_45_0)

**Table 16-347 Primary vtable for time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for time\_get<char, istreambuf\_iterator<char, char\_traits<char> > > |
| vfunc[0]: | time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::~time\_get() |
| vfunc[1]: | time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::~time\_get() |
| vfunc[2]: | time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_date\_order() const |
| vfunc[3]: | time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get\_time(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, tm\*) const |
| vfunc[4]: | time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get\_date(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, tm\*) const |
| vfunc[5]: | time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get\_weekday(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, tm\*) const |
| vfunc[6]: | time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get\_monthname(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, tm\*) const |
| vfunc[7]: | time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get\_year(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, tm\*) const |

#### 16.1.117.2 Interfaces for Class time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >

An LSB conforming implementation shall provide the generic methods for Class std::time\_get<char, std::istreambuf\_iterator<char, std::char\_traits<char> > > specified in [Table 16-348](#ID_TBL_45_LIBSTDCXX_45_CLAWK_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-348 libstdcxx - Class time\_get<char, istreambuf\_iterator<char, char\_traits<char> > > Function Interfaces**

|  |
| --- |
| time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::date\_order() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get\_date(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, tm\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get\_time(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, tm\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get\_year(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, tm\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::get\_weekday(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, tm\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_date\_order() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::get\_monthname(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, tm\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get\_weekday(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, tm\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get\_monthname(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, tm\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::\_M\_extract\_via\_format(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, tm\*, char const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::get\_date(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, tm\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::get\_time(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, tm\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::get\_year(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, tm\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::~time\_get()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::~time\_get()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::~time\_get()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::time\_get<char, std::istreambuf\_iterator<char, std::char\_traits<char> > > specified in [Table 16-349](#ID_TBL_45_LIBSTDCXX_45_CLAWK_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-349 libstdcxx - Class time\_get<char, istreambuf\_iterator<char, char\_traits<char> > > Data Interfaces**

|  |
| --- |
| guard variable for time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::id(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::id(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for time\_get<char, istreambuf\_iterator<char, char\_traits<char> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.118 Class time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >**

#### 16.1.118.1 Class data for time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >

The virtual table for the std::time\_get<wchar\_t, std::istreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > class is described by [Table 16-350](#ID_CLS_45_24260_45_0)

**Table 16-350 Primary vtable for time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > |
| vfunc[0]: | time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~time\_get() |
| vfunc[1]: | time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~time\_get() |
| vfunc[2]: | time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_date\_order() const |
| vfunc[3]: | time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get\_time(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, tm\*) const |
| vfunc[4]: | time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get\_date(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, tm\*) const |
| vfunc[5]: | time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get\_weekday(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, tm\*) const |
| vfunc[6]: | time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get\_monthname(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, tm\*) const |
| vfunc[7]: | time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get\_year(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, tm\*) const |

#### 16.1.118.2 Interfaces for Class time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >

An LSB conforming implementation shall provide the generic methods for Class std::time\_get<wchar\_t, std::istreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > specified in [Table 16-351](#ID_TBL_45_LIBSTDCXX_45_CLAWL_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-351 libstdcxx - Class time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > Function Interfaces**

|  |
| --- |
| time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::date\_order() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get\_date(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, tm\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get\_time(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, tm\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get\_year(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, tm\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::get\_weekday(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, tm\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_date\_order() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::get\_monthname(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, tm\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get\_weekday(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, tm\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get\_monthname(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, tm\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::\_M\_extract\_via\_format(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, tm\*, wchar\_t const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::get\_date(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, tm\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::get\_time(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, tm\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::get\_year(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, tm\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~time\_get()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~time\_get()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~time\_get()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::time\_get<wchar\_t, std::istreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > specified in [Table 16-352](#ID_TBL_45_LIBSTDCXX_45_CLAWL_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-352 libstdcxx - Class time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > Data Interfaces**

|  |
| --- |
| guard variable for time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::id(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::id(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.119 Class time\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >**

#### 16.1.119.1 Class data for time\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >

The virtual table for the std::time\_put<char, std::ostreambuf\_iterator<char, std::char\_traits<char> > > class is described by [Table 16-353](#ID_CLS_45_24276_45_0)

**Table 16-353 Primary vtable for time\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for time\_put<char, ostreambuf\_iterator<char, char\_traits<char> > > |
| vfunc[0]: | time\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::~time\_put() |
| vfunc[1]: | time\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::~time\_put() |
| vfunc[2]: | time\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::do\_put(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, tm const\*, char, char) const |

The Run Time Type Information for the std::time\_put<char, std::ostreambuf\_iterator<char, std::char\_traits<char> > > class is described by [Table 16-354](#ID_RTTI_45_24017)

**Table 16-354 typeinfo for time\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >**

|  |  |  |
| --- | --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |  |
| Name | typeinfo name for time\_put<char, ostreambuf\_iterator<char, char\_traits<char> > > |  |
| flags: | 8 |  |
| basetype: | typeinfo for locale::facet | 2 |
| basetype: | typeinfo for time\_base | 2 |

#### 16.1.119.2 Interfaces for Class time\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >

An LSB conforming implementation shall provide the generic methods for Class std::time\_put<char, std::ostreambuf\_iterator<char, std::char\_traits<char> > > specified in [Table 16-355](#ID_TBL_45_LIBSTDCXX_45_CLAWM_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-355 libstdcxx - Class time\_put<char, ostreambuf\_iterator<char, char\_traits<char> > > Function Interfaces**

|  |
| --- |
| time\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::put(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, tm const\*, char const\*, char const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::put(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, tm const\*, char, char) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::do\_put(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, tm const\*, char, char) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::~time\_put()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::~time\_put()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::~time\_put()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::time\_put<char, std::ostreambuf\_iterator<char, std::char\_traits<char> > > specified in [Table 16-356](#ID_TBL_45_LIBSTDCXX_45_CLAWM_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-356 libstdcxx - Class time\_put<char, ostreambuf\_iterator<char, char\_traits<char> > > Data Interfaces**

|  |
| --- |
| guard variable for time\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::id(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| time\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::id(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for time\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for time\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for time\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.120 Class time\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >**

#### 16.1.120.1 Class data for time\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >

The virtual table for the std::time\_put<wchar\_t, std::ostreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > class is described by [Table 16-357](#ID_CLS_45_24262_45_0)

**Table 16-357 Primary vtable for time\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for time\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > |
| vfunc[0]: | time\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~time\_put() |
| vfunc[1]: | time\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~time\_put() |
| vfunc[2]: | time\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, tm const\*, char, char) const |

The Run Time Type Information for the std::time\_put<wchar\_t, std::ostreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > class is described by [Table 16-358](#ID_RTTI_45_24025)

**Table 16-358 typeinfo for time\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >**

|  |  |  |
| --- | --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |  |
| Name | typeinfo name for time\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > |  |
| flags: | 8 |  |
| basetype: | typeinfo for locale::facet | 2 |
| basetype: | typeinfo for time\_base | 2 |

#### 16.1.120.2 Interfaces for Class time\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >

An LSB conforming implementation shall provide the generic methods for Class std::time\_put<wchar\_t, std::ostreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > specified in [Table 16-359](#ID_TBL_45_LIBSTDCXX_45_CLAWN_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-359 libstdcxx - Class time\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > Function Interfaces**

|  |
| --- |
| time\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, tm const\*, wchar\_t const\*, wchar\_t const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, tm const\*, char, char) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, tm const\*, char, char) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~time\_put()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~time\_put()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~time\_put()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::time\_put<wchar\_t, std::ostreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > specified in [Table 16-360](#ID_TBL_45_LIBSTDCXX_45_CLAWN_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-360 libstdcxx - Class time\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > Data Interfaces**

|  |
| --- |
| guard variable for time\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::id(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| time\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::id(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for time\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for time\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for time\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.121 Class moneypunct<char, false>**

#### 16.1.121.1 Class data for moneypunct<char, false>

The virtual table for the std::moneypunct<char, false> class is described by [Table 16-361](#ID_CLS_45_24288_45_0)

**Table 16-361 Primary vtable for moneypunct<char, false>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for moneypunct<char, false> |
| vfunc[0]: | moneypunct<char, false>::~moneypunct() |
| vfunc[1]: | moneypunct<char, false>::~moneypunct() |
| vfunc[2]: | moneypunct<char, false>::do\_decimal\_point() const |
| vfunc[3]: | moneypunct<char, false>::do\_thousands\_sep() const |
| vfunc[4]: | moneypunct<char, false>::do\_grouping() const |
| vfunc[5]: | moneypunct<char, false>::do\_curr\_symbol() const |
| vfunc[6]: | moneypunct<char, false>::do\_positive\_sign() const |
| vfunc[7]: | moneypunct<char, false>::do\_negative\_sign() const |
| vfunc[8]: | moneypunct<char, false>::do\_frac\_digits() const |
| vfunc[9]: | moneypunct<char, false>::do\_pos\_format() const |
| vfunc[10]: | moneypunct<char, false>::do\_neg\_format() const |

#### 16.1.121.2 Interfaces for Class moneypunct<char, false>

An LSB conforming implementation shall provide the generic methods for Class std::moneypunct<char, false> specified in [Table 16-362](#ID_TBL_45_LIBSTDCXX_45_CLAWO_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-362 libstdcxx - Class moneypunct<char, false> Function Interfaces**

|  |
| --- |
| moneypunct<char, false>::neg\_format() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, false>::pos\_format() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, false>::curr\_symbol() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, false>::do\_grouping() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, false>::frac\_digits() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, false>::decimal\_point() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, false>::do\_neg\_format() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, false>::do\_pos\_format() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, false>::negative\_sign() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, false>::positive\_sign() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, false>::thousands\_sep() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, false>::do\_curr\_symbol() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, false>::do\_frac\_digits() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, false>::do\_decimal\_point() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, false>::do\_negative\_sign() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, false>::do\_positive\_sign() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, false>::do\_thousands\_sep() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, false>::grouping() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, false>::\_M\_initialize\_moneypunct(\_\_locale\_struct\*, char const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, false>::~moneypunct()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, false>::~moneypunct()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, false>::~moneypunct()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::moneypunct<char, false> specified in [Table 16-363](#ID_TBL_45_LIBSTDCXX_45_CLAWO_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-363 libstdcxx - Class moneypunct<char, false> Data Interfaces**

|  |
| --- |
| guard variable for moneypunct<char, false>::id(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| moneypunct<char, false>::id(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, false>::intl(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for moneypunct<char, false>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for moneypunct<char, false>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for moneypunct<char, false>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.122 Class moneypunct<char, true>**

#### 16.1.122.1 Class data for moneypunct<char, true>

The virtual table for the std::moneypunct<char, true> class is described by [Table 16-364](#ID_CLS_45_24289_45_0)

**Table 16-364 Primary vtable for moneypunct<char, true>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for moneypunct<char, true> |
| vfunc[0]: | moneypunct<char, true>::~moneypunct() |
| vfunc[1]: | moneypunct<char, true>::~moneypunct() |
| vfunc[2]: | moneypunct<char, true>::do\_decimal\_point() const |
| vfunc[3]: | moneypunct<char, true>::do\_thousands\_sep() const |
| vfunc[4]: | moneypunct<char, true>::do\_grouping() const |
| vfunc[5]: | moneypunct<char, true>::do\_curr\_symbol() const |
| vfunc[6]: | moneypunct<char, true>::do\_positive\_sign() const |
| vfunc[7]: | moneypunct<char, true>::do\_negative\_sign() const |
| vfunc[8]: | moneypunct<char, true>::do\_frac\_digits() const |
| vfunc[9]: | moneypunct<char, true>::do\_pos\_format() const |
| vfunc[10]: | moneypunct<char, true>::do\_neg\_format() const |

#### 16.1.122.2 Interfaces for Class moneypunct<char, true>

An LSB conforming implementation shall provide the generic methods for Class std::moneypunct<char, true> specified in [Table 16-365](#ID_TBL_45_LIBSTDCXX_45_CLAWP_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-365 libstdcxx - Class moneypunct<char, true> Function Interfaces**

|  |
| --- |
| moneypunct<char, true>::neg\_format() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, true>::pos\_format() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, true>::curr\_symbol() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, true>::do\_grouping() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, true>::frac\_digits() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, true>::decimal\_point() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, true>::do\_neg\_format() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, true>::do\_pos\_format() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, true>::negative\_sign() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, true>::positive\_sign() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, true>::thousands\_sep() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, true>::do\_curr\_symbol() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, true>::do\_frac\_digits() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, true>::do\_decimal\_point() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, true>::do\_negative\_sign() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, true>::do\_positive\_sign() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, true>::do\_thousands\_sep() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, true>::grouping() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, true>::\_M\_initialize\_moneypunct(\_\_locale\_struct\*, char const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, true>::~moneypunct()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, true>::~moneypunct()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, true>::~moneypunct()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::moneypunct<char, true> specified in [Table 16-366](#ID_TBL_45_LIBSTDCXX_45_CLAWP_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-366 libstdcxx - Class moneypunct<char, true> Data Interfaces**

|  |
| --- |
| guard variable for moneypunct<char, true>::id(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| moneypunct<char, true>::id(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, true>::intl(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for moneypunct<char, true>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for moneypunct<char, true>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for moneypunct<char, true>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.123 Class moneypunct<wchar\_t, false>**

#### 16.1.123.1 Class data for moneypunct<wchar\_t, false>

The virtual table for the std::moneypunct<wchar\_t, false> class is described by [Table 16-367](#ID_CLS_45_24286_45_0)

**Table 16-367 Primary vtable for moneypunct<wchar\_t, false>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for moneypunct<wchar\_t, false> |
| vfunc[0]: | moneypunct<wchar\_t, false>::~moneypunct() |
| vfunc[1]: | moneypunct<wchar\_t, false>::~moneypunct() |
| vfunc[2]: | moneypunct<wchar\_t, false>::do\_decimal\_point() const |
| vfunc[3]: | moneypunct<wchar\_t, false>::do\_thousands\_sep() const |
| vfunc[4]: | moneypunct<wchar\_t, false>::do\_grouping() const |
| vfunc[5]: | moneypunct<wchar\_t, false>::do\_curr\_symbol() const |
| vfunc[6]: | moneypunct<wchar\_t, false>::do\_positive\_sign() const |
| vfunc[7]: | moneypunct<wchar\_t, false>::do\_negative\_sign() const |
| vfunc[8]: | moneypunct<wchar\_t, false>::do\_frac\_digits() const |
| vfunc[9]: | moneypunct<wchar\_t, false>::do\_pos\_format() const |
| vfunc[10]: | moneypunct<wchar\_t, false>::do\_neg\_format() const |

#### 16.1.123.2 Interfaces for Class moneypunct<wchar\_t, false>

An LSB conforming implementation shall provide the generic methods for Class std::moneypunct<wchar\_t, false> specified in [Table 16-368](#ID_TBL_45_LIBSTDCXX_45_CLAWQ_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-368 libstdcxx - Class moneypunct<wchar\_t, false> Function Interfaces**

|  |
| --- |
| moneypunct<wchar\_t, false>::neg\_format() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, false>::pos\_format() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, false>::curr\_symbol() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, false>::do\_grouping() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, false>::frac\_digits() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, false>::decimal\_point() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, false>::do\_neg\_format() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, false>::do\_pos\_format() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, false>::negative\_sign() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, false>::positive\_sign() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, false>::thousands\_sep() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, false>::do\_curr\_symbol() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, false>::do\_frac\_digits() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, false>::do\_decimal\_point() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, false>::do\_negative\_sign() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, false>::do\_positive\_sign() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, false>::do\_thousands\_sep() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, false>::grouping() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, false>::\_M\_initialize\_moneypunct(\_\_locale\_struct\*, char const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, false>::~moneypunct()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, false>::~moneypunct()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, false>::~moneypunct()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::moneypunct<wchar\_t, false> specified in [Table 16-369](#ID_TBL_45_LIBSTDCXX_45_CLAWQ_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-369 libstdcxx - Class moneypunct<wchar\_t, false> Data Interfaces**

|  |
| --- |
| guard variable for moneypunct<wchar\_t, false>::id(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| moneypunct<wchar\_t, false>::id(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, false>::intl(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for moneypunct<wchar\_t, false>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for moneypunct<wchar\_t, false>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for moneypunct<wchar\_t, false>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.124 Class moneypunct<wchar\_t, true>**

#### 16.1.124.1 Class data for moneypunct<wchar\_t, true>

The virtual table for the std::moneypunct<wchar\_t, true> class is described by [Table 16-370](#ID_CLS_45_24287_45_0)

**Table 16-370 Primary vtable for moneypunct<wchar\_t, true>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for moneypunct<wchar\_t, true> |
| vfunc[0]: | moneypunct<wchar\_t, true>::~moneypunct() |
| vfunc[1]: | moneypunct<wchar\_t, true>::~moneypunct() |
| vfunc[2]: | moneypunct<wchar\_t, true>::do\_decimal\_point() const |
| vfunc[3]: | moneypunct<wchar\_t, true>::do\_thousands\_sep() const |
| vfunc[4]: | moneypunct<wchar\_t, true>::do\_grouping() const |
| vfunc[5]: | moneypunct<wchar\_t, true>::do\_curr\_symbol() const |
| vfunc[6]: | moneypunct<wchar\_t, true>::do\_positive\_sign() const |
| vfunc[7]: | moneypunct<wchar\_t, true>::do\_negative\_sign() const |
| vfunc[8]: | moneypunct<wchar\_t, true>::do\_frac\_digits() const |
| vfunc[9]: | moneypunct<wchar\_t, true>::do\_pos\_format() const |
| vfunc[10]: | moneypunct<wchar\_t, true>::do\_neg\_format() const |

#### 16.1.124.2 Interfaces for Class moneypunct<wchar\_t, true>

An LSB conforming implementation shall provide the generic methods for Class std::moneypunct<wchar\_t, true> specified in [Table 16-371](#ID_TBL_45_LIBSTDCXX_45_CLAWR_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-371 libstdcxx - Class moneypunct<wchar\_t, true> Function Interfaces**

|  |
| --- |
| moneypunct<wchar\_t, true>::neg\_format() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, true>::pos\_format() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, true>::curr\_symbol() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, true>::do\_grouping() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, true>::frac\_digits() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, true>::decimal\_point() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, true>::do\_neg\_format() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, true>::do\_pos\_format() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, true>::negative\_sign() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, true>::positive\_sign() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, true>::thousands\_sep() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, true>::do\_curr\_symbol() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, true>::do\_frac\_digits() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, true>::do\_decimal\_point() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, true>::do\_negative\_sign() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, true>::do\_positive\_sign() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, true>::do\_thousands\_sep() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, true>::grouping() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, true>::\_M\_initialize\_moneypunct(\_\_locale\_struct\*, char const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, true>::~moneypunct()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, true>::~moneypunct()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, true>::~moneypunct()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::moneypunct<wchar\_t, true> specified in [Table 16-372](#ID_TBL_45_LIBSTDCXX_45_CLAWR_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-372 libstdcxx - Class moneypunct<wchar\_t, true> Data Interfaces**

|  |
| --- |
| guard variable for moneypunct<wchar\_t, true>::id(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| moneypunct<wchar\_t, true>::id(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, true>::intl(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for moneypunct<wchar\_t, true>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for moneypunct<wchar\_t, true>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for moneypunct<wchar\_t, true>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.125 Class moneypunct\_byname<char, false>**

#### 16.1.125.1 Class data for moneypunct\_byname<char, false>

The virtual table for the std::moneypunct\_byname<char, false> class is described by [Table 16-373](#ID_CLS_45_24283_45_0)

**Table 16-373 Primary vtable for moneypunct\_byname<char, false>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for moneypunct\_byname<char, false> |
| vfunc[0]: | moneypunct\_byname<char, false>::~moneypunct\_byname() |
| vfunc[1]: | moneypunct\_byname<char, false>::~moneypunct\_byname() |
| vfunc[2]: | moneypunct<char, false>::do\_decimal\_point() const |
| vfunc[3]: | moneypunct<char, false>::do\_thousands\_sep() const |
| vfunc[4]: | moneypunct<char, false>::do\_grouping() const |
| vfunc[5]: | moneypunct<char, false>::do\_curr\_symbol() const |
| vfunc[6]: | moneypunct<char, false>::do\_positive\_sign() const |
| vfunc[7]: | moneypunct<char, false>::do\_negative\_sign() const |
| vfunc[8]: | moneypunct<char, false>::do\_frac\_digits() const |
| vfunc[9]: | moneypunct<char, false>::do\_pos\_format() const |
| vfunc[10]: | moneypunct<char, false>::do\_neg\_format() const |

The Run Time Type Information for the std::moneypunct\_byname<char, false> class is described by [Table 16-374](#ID_RTTI_45_24083)

**Table 16-374 typeinfo for moneypunct\_byname<char, false>**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for moneypunct\_byname<char, false> |

#### 16.1.125.2 Interfaces for Class moneypunct\_byname<char, false>

An LSB conforming implementation shall provide the generic methods for Class std::moneypunct\_byname<char, false> specified in [Table 16-375](#ID_TBL_45_LIBSTDCXX_45_CLAWS_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-375 libstdcxx - Class moneypunct\_byname<char, false> Function Interfaces**

|  |
| --- |
| moneypunct\_byname<char, false>::~moneypunct\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct\_byname<char, false>::~moneypunct\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct\_byname<char, false>::~moneypunct\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::moneypunct\_byname<char, false> specified in [Table 16-376](#ID_TBL_45_LIBSTDCXX_45_CLAWS_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-376 libstdcxx - Class moneypunct\_byname<char, false> Data Interfaces**

|  |
| --- |
| moneypunct\_byname<char, false>::intl(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for moneypunct\_byname<char, false>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for moneypunct\_byname<char, false>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for moneypunct\_byname<char, false>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.126 Class moneypunct\_byname<char, true>**

#### 16.1.126.1 Class data for moneypunct\_byname<char, true>

The virtual table for the std::moneypunct\_byname<char, true> class is described by [Table 16-377](#ID_CLS_45_24282_45_0)

**Table 16-377 Primary vtable for moneypunct\_byname<char, true>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for moneypunct\_byname<char, true> |
| vfunc[0]: | moneypunct\_byname<char, true>::~moneypunct\_byname() |
| vfunc[1]: | moneypunct\_byname<char, true>::~moneypunct\_byname() |
| vfunc[2]: | moneypunct<char, true>::do\_decimal\_point() const |
| vfunc[3]: | moneypunct<char, true>::do\_thousands\_sep() const |
| vfunc[4]: | moneypunct<char, true>::do\_grouping() const |
| vfunc[5]: | moneypunct<char, true>::do\_curr\_symbol() const |
| vfunc[6]: | moneypunct<char, true>::do\_positive\_sign() const |
| vfunc[7]: | moneypunct<char, true>::do\_negative\_sign() const |
| vfunc[8]: | moneypunct<char, true>::do\_frac\_digits() const |
| vfunc[9]: | moneypunct<char, true>::do\_pos\_format() const |
| vfunc[10]: | moneypunct<char, true>::do\_neg\_format() const |

The Run Time Type Information for the std::moneypunct\_byname<char, true> class is described by [Table 16-378](#ID_RTTI_45_24082)

**Table 16-378 typeinfo for moneypunct\_byname<char, true>**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for moneypunct\_byname<char, true> |

#### 16.1.126.2 Interfaces for Class moneypunct\_byname<char, true>

An LSB conforming implementation shall provide the generic methods for Class std::moneypunct\_byname<char, true> specified in [Table 16-379](#ID_TBL_45_LIBSTDCXX_45_CLAWT_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-379 libstdcxx - Class moneypunct\_byname<char, true> Function Interfaces**

|  |
| --- |
| moneypunct\_byname<char, true>::~moneypunct\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct\_byname<char, true>::~moneypunct\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct\_byname<char, true>::~moneypunct\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::moneypunct\_byname<char, true> specified in [Table 16-380](#ID_TBL_45_LIBSTDCXX_45_CLAWT_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-380 libstdcxx - Class moneypunct\_byname<char, true> Data Interfaces**

|  |
| --- |
| moneypunct\_byname<char, true>::intl(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for moneypunct\_byname<char, true>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for moneypunct\_byname<char, true>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for moneypunct\_byname<char, true>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.127 Class moneypunct\_byname<wchar\_t, false>**

#### 16.1.127.1 Class data for moneypunct\_byname<wchar\_t, false>

The virtual table for the std::moneypunct\_byname<wchar\_t, false> class is described by [Table 16-381](#ID_CLS_45_24269_45_0)

**Table 16-381 Primary vtable for moneypunct\_byname<wchar\_t, false>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for moneypunct\_byname<wchar\_t, false> |
| vfunc[0]: | moneypunct\_byname<wchar\_t, false>::~moneypunct\_byname() |
| vfunc[1]: | moneypunct\_byname<wchar\_t, false>::~moneypunct\_byname() |
| vfunc[2]: | moneypunct<wchar\_t, false>::do\_decimal\_point() const |
| vfunc[3]: | moneypunct<wchar\_t, false>::do\_thousands\_sep() const |
| vfunc[4]: | moneypunct<wchar\_t, false>::do\_grouping() const |
| vfunc[5]: | moneypunct<wchar\_t, false>::do\_curr\_symbol() const |
| vfunc[6]: | moneypunct<wchar\_t, false>::do\_positive\_sign() const |
| vfunc[7]: | moneypunct<wchar\_t, false>::do\_negative\_sign() const |
| vfunc[8]: | moneypunct<wchar\_t, false>::do\_frac\_digits() const |
| vfunc[9]: | moneypunct<wchar\_t, false>::do\_pos\_format() const |
| vfunc[10]: | moneypunct<wchar\_t, false>::do\_neg\_format() const |

The Run Time Type Information for the std::moneypunct\_byname<wchar\_t, false> class is described by [Table 16-382](#ID_RTTI_45_24071)

**Table 16-382 typeinfo for moneypunct\_byname<wchar\_t, false>**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for moneypunct\_byname<wchar\_t, false> |

#### 16.1.127.2 Interfaces for Class moneypunct\_byname<wchar\_t, false>

An LSB conforming implementation shall provide the generic methods for Class std::moneypunct\_byname<wchar\_t, false> specified in [Table 16-383](#ID_TBL_45_LIBSTDCXX_45_CLAWU_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-383 libstdcxx - Class moneypunct\_byname<wchar\_t, false> Function Interfaces**

|  |
| --- |
| moneypunct\_byname<wchar\_t, false>::~moneypunct\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct\_byname<wchar\_t, false>::~moneypunct\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct\_byname<wchar\_t, false>::~moneypunct\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::moneypunct\_byname<wchar\_t, false> specified in [Table 16-384](#ID_TBL_45_LIBSTDCXX_45_CLAWU_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-384 libstdcxx - Class moneypunct\_byname<wchar\_t, false> Data Interfaces**

|  |
| --- |
| moneypunct\_byname<wchar\_t, false>::intl(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for moneypunct\_byname<wchar\_t, false>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for moneypunct\_byname<wchar\_t, false>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for moneypunct\_byname<wchar\_t, false>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.128 Class moneypunct\_byname<wchar\_t, true>**

#### 16.1.128.1 Class data for moneypunct\_byname<wchar\_t, true>

The virtual table for the std::moneypunct\_byname<wchar\_t, true> class is described by [Table 16-385](#ID_CLS_45_24268_45_0)

**Table 16-385 Primary vtable for moneypunct\_byname<wchar\_t, true>**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for moneypunct\_byname<wchar\_t, true> |
| vfunc[0]: | moneypunct\_byname<wchar\_t, true>::~moneypunct\_byname() |
| vfunc[1]: | moneypunct\_byname<wchar\_t, true>::~moneypunct\_byname() |
| vfunc[2]: | moneypunct<wchar\_t, true>::do\_decimal\_point() const |
| vfunc[3]: | moneypunct<wchar\_t, true>::do\_thousands\_sep() const |
| vfunc[4]: | moneypunct<wchar\_t, true>::do\_grouping() const |
| vfunc[5]: | moneypunct<wchar\_t, true>::do\_curr\_symbol() const |
| vfunc[6]: | moneypunct<wchar\_t, true>::do\_positive\_sign() const |
| vfunc[7]: | moneypunct<wchar\_t, true>::do\_negative\_sign() const |
| vfunc[8]: | moneypunct<wchar\_t, true>::do\_frac\_digits() const |
| vfunc[9]: | moneypunct<wchar\_t, true>::do\_pos\_format() const |
| vfunc[10]: | moneypunct<wchar\_t, true>::do\_neg\_format() const |

The Run Time Type Information for the std::moneypunct\_byname<wchar\_t, true> class is described by [Table 16-386](#ID_RTTI_45_24070)

**Table 16-386 typeinfo for moneypunct\_byname<wchar\_t, true>**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for moneypunct\_byname<wchar\_t, true> |

#### 16.1.128.2 Interfaces for Class moneypunct\_byname<wchar\_t, true>

An LSB conforming implementation shall provide the generic methods for Class std::moneypunct\_byname<wchar\_t, true> specified in [Table 16-387](#ID_TBL_45_LIBSTDCXX_45_CLAWV_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-387 libstdcxx - Class moneypunct\_byname<wchar\_t, true> Function Interfaces**

|  |
| --- |
| moneypunct\_byname<wchar\_t, true>::~moneypunct\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct\_byname<wchar\_t, true>::~moneypunct\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct\_byname<wchar\_t, true>::~moneypunct\_byname()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::moneypunct\_byname<wchar\_t, true> specified in [Table 16-388](#ID_TBL_45_LIBSTDCXX_45_CLAWV_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-388 libstdcxx - Class moneypunct\_byname<wchar\_t, true> Data Interfaces**

|  |
| --- |
| moneypunct\_byname<wchar\_t, true>::intl(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for moneypunct\_byname<wchar\_t, true>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for moneypunct\_byname<wchar\_t, true>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for moneypunct\_byname<wchar\_t, true>(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.129 Class money\_base**

#### 16.1.129.1 Class data for money\_base

The Run Time Type Information for the std::money\_base class is described by [Table 16-389](#ID_RTTI_45_24010)

**Table 16-389 typeinfo for money\_base**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_class\_type\_info |
| Name | typeinfo name for money\_base |

#### 16.1.129.2 Interfaces for Class money\_base

An LSB conforming implementation shall provide the generic methods for Class std::money\_base specified in [Table 16-390](#ID_TBL_45_LIBSTDCXX_45_CLAWW_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-390 libstdcxx - Class money\_base Function Interfaces**

|  |
| --- |
| money\_base::\_S\_construct\_pattern(char, char, char)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::money\_base specified in [Table 16-391](#ID_TBL_45_LIBSTDCXX_45_CLAWW_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-391 libstdcxx - Class money\_base Data Interfaces**

|  |
| --- |
| money\_base::\_S\_default\_pattern(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_base::\_S\_atoms(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for money\_base(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for money\_base(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.130 Class money\_get<char, istreambuf\_iterator<char, char\_traits<char> > >**

#### 16.1.130.1 Class data for money\_get<char, istreambuf\_iterator<char, char\_traits<char> > >

The virtual table for the std::money\_get<char, std::istreambuf\_iterator<char, std::char\_traits<char> > > class is described by [Table 16-392](#ID_CLS_45_24281_45_0)

**Table 16-392 Primary vtable for money\_get<char, istreambuf\_iterator<char, char\_traits<char> > >**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for money\_get<char, istreambuf\_iterator<char, char\_traits<char> > > |
| vfunc[0]: | money\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::~money\_get() |
| vfunc[1]: | money\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::~money\_get() |
| vfunc[2]: | money\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, bool, ios\_base&, \_Ios\_Iostate&, long double&) const |
| vfunc[3]: | money\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, bool, ios\_base&, \_Ios\_Iostate&, basic\_string<char, char\_traits<char>, allocator<char> >&) const |

The Run Time Type Information for the std::money\_get<char, std::istreambuf\_iterator<char, std::char\_traits<char> > > class is described by [Table 16-393](#ID_RTTI_45_24081)

**Table 16-393 typeinfo for money\_get<char, istreambuf\_iterator<char, char\_traits<char> > >**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for money\_get<char, istreambuf\_iterator<char, char\_traits<char> > > |

#### 16.1.130.2 Interfaces for Class money\_get<char, istreambuf\_iterator<char, char\_traits<char> > >

An LSB conforming implementation shall provide the generic methods for Class std::money\_get<char, std::istreambuf\_iterator<char, std::char\_traits<char> > > specified in [Table 16-394](#ID_TBL_45_LIBSTDCXX_45_CLAWX_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-394 libstdcxx - Class money\_get<char, istreambuf\_iterator<char, char\_traits<char> > > Function Interfaces**

|  |
| --- |
| istreambuf\_iterator<char, char\_traits<char> > money\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::\_M\_extract<false>(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, basic\_string<char, char\_traits<char>, allocator<char> >&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| istreambuf\_iterator<char, char\_traits<char> > money\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::\_M\_extract<true>(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, basic\_string<char, char\_traits<char>, allocator<char> >&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, bool, ios\_base&, \_Ios\_Iostate&, basic\_string<char, char\_traits<char>, allocator<char> >&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, bool, ios\_base&, \_Ios\_Iostate&, long double&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, bool, ios\_base&, \_Ios\_Iostate&, basic\_string<char, char\_traits<char>, allocator<char> >&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, bool, ios\_base&, \_Ios\_Iostate&, long double&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::~money\_get()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::~money\_get()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::~money\_get()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::money\_get<char, std::istreambuf\_iterator<char, std::char\_traits<char> > > specified in [Table 16-395](#ID_TBL_45_LIBSTDCXX_45_CLAWX_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-395 libstdcxx - Class money\_get<char, istreambuf\_iterator<char, char\_traits<char> > > Data Interfaces**

|  |
| --- |
| guard variable for money\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::id(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| money\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::id(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for money\_get<char, istreambuf\_iterator<char, char\_traits<char> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for money\_get<char, istreambuf\_iterator<char, char\_traits<char> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for money\_get<char, istreambuf\_iterator<char, char\_traits<char> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.131 Class money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >**

#### 16.1.131.1 Class data for money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >

The virtual table for the std::money\_get<wchar\_t, std::istreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > class is described by [Table 16-396](#ID_CLS_45_24267_45_0)

**Table 16-396 Primary vtable for money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > |
| vfunc[0]: | money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~money\_get() |
| vfunc[1]: | money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~money\_get() |
| vfunc[2]: | money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, bool, ios\_base&, \_Ios\_Iostate&, long double&) const |
| vfunc[3]: | money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, bool, ios\_base&, \_Ios\_Iostate&, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >&) const |

The Run Time Type Information for the std::money\_get<wchar\_t, std::istreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > class is described by [Table 16-397](#ID_RTTI_45_24069)

**Table 16-397 typeinfo for money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > |

#### 16.1.131.2 Interfaces for Class money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >

An LSB conforming implementation shall provide the generic methods for Class std::money\_get<wchar\_t, std::istreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > specified in [Table 16-398](#ID_TBL_45_LIBSTDCXX_45_CLAWY_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-398 libstdcxx - Class money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > Function Interfaces**

|  |
| --- |
| istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::\_M\_extract<false>(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, basic\_string<char, char\_traits<char>, allocator<char> >&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::\_M\_extract<true>(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, basic\_string<char, char\_traits<char>, allocator<char> >&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, bool, ios\_base&, \_Ios\_Iostate&, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, bool, ios\_base&, \_Ios\_Iostate&, long double&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, bool, ios\_base&, \_Ios\_Iostate&, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> >&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, bool, ios\_base&, \_Ios\_Iostate&, long double&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~money\_get()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~money\_get()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~money\_get()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::money\_get<wchar\_t, std::istreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > specified in [Table 16-399](#ID_TBL_45_LIBSTDCXX_45_CLAWY_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-399 libstdcxx - Class money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > Data Interfaces**

|  |
| --- |
| guard variable for money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::id(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::id(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.132 Class money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >**

#### 16.1.132.1 Class data for money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >

The virtual table for the std::money\_put<char, std::ostreambuf\_iterator<char, std::char\_traits<char> > > class is described by [Table 16-400](#ID_CLS_45_24280_45_0)

**Table 16-400 Primary vtable for money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > > |
| vfunc[0]: | money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::~money\_put() |
| vfunc[1]: | money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::~money\_put() |
| vfunc[2]: | money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::do\_put(ostreambuf\_iterator<char, char\_traits<char> >, bool, ios\_base&, char, long double) const |
| vfunc[3]: | money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::do\_put(ostreambuf\_iterator<char, char\_traits<char> >, bool, ios\_base&, char, basic\_string<char, char\_traits<char>, allocator<char> > const&) const |

The Run Time Type Information for the std::money\_put<char, std::ostreambuf\_iterator<char, std::char\_traits<char> > > class is described by [Table 16-401](#ID_RTTI_45_24080)

**Table 16-401 typeinfo for money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > > |

#### 16.1.132.2 Interfaces for Class money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >

An LSB conforming implementation shall provide the generic methods for Class std::money\_put<char, std::ostreambuf\_iterator<char, std::char\_traits<char> > > specified in [Table 16-402](#ID_TBL_45_LIBSTDCXX_45_CLAWZ_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-402 libstdcxx - Class money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > > Function Interfaces**

|  |
| --- |
| money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::put(ostreambuf\_iterator<char, char\_traits<char> >, bool, ios\_base&, char, basic\_string<char, char\_traits<char>, allocator<char> > const&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::put(ostreambuf\_iterator<char, char\_traits<char> >, bool, ios\_base&, char, long double) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::do\_put(ostreambuf\_iterator<char, char\_traits<char> >, bool, ios\_base&, char, basic\_string<char, char\_traits<char>, allocator<char> > const&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::do\_put(ostreambuf\_iterator<char, char\_traits<char> >, bool, ios\_base&, char, long double) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ostreambuf\_iterator<char, char\_traits<char> > money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::\_M\_insert<false>(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, basic\_string<char, char\_traits<char>, allocator<char> > const&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ostreambuf\_iterator<char, char\_traits<char> > money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::\_M\_insert<true>(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, basic\_string<char, char\_traits<char>, allocator<char> > const&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::~money\_put()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::~money\_put()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::~money\_put()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::money\_put<char, std::ostreambuf\_iterator<char, std::char\_traits<char> > > specified in [Table 16-403](#ID_TBL_45_LIBSTDCXX_45_CLAWZ_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-403 libstdcxx - Class money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > > Data Interfaces**

|  |
| --- |
| guard variable for money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::id(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::id(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.133 Class money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >**

#### 16.1.133.1 Class data for money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >

The virtual table for the std::money\_put<wchar\_t, std::ostreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > class is described by [Table 16-404](#ID_CLS_45_24266_45_0)

**Table 16-404 Primary vtable for money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > |
| vfunc[0]: | money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~money\_put() |
| vfunc[1]: | money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~money\_put() |
| vfunc[2]: | money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, bool, ios\_base&, wchar\_t, long double) const |
| vfunc[3]: | money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, bool, ios\_base&, wchar\_t, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&) const |

The Run Time Type Information for the std::money\_put<wchar\_t, std::ostreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > class is described by [Table 16-405](#ID_RTTI_45_24068)

**Table 16-405 typeinfo for money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > |

#### 16.1.133.2 Interfaces for Class money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >

An LSB conforming implementation shall provide the generic methods for Class std::money\_put<wchar\_t, std::ostreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > specified in [Table 16-406](#ID_TBL_45_LIBSTDCXX_45_CLAXA_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-406 libstdcxx - Class money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > Function Interfaces**

|  |
| --- |
| money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, bool, ios\_base&, wchar\_t, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, bool, ios\_base&, wchar\_t, long double) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, bool, ios\_base&, wchar\_t, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, bool, ios\_base&, wchar\_t, long double) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::\_M\_insert<false>(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::\_M\_insert<true>(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, basic\_string<wchar\_t, char\_traits<wchar\_t>, allocator<wchar\_t> > const&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~money\_put()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~money\_put()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~money\_put()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::money\_put<wchar\_t, std::ostreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > specified in [Table 16-407](#ID_TBL_45_LIBSTDCXX_45_CLAXA_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-407 libstdcxx - Class money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > Data Interfaces**

|  |
| --- |
| guard variable for money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::id(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::id(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.134 Class locale**

#### 16.1.134.1 Interfaces for Class locale

An LSB conforming implementation shall provide the generic methods for Class std::locale specified in [Table 16-408](#ID_TBL_45_LIBSTDCXX_45_CLAXB_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-408 libstdcxx - Class locale Function Interfaces**

|  |
| --- |
| locale::id::\_M\_id() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::name() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::operator==(locale const&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::\_M\_coalesce(locale const&, locale const&, int)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::\_S\_normalize\_category(int)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::\_Impl::\_M\_install\_facet(locale::id const\*, locale::facet const\*)(GLIBCXX\_3.4) [[LSB]](#ID_REFSTD_46_LIBSTDCXX_46_3) |
| locale::\_Impl::\_M\_replace\_facet(locale::\_Impl const\*, locale::id const\*)(GLIBCXX\_3.4) [[LSB]](#ID_REFSTD_46_LIBSTDCXX_46_3) |
| locale::\_Impl::~\_Impl()(GLIBCXX\_3.4) [[LSB]](#ID_REFSTD_46_LIBSTDCXX_46_3) |
| locale::\_Impl::~\_Impl()(GLIBCXX\_3.4) [[LSB]](#ID_REFSTD_46_LIBSTDCXX_46_3) |
| locale::global(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::classic()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::locale(char const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::locale(locale::\_Impl\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::locale(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::locale(locale const&, locale const&, int)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::locale()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::locale(char const\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::locale(locale::\_Impl\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::locale(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::locale(locale const&, char const\*, int)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::locale(locale const&, locale const&, int)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::locale()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::~locale()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::~locale()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::operator=(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::locale specified in [Table 16-409](#ID_TBL_45_LIBSTDCXX_45_CLAXB_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-409 libstdcxx - Class locale Data Interfaces**

|  |
| --- |
| locale::all(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::none(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::time(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::ctype(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::collate(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::numeric(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::messages(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::monetary(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.135 Class locale::facet**

#### 16.1.135.1 Class data for locale::facet

The virtual table for the std::locale::facet class is described by [Table 16-410](#ID_CLS_45_24255_45_0)

**Table 16-410 Primary vtable for locale::facet**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for locale::facet |
| vfunc[0]: | locale::facet::~facet() |
| vfunc[1]: | locale::facet::~facet() |

The Run Time Type Information for the std::locale::facet class is described by [Table 16-411](#ID_RTTI_45_24005)

**Table 16-411 typeinfo for locale::facet**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_class\_type\_info |
| Name | typeinfo name for locale::facet |

#### 16.1.135.2 Interfaces for Class locale::facet

An LSB conforming implementation shall provide the generic methods for Class std::locale::facet specified in [Table 16-412](#ID_TBL_45_LIBSTDCXX_45_CLAXC_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-412 libstdcxx - Class locale::facet Function Interfaces**

|  |
| --- |
| locale::facet::\_S\_get\_c\_name()(GLIBCXX\_3.4.6) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::facet::\_S\_get\_c\_locale()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::facet::\_S\_clone\_c\_locale(\_\_locale\_struct\*&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::facet::\_S\_create\_c\_locale(\_\_locale\_struct\*&, char const\*, \_\_locale\_struct\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::facet::\_S\_destroy\_c\_locale(\_\_locale\_struct\*&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::facet::~facet()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::facet::~facet()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::facet::~facet()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| locale::locale(locale const&, char const\*, int)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::locale::facet specified in [Table 16-413](#ID_TBL_45_LIBSTDCXX_45_CLAXC_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-413 libstdcxx - Class locale::facet Data Interfaces**

|  |
| --- |
| \_\_timepunct\_cache<char>::\_S\_timezones(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_timepunct\_cache<wchar\_t>::\_S\_timezones(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for locale::facet(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for locale::facet(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for locale::facet(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.136 facet functions**

#### 16.1.136.1 Interfaces for facet functions

An LSB conforming implementation shall provide the generic methods for facet functions specified in [Table 16-414](#ID_TBL_45_LIBSTDCXX_45_FACET_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-414 libstdcxx - facet functions Function Interfaces**

|  |
| --- |
| void \_\_convert\_to\_v<double>(char const\*, double&, \_Ios\_Iostate&, \_\_locale\_struct\* const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| void \_\_convert\_to\_v<long double>(char const\*, long double&, \_Ios\_Iostate&, \_\_locale\_struct\* const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| void \_\_convert\_to\_v<float>(char const\*, float&, \_Ios\_Iostate&, \_\_locale\_struct\* const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bool has\_facet<moneypunct<char, false> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bool has\_facet<moneypunct<wchar\_t, false> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bool has\_facet<ctype<wchar\_t> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bool has\_facet<codecvt<char, char, \_\_mbstate\_t> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bool has\_facet<codecvt<wchar\_t, char, \_\_mbstate\_t> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bool has\_facet<collate<char> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bool has\_facet<collate<wchar\_t> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bool has\_facet<num\_get<char, istreambuf\_iterator<char, char\_traits<char> > > >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bool has\_facet<num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bool has\_facet<num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > > >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bool has\_facet<num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bool has\_facet<messages<char> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bool has\_facet<messages<wchar\_t> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bool has\_facet<numpunct<char> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bool has\_facet<numpunct<wchar\_t> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bool has\_facet<time\_get<char, istreambuf\_iterator<char, char\_traits<char> > > >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bool has\_facet<time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bool has\_facet<time\_put<char, ostreambuf\_iterator<char, char\_traits<char> > > >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bool has\_facet<time\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bool has\_facet<money\_get<char, istreambuf\_iterator<char, char\_traits<char> > > >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bool has\_facet<money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bool has\_facet<money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > > >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| bool has\_facet<money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, false> const& use\_facet<moneypunct<char, false> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<char, true> const& use\_facet<moneypunct<char, true> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, false> const& use\_facet<moneypunct<wchar\_t, false> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| moneypunct<wchar\_t, true> const& use\_facet<moneypunct<wchar\_t, true> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_timepunct<char> const& use\_facet<\_\_timepunct<char> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_timepunct<wchar\_t> const& use\_facet<\_\_timepunct<wchar\_t> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<char> const& use\_facet<ctype<char> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ctype<wchar\_t> const& use\_facet<ctype<wchar\_t> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| codecvt<char, char, \_\_mbstate\_t> const& use\_facet<codecvt<char, char, \_\_mbstate\_t> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| codecvt<wchar\_t, char, \_\_mbstate\_t> const& use\_facet<codecvt<wchar\_t, char, \_\_mbstate\_t> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| collate<char> const& use\_facet<collate<char> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| collate<wchar\_t> const& use\_facet<collate<wchar\_t> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<char, istreambuf\_iterator<char, char\_traits<char> > > const& use\_facet<num\_get<char, istreambuf\_iterator<char, char\_traits<char> > > >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > const& use\_facet<num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > > const& use\_facet<num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > > >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > const& use\_facet<num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| messages<char> const& use\_facet<messages<char> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| messages<wchar\_t> const& use\_facet<messages<wchar\_t> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct<char> const& use\_facet<numpunct<char> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| numpunct<wchar\_t> const& use\_facet<numpunct<wchar\_t> >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<char, istreambuf\_iterator<char, char\_traits<char> > > const& use\_facet<time\_get<char, istreambuf\_iterator<char, char\_traits<char> > > >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > const& use\_facet<time\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_put<char, ostreambuf\_iterator<char, char\_traits<char> > > const& use\_facet<time\_put<char, ostreambuf\_iterator<char, char\_traits<char> > > >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| time\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > const& use\_facet<time\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_get<char, istreambuf\_iterator<char, char\_traits<char> > > const& use\_facet<money\_get<char, istreambuf\_iterator<char, char\_traits<char> > > >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > const& use\_facet<money\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > > const& use\_facet<money\_put<char, ostreambuf\_iterator<char, char\_traits<char> > > >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > const& use\_facet<money\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > >(locale const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.137 Class \_\_num\_base**

#### 16.1.137.1 Class data for \_\_num\_base

#### 16.1.137.2 Interfaces for Class \_\_num\_base

An LSB conforming implementation shall provide the generic methods for Class std::\_\_num\_base specified in [Table 16-415](#ID_TBL_45_LIBSTDCXX_45_CLAXD_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-415 libstdcxx - Class \_\_num\_base Function Interfaces**

|  |
| --- |
| \_\_num\_base::\_S\_format\_float(ios\_base const&, char\*, char)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::\_\_num\_base specified in [Table 16-416](#ID_TBL_45_LIBSTDCXX_45_CLAXD_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-416 libstdcxx - Class \_\_num\_base Data Interfaces**

|  |
| --- |
| \_\_num\_base::\_S\_atoms\_in(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_num\_base::\_S\_atoms\_out(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.138 Class num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >**

#### 16.1.138.1 Class data for num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >

The virtual table for the std::num\_get<char, std::istreambuf\_iterator<char, std::char\_traits<char> > > class is described by [Table 16-417](#ID_CLS_45_24278_45_0)

**Table 16-417 Primary vtable for num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for num\_get<char, istreambuf\_iterator<char, char\_traits<char> > > |
| vfunc[0]: | num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::~num\_get() |
| vfunc[1]: | num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::~num\_get() |
| vfunc[2]: | num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, bool&) const |
| vfunc[3]: | num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, long&) const |
| vfunc[4]: | num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, unsigned short&) const |
| vfunc[5]: | num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, unsigned int&) const |
| vfunc[6]: | num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, unsigned long&) const |
| vfunc[7]: | num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, long long&) const |
| vfunc[8]: | num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, unsigned long long&) const |
| vfunc[9]: | num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, float&) const |
| vfunc[10]: | num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, double&) const |
| vfunc[11]: | num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, long double&) const |
| vfunc[12]: | num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, void\*&) const |

The Run Time Type Information for the std::num\_get<char, std::istreambuf\_iterator<char, std::char\_traits<char> > > class is described by [Table 16-418](#ID_RTTI_45_24078)

**Table 16-418 typeinfo for num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for num\_get<char, istreambuf\_iterator<char, char\_traits<char> > > |
| basetype: | typeinfo for locale::facet |

#### 16.1.138.2 Interfaces for Class num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >

An LSB conforming implementation shall provide the generic methods for Class std::num\_get<char, std::istreambuf\_iterator<char, std::char\_traits<char> > > specified in [Table 16-419](#ID_TBL_45_LIBSTDCXX_45_CLAXE_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-419 libstdcxx - Class num\_get<char, istreambuf\_iterator<char, char\_traits<char> > > Function Interfaces**

|  |
| --- |
| istreambuf\_iterator<char, char\_traits<char> > num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::\_M\_extract\_int<unsigned int>(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, unsigned int&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| istreambuf\_iterator<char, char\_traits<char> > num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::\_M\_extract\_int<long>(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, long&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| istreambuf\_iterator<char, char\_traits<char> > num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::\_M\_extract\_int<unsigned long>(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, unsigned long&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| istreambuf\_iterator<char, char\_traits<char> > num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::\_M\_extract\_int<unsigned short>(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, unsigned short&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| istreambuf\_iterator<char, char\_traits<char> > num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::\_M\_extract\_int<long long>(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, long long&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| istreambuf\_iterator<char, char\_traits<char> > num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::\_M\_extract\_int<unsigned long long>(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, unsigned long long&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::\_M\_extract\_float(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, basic\_string<char, char\_traits<char>, allocator<char> >&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, void\*&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, bool&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, double&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, long double&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, float&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, unsigned int&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, long&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, unsigned long&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, unsigned short&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, long long&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, unsigned long long&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, void\*&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, bool&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, double&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, long double&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, float&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, unsigned int&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, long&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, unsigned long&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, unsigned short&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, long long&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::do\_get(istreambuf\_iterator<char, char\_traits<char> >, istreambuf\_iterator<char, char\_traits<char> >, ios\_base&, \_Ios\_Iostate&, unsigned long long&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::~num\_get()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::~num\_get()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::~num\_get()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::num\_get<char, std::istreambuf\_iterator<char, std::char\_traits<char> > > specified in [Table 16-420](#ID_TBL_45_LIBSTDCXX_45_CLAXE_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-420 libstdcxx - Class num\_get<char, istreambuf\_iterator<char, char\_traits<char> > > Data Interfaces**

|  |
| --- |
| guard variable for num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::id(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >::id(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for num\_get<char, istreambuf\_iterator<char, char\_traits<char> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.139 Class num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >**

#### 16.1.139.1 Class data for num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >

The virtual table for the std::num\_get<wchar\_t, std::istreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > class is described by [Table 16-421](#ID_CLS_45_24264_45_0)

**Table 16-421 Primary vtable for num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > |
| vfunc[0]: | num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~num\_get() |
| vfunc[1]: | num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~num\_get() |
| vfunc[2]: | num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, bool&) const |
| vfunc[3]: | num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, long&) const |
| vfunc[4]: | num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, unsigned short&) const |
| vfunc[5]: | num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, unsigned int&) const |
| vfunc[6]: | num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, unsigned long&) const |
| vfunc[7]: | num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, long long&) const |
| vfunc[8]: | num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, unsigned long long&) const |
| vfunc[9]: | num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, float&) const |
| vfunc[10]: | num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, double&) const |
| vfunc[11]: | num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, long double&) const |
| vfunc[12]: | num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, void\*&) const |

The Run Time Type Information for the std::num\_get<wchar\_t, std::istreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > class is described by [Table 16-422](#ID_RTTI_45_24066)

**Table 16-422 typeinfo for num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > |
| basetype: | typeinfo for locale::facet |

#### 16.1.139.2 Interfaces for Class num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >

An LSB conforming implementation shall provide the generic methods for Class std::num\_get<wchar\_t, std::istreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > specified in [Table 16-423](#ID_TBL_45_LIBSTDCXX_45_CLAXF_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-423 libstdcxx - Class num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > Function Interfaces**

|  |
| --- |
| istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::\_M\_extract\_int<unsigned int>(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, unsigned int&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::\_M\_extract\_int<long>(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, long&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::\_M\_extract\_int<unsigned long>(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, unsigned long&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::\_M\_extract\_int<unsigned short>(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, unsigned short&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::\_M\_extract\_int<long long>(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, long long&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::\_M\_extract\_int<unsigned long long>(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, unsigned long long&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::\_M\_extract\_float(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, basic\_string<char, char\_traits<char>, allocator<char> >&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, void\*&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, bool&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, double&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, long double&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, float&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, unsigned int&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, long&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, unsigned long&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, unsigned short&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, long long&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, unsigned long long&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, void\*&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, bool&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, double&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, long double&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, float&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, unsigned int&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, long&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, unsigned long&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, unsigned short&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, long long&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_get(istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, \_Ios\_Iostate&, unsigned long long&) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~num\_get()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~num\_get()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~num\_get()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::num\_get<wchar\_t, std::istreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > specified in [Table 16-424](#ID_TBL_45_LIBSTDCXX_45_CLAXF_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-424 libstdcxx - Class num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > Data Interfaces**

|  |
| --- |
| guard variable for num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::id(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::id(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for num\_get<wchar\_t, istreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.140 Class num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >**

#### 16.1.140.1 Class data for num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >

The virtual table for the std::num\_put<char, std::ostreambuf\_iterator<char, std::char\_traits<char> > > class is described by [Table 16-425](#ID_CLS_45_24277_45_0)

**Table 16-425 Primary vtable for num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > > |
| vfunc[0]: | num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::~num\_put() |
| vfunc[1]: | num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::~num\_put() |
| vfunc[2]: | num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::do\_put(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, bool) const |
| vfunc[3]: | num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::do\_put(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, long) const |
| vfunc[4]: | num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::do\_put(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, unsigned long) const |
| vfunc[5]: | num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::do\_put(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, long long) const |
| vfunc[6]: | num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::do\_put(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, unsigned long long) const |
| vfunc[7]: | num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::do\_put(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, double) const |
| vfunc[8]: | num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::do\_put(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, long double) const |
| vfunc[9]: | num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::do\_put(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, void const\*) const |

The Run Time Type Information for the std::num\_put<char, std::ostreambuf\_iterator<char, std::char\_traits<char> > > class is described by [Table 16-426](#ID_RTTI_45_24077)

**Table 16-426 typeinfo for num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > > |
| basetype: | typeinfo for locale::facet |

#### 16.1.140.2 Interfaces for Class num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >

An LSB conforming implementation shall provide the generic methods for Class std::num\_put<char, std::ostreambuf\_iterator<char, std::char\_traits<char> > > specified in [Table 16-427](#ID_TBL_45_LIBSTDCXX_45_CLAXG_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-427 libstdcxx - Class num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > > Function Interfaces**

|  |
| --- |
| ostreambuf\_iterator<char, char\_traits<char> > num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::\_M\_insert\_int<long>(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, long) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ostreambuf\_iterator<char, char\_traits<char> > num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::\_M\_insert\_int<unsigned long>(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, unsigned long) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ostreambuf\_iterator<char, char\_traits<char> > num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::\_M\_insert\_int<long long>(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, long long) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ostreambuf\_iterator<char, char\_traits<char> > num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::\_M\_insert\_int<unsigned long long>(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, unsigned long long) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ostreambuf\_iterator<char, char\_traits<char> > num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::\_M\_insert\_float<double>(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, char, double) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ostreambuf\_iterator<char, char\_traits<char> > num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::\_M\_insert\_float<long double>(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, char, long double) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::put(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, void const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::put(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, bool) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::put(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, double) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::put(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, long double) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::put(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, long) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::put(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, unsigned long) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::put(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, long long) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::put(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, unsigned long long) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::do\_put(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, void const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::do\_put(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, bool) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::do\_put(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, double) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::do\_put(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, long double) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::do\_put(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, long) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::do\_put(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, unsigned long) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::do\_put(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, long long) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::do\_put(ostreambuf\_iterator<char, char\_traits<char> >, ios\_base&, char, unsigned long long) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::~num\_put()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::~num\_put()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::~num\_put()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::num\_put<char, std::ostreambuf\_iterator<char, std::char\_traits<char> > > specified in [Table 16-428](#ID_TBL_45_LIBSTDCXX_45_CLAXG_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-428 libstdcxx - Class num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > > Data Interfaces**

|  |
| --- |
| guard variable for num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::id(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >::id(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for num\_put<char, ostreambuf\_iterator<char, char\_traits<char> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.141 Class num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >**

#### 16.1.141.1 Class data for num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >

The virtual table for the std::num\_put<wchar\_t, std::ostreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > class is described by [Table 16-429](#ID_CLS_45_24263_45_0)

**Table 16-429 Primary vtable for num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >**

|  |  |
| --- | --- |
| Base Offset | 0 |
| Virtual Base Offset | 0 |
| RTTI | typeinfo for num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > |
| vfunc[0]: | num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~num\_put() |
| vfunc[1]: | num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~num\_put() |
| vfunc[2]: | num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, bool) const |
| vfunc[3]: | num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, long) const |
| vfunc[4]: | num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, unsigned long) const |
| vfunc[5]: | num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, long long) const |
| vfunc[6]: | num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, unsigned long long) const |
| vfunc[7]: | num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, double) const |
| vfunc[8]: | num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, long double) const |
| vfunc[9]: | num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, void const\*) const |

The Run Time Type Information for the std::num\_put<wchar\_t, std::ostreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > class is described by [Table 16-430](#ID_RTTI_45_24065)

**Table 16-430 typeinfo for num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >**

|  |  |
| --- | --- |
| Base Vtable | vtable for \_\_cxxabiv1::\_\_si\_class\_type\_info |
| Name | typeinfo name for num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > |
| basetype: | typeinfo for locale::facet |

#### 16.1.141.2 Interfaces for Class num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >

An LSB conforming implementation shall provide the generic methods for Class std::num\_put<wchar\_t, std::ostreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > specified in [Table 16-431](#ID_TBL_45_LIBSTDCXX_45_CLAXH_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-431 libstdcxx - Class num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > Function Interfaces**

|  |
| --- |
| ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::\_M\_insert\_int<long>(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, long) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::\_M\_insert\_int<unsigned long>(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, unsigned long) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::\_M\_insert\_int<long long>(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, long long) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::\_M\_insert\_int<unsigned long long>(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, unsigned long long) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::\_M\_insert\_float<double>(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, char, double) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::\_M\_insert\_float<long double>(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, char, long double) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, void const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, bool) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, double) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, long double) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, long) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, unsigned long) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, long long) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, unsigned long long) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, void const\*) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, bool) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, double) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, long double) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, long) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, unsigned long) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, long long) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::do\_put(ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> >, ios\_base&, wchar\_t, unsigned long long) const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~num\_put()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~num\_put()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::~num\_put()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

An LSB conforming implementation shall provide the generic data interfaces for Class std::num\_put<wchar\_t, std::ostreambuf\_iterator<wchar\_t, std::char\_traits<wchar\_t> > > specified in [Table 16-432](#ID_TBL_45_LIBSTDCXX_45_CLAXH_45_DATA), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-432 libstdcxx - Class num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > > Data Interfaces**

|  |
| --- |
| guard variable for num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::id(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >::id(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| typeinfo for num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| typeinfo name for num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |
| vtable for num\_put<wchar\_t, ostreambuf\_iterator<wchar\_t, char\_traits<wchar\_t> > >(GLIBCXX\_3.4) [[CXXABI-1.86]](#ID_REFSTD_46_LIBSTDCXX_46_1) |

### **16.1.142 Class \_\_basic\_file<char>**

#### 16.1.142.1 Class data for \_\_basic\_file<char>

#### 16.1.142.2 Interfaces for Class \_\_basic\_file<char>

An LSB conforming implementation shall provide the generic methods for Class std::\_\_basic\_file<char> specified in [Table 16-433](#ID_TBL_45_LIBSTDCXX_45_CLAXI_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-433 libstdcxx - Class \_\_basic\_file<char> Function Interfaces**

|  |
| --- |
| \_\_basic\_file<char>::is\_open() const(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_basic\_file<char>::fd()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_basic\_file<char>::file()(GLIBCXX\_3.4.1) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_basic\_file<char>::open(char const\*, \_Ios\_Openmode, int)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_basic\_file<char>::sync()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_basic\_file<char>::close()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_basic\_file<char>::sys\_open(\_IO\_FILE\*, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_basic\_file<char>::sys\_open(int, \_Ios\_Openmode)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_basic\_file<char>::showmanyc()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_basic\_file<char>::\_\_basic\_file(pthread\_mutex\_t\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_basic\_file<char>::\_\_basic\_file(pthread\_mutex\_t\*)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_basic\_file<char>::~\_\_basic\_file()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| \_\_basic\_file<char>::~\_\_basic\_file()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.143 Class \_List\_node\_base**

#### 16.1.143.1 Interfaces for Class \_List\_node\_base

An LSB conforming implementation shall provide the generic methods for Class std::\_List\_node\_base specified in [Table 16-434](#ID_TBL_45_LIBSTDCXX_45_CLAXJ_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-434 libstdcxx - Class \_List\_node\_base Function Interfaces**

|  |
| --- |
| \_List\_node\_base::hook(\_List\_node\_base\*)(GLIBCXX\_3.4) [[LSB]](#ID_REFSTD_46_LIBSTDCXX_46_3) |
| \_List\_node\_base::swap(\_List\_node\_base&, \_List\_node\_base&)(GLIBCXX\_3.4) [[LSB]](#ID_REFSTD_46_LIBSTDCXX_46_3) |
| \_List\_node\_base::unhook()(GLIBCXX\_3.4) [[LSB]](#ID_REFSTD_46_LIBSTDCXX_46_3) |
| \_List\_node\_base::reverse()(GLIBCXX\_3.4) [[LSB]](#ID_REFSTD_46_LIBSTDCXX_46_3) |
| \_List\_node\_base::transfer(\_List\_node\_base\*, \_List\_node\_base\*)(GLIBCXX\_3.4) [[LSB]](#ID_REFSTD_46_LIBSTDCXX_46_3) |

### **16.1.144 Class allocator<char>**

#### 16.1.144.1 Class data for allocator<char>

#### 16.1.144.2 Interfaces for Class allocator<char>

An LSB conforming implementation shall provide the generic methods for Class std::allocator<char> specified in [Table 16-435](#ID_TBL_45_LIBSTDCXX_45_CLAXK_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-435 libstdcxx - Class allocator<char> Function Interfaces**

|  |
| --- |
| allocator<char>::allocator(allocator<char> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| allocator<char>::allocator()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| allocator<char>::allocator(allocator<char> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| allocator<char>::allocator()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| allocator<char>::~allocator()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| allocator<char>::~allocator()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.145 Class allocator<wchar\_t>**

#### 16.1.145.1 Class data for allocator<wchar\_t>

#### 16.1.145.2 Interfaces for Class allocator<wchar\_t>

An LSB conforming implementation shall provide the generic methods for Class std::allocator<wchar\_t> specified in [Table 16-436](#ID_TBL_45_LIBSTDCXX_45_CLAXL_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-436 libstdcxx - Class allocator<wchar\_t> Function Interfaces**

|  |
| --- |
| allocator<wchar\_t>::allocator(allocator<wchar\_t> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| allocator<wchar\_t>::allocator()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| allocator<wchar\_t>::allocator(allocator<wchar\_t> const&)(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| allocator<wchar\_t>::allocator()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| allocator<wchar\_t>::~allocator()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |
| allocator<wchar\_t>::~allocator()(GLIBCXX\_3.4) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.146 Class \_\_gnu\_cxx::\_\_pool<true>**

#### 16.1.146.1 Interfaces for Class \_\_gnu\_cxx::\_\_pool<true>

An LSB conforming implementation shall provide the generic methods for Class \_\_gnu\_cxx::\_\_pool<true> specified in [Table 16-437](#ID_TBL_45_LIBSTDCXX_45_CLAXM_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-437 libstdcxx - Class \_\_gnu\_cxx::\_\_pool<true> Function Interfaces**

|  |
| --- |
| \_\_gnu\_cxx::\_\_pool<true>::\_M\_destroy()(GLIBCXX\_3.4.4) [[LSB]](#ID_REFSTD_46_LIBSTDCXX_46_3) |
| \_\_gnu\_cxx::\_\_pool<true>::\_M\_initialize(void (\*)(void\*))(GLIBCXX\_3.4.4) [[LSB]](#ID_REFSTD_46_LIBSTDCXX_46_3) |
| \_\_gnu\_cxx::\_\_pool<true>::\_M\_initialize()(GLIBCXX\_3.4.6) [[LSB]](#ID_REFSTD_46_LIBSTDCXX_46_3) |
| \_\_gnu\_cxx::\_\_pool<true>::\_M\_get\_thread\_id()(GLIBCXX\_3.4.4) [[LSB]](#ID_REFSTD_46_LIBSTDCXX_46_3) |
| \_\_gnu\_cxx::\_\_pool<true>::\_M\_destroy\_thread\_key(void\*)(GLIBCXX\_3.4.4) [[LSB]](#ID_REFSTD_46_LIBSTDCXX_46_3) |

### **16.1.147 Class \_\_gnu\_cxx::\_\_pool<false>**

#### 16.1.147.1 Interfaces for Class \_\_gnu\_cxx::\_\_pool<false>

An LSB conforming implementation shall provide the generic methods for Class \_\_gnu\_cxx::\_\_pool<false> specified in [Table 16-438](#ID_TBL_45_LIBSTDCXX_45_CLAXN_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-438 libstdcxx - Class \_\_gnu\_cxx::\_\_pool<false> Function Interfaces**

|  |
| --- |
| \_\_gnu\_cxx::\_\_pool<false>::\_M\_destroy()(GLIBCXX\_3.4.4) [[LSB]](#ID_REFSTD_46_LIBSTDCXX_46_3) |
| \_\_gnu\_cxx::\_\_pool<false>::\_M\_initialize()(GLIBCXX\_3.4.4) [[LSB]](#ID_REFSTD_46_LIBSTDCXX_46_3) |

### **16.1.148 Class \_\_gnu\_cxx::free\_list**

#### 16.1.148.1 Interfaces for Class \_\_gnu\_cxx::free\_list

An LSB conforming implementation shall provide the generic methods for Class \_\_gnu\_cxx::free\_list specified in [Table 16-439](#ID_TBL_45_LIBSTDCXX_45_CLAXO_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-439 libstdcxx - Class \_\_gnu\_cxx::free\_list Function Interfaces**

|  |
| --- |
| \_\_gnu\_cxx::free\_list::\_M\_clear()(GLIBCXX\_3.4.4) [[LSB]](#ID_REFSTD_46_LIBSTDCXX_46_3) |

### **16.1.149 Class char\_traits<char>**

#### 16.1.149.1 Interfaces for Class char\_traits<char>

An LSB conforming implementation shall provide the generic methods for Class std::char\_traits<char> specified in [Table 16-440](#ID_TBL_45_LIBSTDCXX_45_CLAXP_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-440 libstdcxx - Class char\_traits<char> Function Interfaces**

|  |
| --- |
| char\_traits<char>::eq(char const&, char const&)(GLIBCXX\_3.4.5) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

### **16.1.150 Class char\_traits<wchar\_t>**

#### 16.1.150.1 Interfaces for Class char\_traits<wchar\_t>

An LSB conforming implementation shall provide the generic methods for Class std::char\_traits<wchar\_t> specified in [Table 16-441](#ID_TBL_45_LIBSTDCXX_45_CLAXQ_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 16-441 libstdcxx - Class char\_traits<wchar\_t> Function Interfaces**

|  |
| --- |
| char\_traits<wchar\_t>::eq(wchar\_t const&, wchar\_t const&)(GLIBCXX\_3.4.5) [[ISOCXX]](#ID_REFSTD_46_LIBSTDCXX_46_2) |

## **16.2 Interface Definitions for libstdcxx**

The interfaces defined on the following pages are included in libstdcxx and are defined by this specification. Unless otherwise noted, these interfaces shall be included in the source standard.

Other interfaces listed in [Section 16.1](#ID_LIBSTDCXX) shall behave as described in the referenced base document.

# **VI Commands and Utilities**

# **17 Commands and Utilities**

## **17.1 Commands and Utilities**

An LSB conforming implementation shall provide the commands and utilities as described in [Table 17-1](#ID_TBL_45_CMDS), with at least the behavior described as mandatory in the referenced underlying specification, with the following exceptions:

1. If any operand (except one which follows --) starts with a hyphen, the behavior is unspecified.

**Rationale (Informative):** Applications should place options before operands, or use --, as needed. This text is needed because, by default, GNU option parsing differs from POSIX, unless the environment variable POSIXLY\_CORRECT is set. For example, **ls . -a** in GNU **ls** means to list the current directory, showing all files (that is, "." is an operand and -a is an option). In POSIX, "." and -a are both operands, and the command means to list the current directory, and also the file named -a. Suggesting that applications rely on the setting of the POSIXLY\_CORRECT environment variable, or try to set it, seems worse than just asking the applications to invoke commands in ways which work with either the POSIX or GNU behaviors.

**Table 17-1 Commands And Utilities**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| [ [[1]](#ID_STD_45_CMD_45_142) | du [[2]](#ID_STD_45_CMD_45_10) | install [[2]](#ID_STD_45_CMD_45_10) | mv [[1]](#ID_STD_45_CMD_45_142) | strings [[1]](#ID_STD_45_CMD_45_142) |
| ar [[2]](#ID_STD_45_CMD_45_10) | echo [[2]](#ID_STD_45_CMD_45_10) | install\_initd [[2]](#ID_STD_45_CMD_45_10) | newgrp [[2]](#ID_STD_45_CMD_45_10) | strip [[1]](#ID_STD_45_CMD_45_142) |
| at [[2]](#ID_STD_45_CMD_45_10) | ed [[1]](#ID_STD_45_CMD_45_142) | ipcrm [[2]](#ID_STD_45_CMD_45_10) | nice [[1]](#ID_STD_45_CMD_45_142) | stty [[1]](#ID_STD_45_CMD_45_142) |
| awk [[2]](#ID_STD_45_CMD_45_10) | egrep [[2]](#ID_STD_45_CMD_45_10) | ipcs [[2]](#ID_STD_45_CMD_45_10) | nl [[1]](#ID_STD_45_CMD_45_142) | su [[2]](#ID_STD_45_CMD_45_10) |
| basename [[1]](#ID_STD_45_CMD_45_142) | env [[1]](#ID_STD_45_CMD_45_142) | join [[1]](#ID_STD_45_CMD_45_142) | nohup [[1]](#ID_STD_45_CMD_45_142) | sync [[2]](#ID_STD_45_CMD_45_10) |
| batch [[2]](#ID_STD_45_CMD_45_10) | expand [[1]](#ID_STD_45_CMD_45_142) | kill [[1]](#ID_STD_45_CMD_45_142) | od [[2]](#ID_STD_45_CMD_45_10) | tail [[1]](#ID_STD_45_CMD_45_142) |
| bc [[2]](#ID_STD_45_CMD_45_10) | expr [[1]](#ID_STD_45_CMD_45_142) | killall [[2]](#ID_STD_45_CMD_45_10) | passwd [[2]](#ID_STD_45_CMD_45_10) | tar [[2]](#ID_STD_45_CMD_45_10) |
| cat [[1]](#ID_STD_45_CMD_45_142) | false [[1]](#ID_STD_45_CMD_45_142) | ln [[1]](#ID_STD_45_CMD_45_142) | paste [[1]](#ID_STD_45_CMD_45_142) | tee [[1]](#ID_STD_45_CMD_45_142) |
| chfn [[2]](#ID_STD_45_CMD_45_10) | fgrep [[2]](#ID_STD_45_CMD_45_10) | locale [[1]](#ID_STD_45_CMD_45_142) | patch [[2]](#ID_STD_45_CMD_45_10) | test [[1]](#ID_STD_45_CMD_45_142) |
| chgrp [[1]](#ID_STD_45_CMD_45_142) | file [[2]](#ID_STD_45_CMD_45_10) | localedef [[1]](#ID_STD_45_CMD_45_142) | pathchk [[1]](#ID_STD_45_CMD_45_142) | tic [[3]](#ID_STD_45_CMD_45_177) |
| chmod [[1]](#ID_STD_45_CMD_45_142) | find [[1]](#ID_STD_45_CMD_45_142) | logger [[1]](#ID_STD_45_CMD_45_142) | pax [[1]](#ID_STD_45_CMD_45_142) | time [[1]](#ID_STD_45_CMD_45_142) |
| chown [[1]](#ID_STD_45_CMD_45_142) | fold [[1]](#ID_STD_45_CMD_45_142) | logname [[1]](#ID_STD_45_CMD_45_142) | pidof [[2]](#ID_STD_45_CMD_45_10) | touch [[1]](#ID_STD_45_CMD_45_142) |
| chsh [[2]](#ID_STD_45_CMD_45_10) | fuser [[2]](#ID_STD_45_CMD_45_10) | lp [[1]](#ID_STD_45_CMD_45_142) | pr [[1]](#ID_STD_45_CMD_45_142) | tput [[3]](#ID_STD_45_CMD_45_177) |
| cksum [[1]](#ID_STD_45_CMD_45_142) | gencat [[1]](#ID_STD_45_CMD_45_142) | lpr [[2]](#ID_STD_45_CMD_45_10) | printf [[1]](#ID_STD_45_CMD_45_142) | tr [[1]](#ID_STD_45_CMD_45_142) |
| cmp [[1]](#ID_STD_45_CMD_45_142) | getconf [[1]](#ID_STD_45_CMD_45_142) | ls [[2]](#ID_STD_45_CMD_45_10) | ps [[1]](#ID_STD_45_CMD_45_142) | true [[1]](#ID_STD_45_CMD_45_142) |
| col [[2]](#ID_STD_45_CMD_45_10) | gettext [[2]](#ID_STD_45_CMD_45_10) | lsb\_release [[2]](#ID_STD_45_CMD_45_10) | pwd [[1]](#ID_STD_45_CMD_45_142) | tsort [[1]](#ID_STD_45_CMD_45_142) |
| comm [[1]](#ID_STD_45_CMD_45_142) | grep [[2]](#ID_STD_45_CMD_45_10) | m4 [[2]](#ID_STD_45_CMD_45_10) | remove\_initd [[2]](#ID_STD_45_CMD_45_10) | tty [[1]](#ID_STD_45_CMD_45_142) |
| cp [[1]](#ID_STD_45_CMD_45_142) | groupadd [[2]](#ID_STD_45_CMD_45_10) | mailx [[1]](#ID_STD_45_CMD_45_142) | renice [[2]](#ID_STD_45_CMD_45_10) | umount [[2]](#ID_STD_45_CMD_45_10) |
| cpio [[2]](#ID_STD_45_CMD_45_10) | groupdel [[2]](#ID_STD_45_CMD_45_10) | make [[1]](#ID_STD_45_CMD_45_142) | rm [[1]](#ID_STD_45_CMD_45_142) | uname [[1]](#ID_STD_45_CMD_45_142) |
| crontab [[2]](#ID_STD_45_CMD_45_10) | groupmod [[2]](#ID_STD_45_CMD_45_10) | man [[1]](#ID_STD_45_CMD_45_142) | rmdir [[1]](#ID_STD_45_CMD_45_142) | unexpand [[1]](#ID_STD_45_CMD_45_142) |
| csplit [[1]](#ID_STD_45_CMD_45_142) | groups [[2]](#ID_STD_45_CMD_45_10) | md5sum [[2]](#ID_STD_45_CMD_45_10) | sed [[2]](#ID_STD_45_CMD_45_10) | uniq [[1]](#ID_STD_45_CMD_45_142) |
| cut [[1]](#ID_STD_45_CMD_45_142) | gunzip [[2]](#ID_STD_45_CMD_45_10) | mkdir [[1]](#ID_STD_45_CMD_45_142) | sendmail [[2]](#ID_STD_45_CMD_45_10) | useradd [[2]](#ID_STD_45_CMD_45_10) |
| date [[1]](#ID_STD_45_CMD_45_142) | gzip [[2]](#ID_STD_45_CMD_45_10) | mkfifo [[1]](#ID_STD_45_CMD_45_142) | seq [[2]](#ID_STD_45_CMD_45_10) | userdel [[2]](#ID_STD_45_CMD_45_10) |
| dd [[1]](#ID_STD_45_CMD_45_142) | head [[1]](#ID_STD_45_CMD_45_142) | mknod [[2]](#ID_STD_45_CMD_45_10) | sh [[2]](#ID_STD_45_CMD_45_10) | usermod [[2]](#ID_STD_45_CMD_45_10) |
| df [[2]](#ID_STD_45_CMD_45_10) | hostname [[2]](#ID_STD_45_CMD_45_10) | mktemp [[2]](#ID_STD_45_CMD_45_10) | shutdown [[2]](#ID_STD_45_CMD_45_10) | wc [[1]](#ID_STD_45_CMD_45_142) |
| diff [[1]](#ID_STD_45_CMD_45_142) | iconv [[1]](#ID_STD_45_CMD_45_142) | more [[2]](#ID_STD_45_CMD_45_10) | sleep [[1]](#ID_STD_45_CMD_45_142) | xargs [[2]](#ID_STD_45_CMD_45_10) |
| dirname [[1]](#ID_STD_45_CMD_45_142) | id [[1]](#ID_STD_45_CMD_45_142) | mount [[2]](#ID_STD_45_CMD_45_10) | sort [[1]](#ID_STD_45_CMD_45_142) | zcat [[2]](#ID_STD_45_CMD_45_10) |
| dmesg [[2]](#ID_STD_45_CMD_45_10) | infocmp [[3]](#ID_STD_45_CMD_45_177) | msgfmt [[2]](#ID_STD_45_CMD_45_10) | split [[1]](#ID_STD_45_CMD_45_142) |  |

*Referenced Specification(s)*

**[1].** [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4)

**[2].** [This Specification](#ID_STD_46_LSB)

**[3].** [Libncursesw API](#ID_STD_46_LIBNCURSESW)

An LSB conforming implementation shall provide the shell built in utilities as described in [Table 17-2](#ID_TBL_45_BUILTINS), with at least the behavior described as mandatory in the referenced underlying specification, with the following exceptions:

1. The built in commands and utilities shall be provided by the **sh** utility itself, and need not be implemented in a manner so that they can be accessed via the exec family of functions as defined in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) and should not be invoked directly by those standard utilities that execute other utilities ( **env**, **find**, **nice**, **nohup**, **time**, **xargs**).

**Rationale (Informative):** Since the built in utilities must affect the environment of the calling process, they have no effect when executed as a file.

**Table 17-2 Built In Utilities**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| alias [[1]](#ID_STD_45_CMDBI_45_142) | command [[1]](#ID_STD_45_CMDBI_45_142) | getopts [[1]](#ID_STD_45_CMDBI_45_142) | read [[1]](#ID_STD_45_CMDBI_45_142) | umask [[1]](#ID_STD_45_CMDBI_45_142) |
| bg [[1]](#ID_STD_45_CMDBI_45_142) | fc [[1]](#ID_STD_45_CMDBI_45_142) | hash [[1]](#ID_STD_45_CMDBI_45_142) | type [[1]](#ID_STD_45_CMDBI_45_142) | unalias [[1]](#ID_STD_45_CMDBI_45_142) |
| cd [[1]](#ID_STD_45_CMDBI_45_142) | fg [[1]](#ID_STD_45_CMDBI_45_142) | jobs [[1]](#ID_STD_45_CMDBI_45_142) | ulimit [[1]](#ID_STD_45_CMDBI_45_142) | wait [[1]](#ID_STD_45_CMDBI_45_142) |

*Referenced Specification(s)*

**[1].** [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4)

## **17.2 Command Behavior**

This section contains descriptions for commands and utilities whose specified behavior in the LSB contradicts or extends the standards referenced. It also contains commands and utilities only required by the LSB and not specified by other standards.

#### ar

##### Name

ar — create and maintain library archives (DEPRECATED)

##### Description

**ar** is deprecated from the LSB and is expected to disappear from a future version of the LSB.

**Rationale:** The LSB generally does not include software development utilities nor does it specify .o and .a file formats.

**ar** is as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) but with differences as listed below.

##### Differences

-T

-C

  need not be accepted.

-l

  has unspecified behavior.

-q

  has unspecified behavior; using -r is suggested.

#### at

##### Name

at — examine or delete jobs for later execution

##### Description

**at** is as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) but with differences as listed below.

##### Differences

#### Options

-d

  is functionally equivalent to the -r option specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

-r

  need not be supported, but the '-d' option is equivalent.

-t time

  need not be supported.

##### Optional Control Files

The implementation shall support the XSI optional behavior for access control; however the files at.allow and at.deny may reside in /etc rather than /usr/lib/cron.

#### awk

##### Name

awk — pattern scanning and processing language

##### Description

**awk** is as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) but with differences as listed below.

##### Differences

Certain aspects of internationalized regular expressions are optional; see [Regular Expressions](#ID_LOCALIZATION_45_REGEX).

#### batch

##### Name

batch — schedule commands to be executed in a batch queue

##### Description

The specification for **batch** is as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but with differences as listed below.

##### Optional Control Files

The implementation shall support the XSI optional behavior for access control; however the files at.allow and at.deny may reside in /etc rather than /usr/lib/cron.

#### bc

##### Name

bc — an arbitrary precision calculator language

##### Description

**bc** is as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) but with extensions as listed below.

##### Extensions

The bc language may be extended in an implementation defined manner. If an implementation supports extensions, it shall also support the additional options:

-s|--standard

  processes exactly the POSIX **bc** language.

-w|--warn

  gives warnings for extensions to POSIX bc.

#### chfn

##### Name

chfn — change user name and information

##### Synopsis

**chfn** [-f *full\_name*] [-h *home\_phone*] [user]

##### Description

**chfn** shall update the user database. An unprivileged user may only change the fields for their own account, a user with appropriate privileges may change the fields for any account.

The fields *full\_name* and *home\_phone* may contain any character except:

|  |
| --- |
| any control character |
| comma |
| colon |
| equal sign |

If none of the options are selected, **chfn** operates in an interactive fashion. The prompts and expected input in interactive mode are unspecified and should not be relied upon.

As it is possible for the system to be configured to restrict which fields a non-privileged user is permitted to change, applications should be written to gracefully handle these situations.

##### Standard Options

*-f full\_name*

  sets the user's full name.

*-h home\_phone*

  sets the user's home phone number.

##### Future Directions

The following two options are expected to be added in a future version of the LSB:

-o office

  sets the user's office room number.

-p office\_phone

  sets the user's office phone number.

Note that some implementations contain a "-o other" option which specifies an additional field called "other". Traditionally, this field is not subject to the constraints about legitimate characters in fields. Also, one traditionally shall have appropriate privileges to change the other field. At this point there is no consensus about whether it is desirable to specify the other field; applications may wish to avoid using it.

The "-w work\_phone" field found in some implementations should be replaced by the "-p office\_phone" field. The "-r room\_number" field found in some implementations is the equivalent of the "-o office" option mentioned above; which one of these two options to specify will depend on implementation experience and the decision regarding the other field.

#### chsh

##### Name

chsh — change login shell

##### Synopsis

**chsh** [-s *login\_shell*] [*user*]

##### Description

**chsh** changes the user login shell. This determines the name of the user's initial login command. An unprivileged user may only change the login shell for their own account, a user with appropriate privilege may change the login shell for any account specified by *user*.

Unless the user has appropriate privilege, the initial login command name shall be one of those listed in /etc/shells. The *login\_shell* shall be the absolute path (i.e. it must start with '/') to an executable file. Accounts which are restricted (in an implementation-defined manner) may not change their login shell.

If the *-s* option is not selected, **chsh** operates in an interactive mode. The prompts and expected input in this mode are unspecified.

##### Standard Options

*-s login\_shell*

  sets the login shell.

#### col

##### Name

col — filter reverse line feeds from input

##### Description

**col** is as specified in [SUSv2](#ID_STD_46_SUSV2) but with differences as listed below.

##### Differences

The *-p* option has unspecified behavior.

**Note:** Although **col** is shown as legacy in [SUSv2](#ID_STD_46_SUSV2), it is not (yet) deprecated in the LSB.

#### cpio

##### Name

cpio — copy file archives in and out

##### Description

**cpio** is as specified in [SUSv2](#ID_STD_46_SUSV2), but with differences as listed below.

##### Differences

Some elements of the Pattern Matching Notation are optional; see [Pattern Matching Notation](#ID_LOCALIZATION_45_GLOB).

#### crontab

##### Name

crontab — maintain crontab files for individual users

##### Synopsis

**crontab** [-u user] file **crontab** [-u user] {-l | -r | -e}

##### Description

**crontab** is as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but with differences as listed below.

##### Optional Control Files

The implementation shall support the XSI optional behavior for access control; however the files cron.allow and cron.deny may reside in /etc rather than /usr/lib/cron.

#### df

##### Name

df — report file system disk space usage

##### Description

The **df** command shall behave as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but with differences as listed below.

##### Differences

##### Options

If the *-k* option is not specified, disk space is shown in unspecified units. If the *-P* option is specified, the size of the unit shall be printed on the header line in the format "%4s-blocks". Applications should specify *-k*.

The XSI option *-t* has unspecified behavior. Applications should not specify *-t*.

**Rationale:** The most common implementation of **df** uses the *-t* option for a different purpose (restricting output to a particular file system type), and use of *-t* is therefore non-portable.

##### Operand May Identify Special File

If an argument is the absolute file name of a special file containing a mounted file system, **df** shall show the space available on that file system rather than on the file system containing the special file (which is typically the root file system).

**Note:** In [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) the XSI optional behavior permits an operand to name a special file, but appears to require the operation be performed on the file system containing the special file. A defect report has been submitted for this case.

#### dmesg

##### Name

dmesg — print or control the system message buffer

##### Synopsis

**dmesg** [-c | -n *level* | -s *bufsize*]

##### Description

**dmesg** examines or controls the system message buffer. Only a user with appropriate privileges may modify the system message buffer parameters or contents.

##### Standard Options

*-c*

  If the user has appropriate privilege, clears the system message buffer contents after printing.

*-n level*

  If the user has appropriate privilege, sets the level at which logging of messages is done to the console.

*-s bufsize*

  uses a buffer of *bufsize* to query the system message buffer. This is 16392 by default.

#### du

##### Name

du — estimate file space usage

##### Description

**du** is as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but with differences as listed below.

##### Differences

If the *-k* option is not specified, disk space is shown in unspecified units. Applications should specify *-k*.

#### echo

##### Name

echo — write arguments to standard output

##### Synopsis

**echo** [string...]

##### Description

The **echo** command is as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but with the following differences.

Implementations may support implementation-defined options to **echo**. The behavior of **echo** if any arguments contain backslashes is also implementation defined.

##### Application Usage

Conforming applications should not run **echo** with a first argument starting with a hyphen, or with any arguments containing backslashes; they should use **printf** in those cases.

**Note:** The behavior specified here is similar to that specified by [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) without the XSI option. However, the LSB strongly recommends conforming applications not use any options (even if the implementation provides them) while [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) specifies behavior if the first operand is the string *-n*.

#### egrep

##### Name

egrep — search a file with an Extended Regular Expression pattern

##### Description

**egrep** is equivalent to **grep -E**. For further details, see the specification for **grep**.

#### fgrep

##### Name

fgrep — search a file with a fixed pattern

##### Description

**fgrep** is equivalent to grep -F. For further details, see the specification for **grep**.

#### file

##### Name

file — determine file type

##### Description

**file** is as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but with differences as listed below.

##### Differences

The *-M*, *-h*, *-d*, and *-i* options need not be supported.

#### fuser

##### Name

fuser — identify processes using files or sockets

##### Description

**fuser** is as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but with differences as listed below.

##### Differences

The **fuser** command is a system administration utility, see [Path For System Administration Utilities](#ID_FHS_45_SYSADMIN).

#### Option Differences

-c

  has unspecified behavior.

-f

  has unspecified behavior.

#### gettext

##### Name

gettext — retrieve text string from message catalog

##### Synopsis

**gettext** [options] [textdomain] msgid **gettext** -s [options] msgid...

##### Description

The **gettext** utility retrieves a translated text string corresponding to string *msgid* from a message object generated with **msgfmt** utility.

The message object name is derived from the optional argument *textdomain* if present, otherwise from the TEXTDOMAIN environment variable. If no domain is specified, or if a corresponding string cannot be found, **gettext** prints *msgid*.

Ordinarily **gettext** looks for its message object in *dirname*/*lang*/LC\_MESSAGES where *dirname* is the implementation-defined default directory and *lang* is the locale name. If present, the TEXTDOMAINDIR environment variable replaces the *dirname*.

This utility interprets C escape sequences such as \t for tab. Use \\ to print a backslash. To produce a message on a line of its own, either put a \n at the end of *msgid,* or use this command in conjunction with the **printf** utility.

When used with the *-s* option the **gettext** utility behaves like the **echo** utility, except that the message corresponding to *msgid* in the selected catalog provides the arguments.

##### Options

*-d domainname*

*--domain=domainname*

  PARAMETER translated messages from domainname.

*-e*

  Enable expansion of some escape sequences.

*-n*

  Suppress trailing newline.

##### Operands

The following operands are supported:

*textdomain*

  A domain name used to retrieve the messages.

*msgid*

  A key to retrieve the localized message.

##### Environment Variables

LANGUAGE

  Specifies one or more locale names.

LANG

  Specifies locale name.

LC\_MESSAGES

  Specifies messaging locale, and if present overrides LANG for messages.

TEXTDOMAIN

  Specifies the text domain name, which is identical to the message object filename without .mo suffix.

TEXTDOMAINDIR

  Specifies the pathname to the message catalog, and if present replaces the implementation-defined default directory.

##### Exit Status

The following exit values are returned:

0

  Successful completion.

>0

  An error occurred.

#### grep

##### Name

grep — print lines matching a pattern

##### Description

**grep** is as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but with differences as listed below.

##### LSB Differences

Certain aspects of regular expression matching are optional; see [Regular Expressions](#ID_LOCALIZATION_45_REGEX).

#### groupadd

##### Name

groupadd — create a new group

##### Synopsis

**groupadd** [-g gid [-o]] group

##### Description

If the caller has appropriate privilege, the **groupadd** command shall create a new group named *group*. The group name shall be unique in the group database. If no *gid* is specified, **groupadd** shall create the new group with a unique group ID. If the group named *group* already exists, or if a group with *gid* ID exists and *-o* option is not set, **groupadd** shall issue a diagnostic message and exit with a non-zero exit status.

The **groupadd** command is a system administration utility, see [Path For System Administration Utilities](#ID_FHS_45_SYSADMIN).

##### Options

*-g gid [-o]*

  The new group shall have group ID *gid*. If the *-o* option is not used, no other group shall have this group ID. The value of *gid* shall be non-negative.

#### groupdel

##### Name

groupdel — delete a group

##### Synopsis

**groupdel** group

##### Description

If the caller has sufficient privilege, the **groupdel** command shall modify the system group database, deleting the group named *group*. If the group named *group* does not exist or is a primary group for existing user, **groupdel** shall issue a diagnostic message and exit with a non-zero exit status.

The **groupdel** command is a system administration utility, see [Path For System Administration Utilities](#ID_FHS_45_SYSADMIN).

#### groupmod

##### Name

groupmod — modify a group

##### Synopsis

**groupmod** [-g *gid* [-o]] [-n *group\_name*] group

##### Description

If the caller has appropriate privilege, the **groupmod** command shall modify the entry in the system group database corresponding to a group named *group*.

The **groupmod** command is a system administration utility, see [Path For System Administration Utilities](#ID_FHS_45_SYSADMIN).

##### Options

*-g gid [-o]*

  Modify the group's group ID, setting it to *gid*. If the *-o* option is not used, no other group shall have this group ID. The value of *gid*shall be non-negative.

**Note:** Only the group ID in the database is altered; any files with group ownership set to the original group ID are unchanged by this modification.

*-n group\_name*

  changes the name of the group from *group* to *group\_name*.

#### groups

##### Name

groups — display a group

##### Synopsis

**groups** [user]

##### Description

The **groups** command shall behave as **id -Gn *[user]***, as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4). The optional *user* parameter will display the groups for the named user.

#### gunzip

##### Name

gunzip — uncompress files

##### Description

**gunzip** is equivalent to **gzip -d**. See the specification for **gzip** for further details.

[Filesystem Hierarchy Standard](#ID_STD_46_FHS) requires that if **gunzip** exists, it must be a symbolic or hard link to /bin/gzip. This specification additionally allows **gunzip** to be a wrapper script which calls **gzip -d**.

#### gzip

##### Name

gzip — compress or expand files

##### Synopsis

**gzip** [-cdfhlLnNrtvV19] [-S suffix] [name...]

##### Description

The **gzip** command shall attempt to reduce the size of the named files. Whenever possible, each file is replaced by one with the extension .gz, while keeping the same ownership, modes, access and modification times. If no files are specified, or if a file name is -, the standard input is compressed to the standard output. **gzip** shall only attempt to compress regular files. In particular, it will ignore symbolic links.

When compressing, gzip uses the deflate algorithm specified in [RFC 1951: DEFLATE Compressed Data Format Specification](#ID_STD_46_RFC1951) and stores the result in a file using the gzip file format specified in [RFC 1952: GZIP File Format Specification](#ID_STD_46_RFC1952).

##### Options

-c, --stdout, --to-stdout

  writes output on standard output, leaving the original files unchanged. If there are several input files, the output consists of a sequence of independently compressed members. To obtain better compression, concatenate all input files before compressing them.

-d, --decompress, --uncompress

  the name operands are compressed files, and **gzip** shall decompress them.

-f, --force

  forces compression or decompression even if the file has multiple links or the corresponding file already exists, or if the compressed data is read from or written to a terminal. If the input data is not in a format recognized by **gzip**, and if the option *--stdout* is also given, copy the input data without change to the standard ouput: let **gzip** behave as **cat**. If *-f* is not given, and when not running in the background, **gzip** prompts to verify whether an existing file should be overwritten.

-l, --list

  lists the compressed size, uncompressed size, ratio and uncompressed name for each compressed file. For files that are not in **gzip** format, the uncompressed size shall be given as -1. If the *--verbose* or *-v* option is also specified, the crc and timestamp for the uncompressed file shall also be displayed.

For decompression, **gzip** shall support at least the following compression methods:

• deflate ([RFC 1951: DEFLATE Compressed Data Format Specification](#ID_STD_46_RFC1951))

• compress ([POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4))

The crc shall be given as ffffffff for a file not in **gzip** format.

If the *--name* or *-N* option is also specified, the uncompressed name, date and time are those stored within the compressed file, if present.

If the *--quiet* or *-q* option is also specified, the title and totals lines are not displayed.

-L, --license

  displays the **gzip** license and quit.

-n, --no-name

  does not save the original file name and time stamp by default when compressing. (The original name is always saved if the name had to be truncated.) When decompressing, do not restore the original file name if present (remove only the gzip suffix from the compressed file name) and do not restore the original time stamp if present (copy it from the compressed file). This option is the default when decompressing.

-N, --name

  always saves the original file name and time stamp when compressing; this is the default. When decompressing, restore the original file name and time stamp if present. This option is useful on systems which have a limit on file name length or when the time stamp has been lost after a file transfer.

-q, --quiet

  suppresses all warnings.

-r, --recursive

  travels the directory structure recursively. If any of the file names specified on the command line are directories, **gzip** will descend into the directory and compress all the files it finds there (or decompress them in the case of **gunzip**).

-S .suf, --sufix .suf

  uses suffix .suf instead of .gz.

-t, --test

  checks the compressed file integrity.

-v, --verbose

  displays the name and percentage reduction for each file compressed or decompressed.

-#, --fast, --best

  regulates the speed of compression using the specified digit #, where *-1* or *--fast* indicates the fastest compression method (less compression) and *-9* or *--best* indicates the slowest compression method (best compression). The default compression level is *-6* (that is, biased towards high compression at expense of speed).

##### LSB Deprecated Options

The behaviors specified in this section are expected to disappear from a future version of the LSB; applications should only use the non-LSB-deprecated behaviors.

-V, --version

  displays the version number and compilation options, then quits.

#### hostname

##### Name

hostname — show or set the system's host name

##### Synopsis

**hostname** [name]

##### Description

**hostname** is used to either display or, with appropriate privileges, set the current host name of the system. The host name is used by many applications to identify the machine.

When called without any arguments, the program displays the name of the system as returned by the gethostname() function.

When called with a *name* argument, and the user has appropriate privilege, the command sets the host name.

**Note:** It is not specified if the hostname displayed will be a fully qualified domain name. Applications requiring a particular format of hostname should check the output and take appropriate action.

#### install

##### Name

install — copy files and set attributes

##### Synopsis

**install** [option...] SOURCE DEST **install** [option...] SOURCE... DEST **install** [-d | --directory] [option...] DIRECTORY...

##### Description

In the first two formats, copy *SOURCE* to *DEST* or multiple *SOURCE(s)* to the existing *DEST* directory, optionally setting permission modes and file ownership. In the third format, each *DIRECTORY* and any missing parent directories shall be created.

##### Standard Options

--backup[=METHOD]

  makes a backup of each existing destination file. *METHOD* may be one of the following:

*none* or *off*

  never make backups.

*numbered* or *t*

  make numbered backups. A numbered backup has the form "%s.~%d~", target\_name, version\_number. Each backup shall increment the version number by 1.

*existing* or *nil*

  behave as numbered if numbered backups exist, or simple otherwise.

*simple* or *never*

  append a suffix to the name. The default suffix is '~', but can be overriden by setting SIMPLE\_BACKUP\_SUFFIX in the environment, or via the *-S* or *--suffix* option.

If no *METHOD* is specified, the environment variable VERSION\_CONTROL shall be examined for one of the above. Unambiguous abbreviations of *METHOD* shall be accepted. If no *METHOD* is specified, or if *METHOD* is empty, the backup method shall default to existing.

If *METHOD* is invalid or ambiguous, **install** shall fail and issue a diagnostic message.

-b

  is equivalent to *--backup=existing*.

-d, --directory

  treats all arguments as directory names; creates all components of the specified directories.

-D

  creates all leading components of DEST except the last, then copies SOURCE to DEST; useful in the 1st format.

-g GROUP, --group=GROUP

  if the user has appropriate privilege, sets group ownership, instead of process' current group. *GROUP* is either a name in the user group database, or a positive integer, which shall be used as a group-id.

-m MODE, --mode=MODE

  sets permission mode (specified as in **chmod**), instead of the default rwxr-xr-x.

-o OWNER, --owner=OWNER

  if the user has appropriate privilege, sets ownership. *OWNER* is either a name in the user login database, or a positive integer, which shall be used as a user-id.

-p, --preserve-timestamps

  copies the access and modification times of *SOURCE* files to corresponding destination files.

-s, --strip

  strips symbol tables, only for 1st and 2nd formats.

-S SUFFIX, --suffix=SUFFIX

  equivalent to *--backup=existing*, except if a simple suffix is required, use *SUFFIX*.

--verbose

  prints the name of each directory as it is created.

-v, --verbose

  print the name of each file before copying it to stdout.

#### install\_initd

##### Name

install\_initd — activate an init script

##### Synopsis

**/usr/lib/lsb/install\_initd** initd\_file

##### Description

**install\_initd** shall activate a system initialization file that has been copied to an implementation defined location such that this file shall be run at the appropriate point during system initialization. The **install\_initd** command is typically called in the postinstall script of a package, after the script has been copied to /etc/init.d. See also [Installation and Removal of Init Scripts](#ID_INITSRCINSTRM).

#### ipcrm

##### Name

ipcrm — remove IPC Resources

##### Synopsis

**ipcrm** [-q *msgid* | -Q *msgkey* | -s *semid* | -S *semkey* | -m *shmid* | -M *shmkey*]...**ipcrm** [shm | msg | msg] id...

##### Description

If any of the *-q*, *-Q*, *-s*, *-S*, *-m*, *or -M* arguments are given, the **ipcrm** shall behave as described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

Otherwise, **ipcrm** shall remove the resource of the specified type identified by *id*.

##### Future Directions

A future revision of this specification may deprecate the second synopsis form.

**Rationale:** In its first Linux implementation, **ipcrm** used the second syntax shown in the SYNOPSIS. Functionality present in other implementations of **ipcrm** has since been added, namely the ability to delete resources by key (not just identifier), and to respect the same command line syntax. The previous syntax is still supported for backwards compatibility only.

#### ipcs

##### Name

ipcs — provide information on ipc facilities

##### Synopsis

**ipcs** [-smq] [-tcp]

##### Description

**ipcs** provides information on the ipc facilities for which the calling process has read access.

**Note:** Although this command has many similarities with the optional **ipcs** utility described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), it has substantial differences and is therefore described separately. The options specified here have similar meaning to those in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4); other options specified there have unspecified behavior on an LSB conforming implementation. See [Application Usage](#ID_IPCS_46_APP_46_USAGE) below. The output format is not specified.

##### Resource display options

-m

  shared memory segments.

-q

  message queues.

-s

  semaphore arrays.

##### Output format options

-t

  time.

-p

  pid.

-c

  creator.

##### Application Usage

In some implementations of ipcs the *-a* option will print all information available. In other implementations the *-a* option will print all resource types. Therefore, applications shall not use the *-a* option.

Some implementations of **ipcs** provide more output formats than are specified here. These options are not consistent between differing implementations of **ipcs**. Therefore, only the *-t*, *-c* and *-p* option formatting flags may be used. At least one of the *-t*, *-c* and *-p* options and at least one of *-m*, *-q* and *-s* options shall be specified. If no options are specified, the output is unspecified.

#### killall

##### Name

killall — kill processes by name

##### Synopsis

**killall** [-egiqvw] [-signal] name... **killall** -l **killall** -V

##### Description

**killall** sends a signal to all processes running any of the specified commands. If no signal name is specified, SIGTERM is sent.

Signals can be specified either by name (e.g. -HUP) or by number (e.g. -1). Signal 0 (check if a process exists) can only be specified by number.

If the command name contains a slash (/), processes executing that particular file will be selected for killing, independent of their name.

**killall** returns a non-zero return code if no process has been killed for any of the listed commands. If at least one process has been killed for each command, **killall** returns zero.

A **killall** process never kills itself (but may kill other **killall** processes).

##### Standard Options

-e

  requires an exact match for very long names. If a command name is longer than 15 characters, the full name may be unavailable (i.e. it is swapped out). In this case, **killall** will kill everything that matches within the first 15 characters. With -e, such entries are skipped. **killall** prints a message for each skipped entry if -v is specified in addition to -e.

-g

  kills the process group to which the process belongs. The kill signal is only sent once per group, even if multiple processes belonging to the same process group were found.

-i

  asks interactively for confirmation before killing.

-l

  lists all known signal names.

-q

  does not complain if no processes were killed.

-v

  reports if the signal was successfully sent.

##### LSB Deprecated Options

The behaviors specified in this section are expected to disappear from a future version of the LSB; applications should only use the non-LSB-deprecated behaviors.

-V

  displays version information.

#### lpr

##### Name

lpr — off line print

##### Synopsis

**lpr** [-l] [-p] [-Pprinter] [-h] [-s] [-#copies] [-J name] [-T title] [name ......]

##### Description

**lpr** uses a spooling daemon to print the named files when facilities become available. If no names appear, the standard input is assumed.

##### Standard Options

-l

  identifies binary data that is not to be filtered but sent as raw input to printer.

-p

  formats with "pr" before sending to printer.

-Pprinter

  sends output to the printer named printer instead of the default printer.

-h

  suppresses header page.

-s

  uses symbolic links.

-#copies

  specifies copies as the number of copies to print.

-J name

  specifies name as the job name for the header page.

-T title

  specifies title as the title used for "pr".

#### ls

##### Name

ls — list directory contents

##### Description

**ls** shall behave as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but with extensions listed below.

##### Extensions

-l

  If the file is a character special or block special file, the size of the file shall be replaced with two unsigned numbers in the format "%u, %u", representing the major and minor device numbers associated with the special file.

**Note:** The LSB does not specify the meaning of the major and minor devices numbers.

-p

  in addition to [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) XSI optional behavior of printing a slash for a directory, **ls -p** may display other characters for other file types.

#### lsb\_release

##### Name

lsb\_release — print distribution specific information

##### Synopsis

**lsb\_release** [option...]

##### Description

The **lsb\_release** command prints certain LSB (Linux Standard Base) and Distribution information.

If no options are given, the *-v* option is the default.

##### Options

-v, --version

  displays the LSB version the distribution is compliant with. The version is expressed as a colon separated list of versioned LSB module identifiers. An LSB module identifier is a dash-separated tuple consisting of module name and version or module name, version and architecture name, in that order. The version output is presented as a single line of text beginning with LSB Version: followed by a tab character, then the list of LSB module identifiers. This format is intended to be easily parsable by programs which need to consume the data. Excepting core, which must always be present, the list of module identifiers may change over time depending on installations and removals of system software.

Example: LSB Version: core-5.0-amd64:core-5.0-noarch:desktop-5.0-amd64:desktop-5.0-noarch:languages-5.0:imaging-5.0

**Note:** An implementation may support multiple releases of the same module. To claim compliance, the implementation shall list all relevant module identifiers. Version specific library interfaces, if any, will be selected by the program interpreter, which may change from release to release. Version specific commands and utilities, if any, will be described in the relevant specification.

For reporting or querying compliance with this release of the specification, use the module names found in the *Package Dependencies* section of the *Package Format and Installation* chapter for that module specification.

-i, --id

  displays a string identifying the distribution provider. The id output is a single line of text beginning with Distributor ID: followed by a tab character, then the id string. This specification assigns no meaning to the value of the string, the contents are at the discretion of the distribution provider.

Example: Distributor ID: Frobnozz

-d, --description

  displays text describing the distribution. The description output is a single line of text beginning with Description: followed by a tab character, then the description string. This specification assigns no meaning to the value of the string, the contents are at the discretion of the distribution provider.

Example: Description: Frobnozz release 9 (Gilpher)

-r, --release

  displays the release number of distribution. The release output is a single line of text beginning with Release: followed by a tab character, then the release string. This specification assigns no meaning to the value of the string, the contents are at the discretion of the distribution provider.

Example: Release: 9

-c, --codename

  displays a codename which corresponds to the distribution release. The codename output is a single line of text beginning with Codename: followed by a tab character, then the codename string. This specification assigns no meaning to the value of the string, the contents are at the discretion of the distribution provider.

Example: Codename: Gilpher

-a, --all

  displays all of the above information.

-s, --short

  displays all of the above information in a short output format.

-h, --help

  displays a human-readable help message.

#### m4

##### Name

m4 — macro processor

##### Description

**m4** is as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but with extensions as listed below.

##### Extensions

-P

  forces all builtins to be prefixed with m4\_. For example, define becomes m4\_define.

-I *directory*

  Add *directory* to the end of the search path for includes.

#### md5sum

##### Name

md5sum — generate or check MD5 message digests

##### Synopsis

**md5sum** [-c [file] | file]

##### Description

For each file, write to standard output a line containing the MD5 message digest of that file, followed by one or more blank characters, followed by the name of the file. The MD5 message digest shall be calculated according to [RFC 1321: The MD5 Message-Digest Algorithm](#ID_STD_46_RFC1321) and output as 32 hexadecimal digits.

If no file names are specified as operands, read from standard input and use "-" as the file name in the output.

##### Options

-c [file]

  checks the MD5 message digest of all files named in *file* against the message digest listed in the same file. The actual format of *file* is the same as the output of **md5sum**. That is, each line in the file describes a file. If *file* is not specified, read message digests from stdin.

##### Exit Status

**md5sum** shall exit with status 0 if the sum was generated successfully, or, in check mode, if the check matched. Otherwise, **md5sum** shall exit with a non-zero status.

#### mknod

##### Name

mknod — make special files

##### Synopsis

**mknod** [-m *mode* | --mode=*mode*] name type [major minor]**mknod** [--version]

##### Description

The **mknod** command shall create a special file named *name* of the given *type*.

The *type* shall be one of the following:

b

  creates a block (buffered) special file with the specified *major* and *minor* device numbers.

c, u

  creates a character (unbuffered) special file with the specified *major* and *minor* device numbers.

p

  creates a FIFO.

##### Options

-m *mode*, --mode=*mode*

  create the special file with file access permissions set as described in *mode*. The permissions may be any absolute value (i.e. one not containing '+' or '-') acceptable to the **chmod** command.

--version

  output version information and exit.

**Note:** This option may be deprecated in a future release of this specification.

If *type* is p, *major* and *minor* shall not be specified. Otherwise, these parameters are mandatory.

##### Future Directions

This command may be deprecated in a future version of this specification. The *major* and *minor* operands are insufficently portable to be specified usefully here. Only a FIFO can be portably created by this command, and the **mkfifo** command is a simpler interface for that purpose.

#### mktemp

##### Name

mktemp — make temporary file name (unique)

##### Synopsis

**mktemp** [-q] [-u] template

##### Description

The **mktemp** command takes the given file name *template* and overwrites a portion of it to create a file name. This file name shall be unique and suitable for use by the application.

The *template* should have at least six trailing 'X' characters. These characters are replaced with characters from the portable filename character set in order to generate a unique name.

If **mktemp** can successfully generate a unique file name, and the *-u* option is not present, the file shall be created with read and write permission only for the current user. The **mktemp** command shall write the filename generated to the standard output.

##### Options

-q

  fail silently if an error occurs. Diagnostic messages to stderr are suppressed, but the command shall still exit with a non-zero exit status if an error occurs.

-u

  operates in `unsafe' mode. A unique name is generated, but the temporary file shall be unlinked before **mktemp** exits. Use of this option is not encouraged.

#### more

##### Name

more — display files on a page-by-page basis

##### Description

**more** is as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but with differences as listed below.

##### Differences

The **more** command need not respect the LINES and COLUMNS environment variables.

The following additional options may be supported:

-*num*

  specifies an integer which is the screen size (in lines).

+*num*

  starts at line number *num*.

+/*pattern*

  Start at the first line matching the pattern, equivalent to executing the search forward (/) command with the given pattern immediately after opening each file.

The following options from [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) may behave differently:

-e

  has unspecified behavior.

-i

  has unspecified behavior.

-n

  has unspecified behavior.

-p

  Either clear the whole screen before displaying any text (instead of the usual scrolling behavior), or provide the behavior specified by [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4). In the latter case, the syntax is "*-p command*".

-t

  has unspecified behavior.

The **more** command need not support the following interactive commands:

|  |
| --- |
| g |
| G |
| u |
| control u |
| control f |
| newline |
| j |
| k |
| r |
| R |
| m |
| ' (return to mark) |
| /! |
| ? |
| N |
| :e |
| :t |
| control g |
| ZZ |

##### Rationale

The +*num* and +/*string* options are deprecated in [SUSv2](#ID_STD_46_SUSV2), and have been removed in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4); however this specification continues to specify them because the publicly available util-linux package does not support the replacement (-p *command*). The +*command* option as found in [SUSv2](#ID_STD_46_SUSV2) is more general than is specified here, but the util-linux package appears to only support the more specific +*num* and +/*string* forms.

#### mount

##### Name

mount — mount a file system

##### Synopsis

**mount** [-hV]**mount** [-a] [-fFnrsvw] [-t *vfstype*]**mount** [-fnrsvw] [-o *options* [,...]] [device | dir]**mount** [-fnrsvw] [-t *vfstype*] [-o options] device dir

##### Description

As described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), all files in the system are organized in a directed graph, known as the file hierarchy, rooted at /. These files can be spread out over several underlying devices. The **mount** command shall attach the file system found on some underlying device to the file hierarchy.

##### Options

-v

  invoke verbose mode. The **mount** command shall provide diagnostic messages on stdout.

-a

  mount all file systems (of the given types) mentioned in /etc/fstab.

-F

  If the *-a* option is also present, fork a new incarnation of **mount** for each device to be mounted. This will do the mounts on different devices or different NFS servers in parallel.

-f

  cause everything to be done except for the actual system call; if it's not obvious, this `fakes' mounting the file system.

-n

  mount without writing in /etc/mtab. This is necessary for example when /etc is on a read-only file system.

-s

  ignore **mount** options not supported by a file system type. Not all file systems support this option.

-r

  mount the file system read-only. A synonym is *-o ro*.

-w

  mount the file system read/write. (default) A synonym is *-o rw*.

-L label

  If the file /proc/partitions is supported, mount the partition that has the specified label.

-U uuid

  If the file /proc/partitions is supported, mount the partition that has the specified uuid.

-t vfstype

  indicate a file system type of *vfstype*.

More than one type may be specified in a comma separated list. The list of file system types can be prefixed with no to specify the file system types on which no action should be taken.

-o

  options are specified with a *-o* flag followed by a comma-separated string of options. Some of these options are only useful when they appear in the /etc/fstab file. The following options apply to any file system that is being mounted:

async

  perform all I/O to the file system asynchronously.

atime

  update inode access time for each access. (default)

auto

  in /etc/fstab, indicate the device is mountable with *-a*.

defaults

  use default options: rw, suid, dev, exec, auto, nouser, async.

dev

  interpret character or block special devices on the file system.

exec

  permit execution of binaries.

noatime

  do not update file access times on this file system.

noauto

  in /etc/fstab, indicates the device is only explicitly mountable.

nodev

  do not interpret character or block special devices on the file system.

noexec

  do not allow execution of any binaries on the mounted file system.

nosuid

  do not allow set-user-identifier or set-group-identifier bits to take effect.

nouser

  forbid an unprivileged user to mount the file system. (default)

remount

  remount an already-mounted file system. This is commonly used to change the mount options for a file system, especially to make a read-only file system writable.

ro

  mount the file system read-only.

rw

  mount the file system read-write.

suid

  allow set-user-identifier or set-group-identifier bits to take effect.

sync

  do all I/O to the file system synchronously.

user

  allow an unprivilieged user to mount the file system. This option implies the options noexec, nosuid, nodev unless overridden by subsequent options.

##### LSB Deprecated Options

The behaviors specified in this section are expected to disappear from a future version of the LSB; applications should only use the non-LSB-deprecated behaviors.

-V

  output version and exit.

#### msgfmt

##### Name

msgfmt — create a message object from a message file

##### Synopsis

**msgfmt** [options...] *filename*...

##### Description

The **msgfmt** command generates a binary message catalog from a textual translation description. Message catalogs, or message object files, are stored in files with a .mo extension.

**Note:** The format of message object files is not guaranteed to be portable. Message catalogs should always be generated on the target architecture using the **msgfmt** command.

The source message files, otherwise known as portable object files, have a .po extension.

The *filename* operands shall be portable object files. The .po file contains messages to be displayed to users by system utilities or by application programs. The portable object files are text files, and the messages in them can be rewritten in any language supported by the system.

If any *filename* is -, a portable object file shall be read from the standard input.

The **msgfmt** command interprets data as characters according to the current setting of the LC\_CTYPE locale category.

##### Options

-c

--check

  Detect and diagnose input file anomalies which might represent translation errors. The msgid and msgstr strings are studied and compared. It is considered abnormal that one string starts or ends with a newline while the other does not.

If the message is flagged as c-format (see [Comment Handling](#ID_MSGFMT_45_COMMENTS)), check that the msgid string and the msgstr translation have the same number of % format specifiers, with matching types.

-D *directory*

--directory=*directory*

  Add directory to list for input files search. If *filename* is not an absolute pathname and *filename* cannot be opened, search for it in *directory*. This option may be repeated. Directories shall be searched in order, with the leftmost *directory* searched first.

-f

--use-fuzzy

  Use entries marked as fuzzy in output. If this option is not specified, such entries are not included into the output. See [Comment Handling](#ID_MSGFMT_45_COMMENTS) below.

-o *output-file*

--output-file=*output-file*

  Specify the output file name as output-file. If multiple domains or duplicate msgids in the .po file are present, the behavior is unspecified. If output-file is -, output is written to standard output.

--strict

  Ensure that all output files have a .mo extension. Output files are named either by the *-o* (or *--output-file*) option, or by domains found in the input files.

-v

--verbose

  Print additional information to the standard error, including the number of translated strings processed.

##### Operands

The *filename* operands are treated as portable object files. The format of portable object files is defined in EXTENDED DESCRIPTION.

##### Standard Input

The standard input is not used unless a *filename* operand is specified as "-".

##### Environment Variables

LANGUAGE

  Specifies one or more locale names.

LANG

  Specifies locale name.

LC\_ALL

  Specifies locale name for all categories. If defined, overrides LANG, LC\_CTYPE and LC\_MESSAGES.

LC\_CTYPE

  Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files).

LC\_MESSAGES

  Specifies messaging locale, and if present overrides LANG for messages.

##### Standard Output

The standard output is not used unless the option-argument of the *-o* option is specified as -.

##### Extended Description

The format of portable object files (.po files) is defined as follows. Each .po file contains one or more lines, with each line containing either a comment or a statement. Comments start the line with a hash mark (#) and end with the newline character. Empty lines, or lines containing only white-space, shall be ignored. Comments can in certain circumstances alter the behavior of **msgfmt**. See [Comment Handling](#ID_MSGFMT_45_COMMENTS) below for details on comment processing. The format of a statement is:

directive value

Each directive starts at the beginning of the line and is separated from value by white space (such as one or more space or tab characters). The value consists of one or more quoted strings separated by white space. If two or more strings are specified as value, they are normalized into single string using the string normalization syntax specified in [ISO C (1999)](#ID_STD_46_ISOC99). The following directives are supported:

domain domainname

msgid message\_identifier

msgid\_plural untranslated\_string\_plural

msgstr message\_string

msgstr[*n*] message\_string

The behavior of the domain directive is affected by the options used. See OPTIONS for the behavior when the *-o* option is specified. If the *-o* option is not specified, the behavior of the domain directive is as follows:

1. All msgids from the beginning of each .po file to the first domain directive are put into a default message object file, messages (or messages.mo if the *--strict* option is specified).

2. When **msgfmt** encounters a domain domainname directive in the .po file, all following *msgids* until the next domain directive are put into the message object file domainname (or domainname.mo if *--strict* option is specified).

3. Duplicate *msgids* are defined in the scope of each domain. That is, a *msgid* is considered a duplicate only if the identical *msgid* exists in the same domain.

4. All duplicate *msgids* are ignored.

The msgid directive specifies the value of a message identifier associated with the directive that follows it. The msgid\_plural directive specifies the plural form message specified to the plural message handling functions ngettext(), dngettext() or dcngettext(). The message\_identifier string identifies a target string to be used at retrieval time. Each statement containing a msgid directive shall be followed by a statement containing a msgstr directive or msgstr[n] directives.

The msgstr directive specifies the target string associated with the *message\_identifier* string declared in the immediately preceding msgid directive.

The msgstr[*n*] (where *n* = 0, 1, 2, ...) directive specifies the target string to be used with plural form handling functions ngettext(), dngettext() and dcngettext().

Message strings can contain the following escape sequences:

**Table 17-1 Escape Sequences**

|  |  |
| --- | --- |
| \n | newline |
| \t | tab |
| \v | vertical tab |
| \b | backspace |
| \r | carriage return |
| \f | formfeed |
| \\ | backslash |
| \" | double quote |
| \ddd | octal bit pattern |
| \xHH | hexadecimal bit pattern |

##### Comment Handling

Comments are introduced by a #, and continue to the end of the line. The second character (i.e. the character following the #) has special meaning. Regular comments should follow a space character. Other comment types include:

# normal-comments

#. automatic-comments

#: reference...

#, flag

Automatic and reference comments are typically generated by external utilities, and are not specified by the LSB. The **msgfmt** command shall ignore such comments.

**Note:** Portable object files may be produced by unspecified tools. Some of the comment types described here may arise from the use of such tools. It is beyond the scope of this specification to describe these tools.

The #, comments require one or more flags separated by the comma (,) character. The following flags can be specified:

fuzzy

  This flag shows that the following msgstr string might not be a correct translation. Only the translator (i.e. the individual undertaking the translation) can judge if the translation requires further modification, or is acceptable as is. Once satisfied with the translation, the translator then removes this fuzzy flag.

If this flag is specified, the **msgfmt** utility will not generate the entry for the immediately following msgid in the output message catalog, unless the *--use-fuzzy* is specified.

c-format

no-c-format

  The c-format flag indicates that the msgid string is used as format string by printf()-like functions. If the c-format flag is given for a string the **msgfmt** utility may perform additional tests to check the validity of the translation.

##### Plurals

The msgid entry with empty string ("") is called the header entry and is treated specially. If the message string for the header entry contains nplurals=value, the value indicates the number of plural forms. For example, if nplurals=4, there are 4 plural forms. If nplurals is defined, there should be a plural=expression on the same line, separated by a semicolon (;) character. The expression is a C language expression to determine which version of msgstr[n] to be used based on the value of n, the last argument of ngettext(), dngettext() or dcngettext(). For example:

nplurals=2; plural=n == 1 ? 0 : 1

indicates that there are 2 plural forms in the language; msgstr[0] is used if n == 1, otherwise msgstr[1] is used. Another example:

nplurals=3; plural=n==1 ? 0 : n==2 ? 1 : 2

indicates that there are 3 plural forms in the language; msgstr[0] is used if n == 1, msgstr[1] is used if n == 2, otherwise msgstr[2] is used.

If the header entry contains charset=*codeset* string, the *codeset* is used to indicate the codeset to be used to encode the message strings. If the output string's codeset is different from the message string's codeset, codeset conversion from the message strings's codeset to the output string's codeset will be performed upon the call of gettext(), dgettext(), dcgettext(), ngettext(), dngettext(), and dcngettext(). The output string's codeset is determined by the current locale's codeset (the return value of nl\_langinfo(CODESET)) by default, and can be changed by the call of bind\_textdomain\_codeset().

##### Exit Status

The following exit values are returned:

0

  Successful completion.

>0

  An error occurred.

##### Application Usage

Neither **msgfmt** nor any gettext() function imposes a limit on the total length of a message. Installing message catalogs under the C locale is pointless, since they are ignored for the sake of efficiency.

##### Examples

Example 1: Examples of creating message objects from message files.

In this example module1.po, module2.po and module3.po are portable message object files.

example% cat module1.po

# default domain "messages"

msgid "message one"

msgstr "mensaje número uno"

#

domain "help\_domain"

msgid "help two"

msgstr "ayuda número dos"

#

domain "error\_domain"

msgid "error three"

msgstr "error número tres"

example% cat module2.po

# default domain "messages"

msgid "message four"

msgstr "mensaje número cuatro"

#

domain "error\_domain"

msgid "error five"

msgstr "error número cinco"

#

domain "window\_domain"

msgid "window six"

msgstr "ventana número seises"

example% cat module3.po

# default domain "messages"

msgid "message seven"

msgstr "mensaje número siete"

The following command will produce the output files messages, help\_domain, and error\_domain.

example% msgfmt module1.po

The following command will produce the output files messages.mo, help\_domain.mo, error\_domain.mo, and window\_domain.mo.

example% msgfmt module1.po module2.po

The following example will produce the output file hello.mo.

example% msgfmt -o hello.mo module3.po

#### newgrp

##### Name

newgrp — change group ID

##### Synopsis

**newgrp** [group]

##### Description

The **newgrp** command is as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but with differences as listed below.

##### Differences

The *-l* option specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) need not be supported.

#### od

##### Name

od — dump files in octal and other formats

##### Synopsis

**od** [-abcdfilox] [-w *width* | --width-*width*] [-v] [-A *address\_base*] [-j *skip*] [-n *count*] [-t *type\_string*] [file...]**od** --traditional [options] [file] [[+]offset [.] [b]] [[+]label [.] [b]]

##### Description

The **od** command shall provide all of the madatory functionality specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but with extensions and differences to the XSI optional behavior as listed below.

##### Extensions and Differences

-s

  unspecified behavior.

**Note:** Applications wishing to achieve the [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) behavior for *-s* should instead use *-t d2*.

-w*width*, --width[=*width*]

  each output line is limited to *width* bytes from the input.

--traditional

  accepts arguments in traditional form, see [Traditional Usage](#ID_OD_46_TRAD_46_USAGE) below.

**Note:** The XSI optional behavior for offset handling described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) is not supported unless the *--traditional* option is also specified.

##### Pre-POSIX and XSI Specifications

The LSB supports mixing options between the mandatory and XSI optional synopsis forms in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4). The LSB shall support the following options:

-a

  is equivalent to *-t a*, selects named characters.

-b

  is equivalent to *-t o1*, selects octal bytes.

-c

  is equivalent to *-t c*, selects characters.

-d

  is equivalent to *-t u2*, selects unsigned decimal two byte units.

-f

  is equivalent to *-t fF*, selects floats.

-i

  is equivalent to *-t d2*, selects decimal two byte units.

**Note:** This usage may change in future releases; portable applications should use *-t d2*.

-l

  is equivalent to *-t d4*, selects decimal longs.

-o

  is equivalent to *-t o2*, selects octal two byte units.

-x

  is equivalent to *-t x2*, selects hexadecimal two byte units.

Note that the XSI option *-s* need not be supported.

##### Traditional Usage

If the *--traditional* option is specified, there may be between zero and three operands specified.

If no operands are specified, then **od** shall read the standard input.

If there is exactly one operand, and it is an offset of the form [+]offset[.][b], then it shall be interpreted as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4). The file to be dumped shall be the standard input.

If there are exactly two operands, and they are both of the form [+]offset[.][b], then the first shall be treated as an offset (as above), and the second shall be a label, in the same format as the offset. If a label is specified, then the first output line produced for each input block shall be preceded by the input offset, cumulative across input files, of the next byte to be written, followed by the label, in parentheses. The label shall increment in the same manner as the offset.

If there are three operands, then the first shall be the file to dump, the second the offset, and the third the label.

**Note:** Recent versions of **coreutils** contain an **od** utility that conforms to [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4). However, in April 2005, this version was not in widespread use. A future version of this specification may remove the differences.

#### passwd

##### Name

passwd — change user password

##### Synopsis

**passwd** [-x max] [-n min] [-w warn] [-i inact] name **passwd** {-l | -u} name

##### Description

**passwd** changes authentication information for user and group accounts, including passwords and password expiry details, and may be used to enable and disable accounts. Only a user with appropriate privilege may change the password for other users or modify the expiry information.

##### Options

-x max

  sets the maximum number of days a password remains valid.

-n min

  sets the minimum number of days before a password may be changed.

-w warn

  sets the number of days warning the user will receive before their password will expire.

-i inactive

  disables an account after the password has been expired for the given number of days.

-l

  disables an account by changing the password to a value which matches no possible encrypted value.

-u

  re-enables an account by changing the password back to its previous value.

#### patch

##### Name

patch — apply a diff file to an original

##### Description

**patch** is as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but with extensions as listed below.

##### Extensions

--binary

  reads and write all files in binary mode, except for standard output and /dev/tty. This option has no effect on POSIX-compliant systems.

-u, --unified

  interprets the patch file as a unified context diff.

#### pidof

##### Name

pidof — find the process ID of a running program

##### Synopsis

**pidof** [-s] [-x] [-o omitpid...] program...

##### Description

Return the process ID of a process which is running the program named on the command line.

The **pidof** command is a system administration utility, see [Path For System Administration Utilities](#ID_FHS_45_SYSADMIN).

##### Options

-s

  instructs the program to only return one pid.

-x

  causes the program to also return process id's of shells running the named scripts.

-o

  omits processes with specified process id.

#### remove\_initd

##### Name

remove\_initd — clean up init script system modifications introduced by install\_initd

##### Synopsis

**/usr/lib/lsb/remove\_initd** initd\_file

##### Description

**remove\_initd** processes the removal of the modifications made to a distribution's init script system by the **install\_initd** program. This cleanup is performed in the preuninstall script of a package; however, the package manager is still responsible for removing the script from the repository. See also [Installation and Removal of Init Scripts](#ID_INITSRCINSTRM).

#### renice

##### Name

renice — alter priority of running processes

##### Description

**renice** is as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but with differences as listed below.

##### Differences

-n increment

  has unspecified behavior.

#### sed

##### Name

sed — stream editor

##### Description

**sed** is as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but with differences as listed below.

##### LSB Differences

Certain aspects of internationalized regular expressions are optional; see [Regular Expressions](#ID_LOCALIZATION_45_REGEX).

#### sendmail

##### Name

sendmail — an electronic mail transport agent

##### Synopsis

**/usr/sbin/sendmail** [options] [address...]

##### Description

To deliver electronic mail (email), implementations shall provide the interface described here as a command named **sendmail**. This interface shall be the default delivery method for applications needing to transmit an email message.

This program sends an email message to one or more recipients, routing the message as necessary. This program is not intended as a user interface routine.

With no options, **sendmail** reads its standard input up to an end-of-file or a line consisting only of a single dot and sends a copy of the message found there to all of the addresses listed. It determines the network(s) to use based on the syntax and contents of the addresses.

If an address is preceded by a backslash, '\', it is unspecified whether the address is subject to local alias expansion.

The format of messages shall be as defined in [RFC 2822:Internet Message Format](#ID_STD_46_RFC2822).

**Note:** The required name **sendmail** was chosen for historical reasons, but the behaviors specified here are intended to reflect functionality which can be provided by a number of other implementations such as **smail**, **exim** and others. This specification does not require that the specific **sendmail** program be the implementation chosen to provide this functionality.

##### Options

-bm

  read mail from standard input and deliver it to the recipient addresses. This is the default mode of operation.

-bp

  If the user has sufficient privilege, list information about messages currently in the mail queue.

-bs

  use the SMTP protocol as described in [RFC 2821:Simple Mail Transfer Protocol](#ID_STD_46_RFC2821); read SMTP commands on standard input and write SMTP responses on standard output.

In this mode, **sendmail** shall accept \r\n (CR-LF), as required by [RFC 2821:Simple Mail Transfer Protocol](#ID_STD_46_RFC2821), and \n (LF) line terminators.

-F fullname

  explicitly set the full name of the sender for incoming mail unless the message already contains a From: message header.

If the user running **sendmail** is not sufficiently trusted, then the actual sender may be indicated in the message, depending on the configuration of the agent.

-f name

  explicitly set the envelope sender address for incoming mail. If there is no From: header, the address specified in the From: header will also be set.

If the user running **sendmail** is not sufficiently trusted, then the actual sender shall be indicated in the message.

-i

  ignore dots alone on lines by themselves in incoming messages. If this options is not specified, a line consisting of a single dot shall terminate the input. If -bs is also used, the behavior is unspecified.

-odb

  deliver any mail in background, if supported; otherwise ignored.

-odf

  deliver any mail in foreground, if supported; otherwise ignored.

-oem or -em

  mail errors back to the sender. (default)

-oep or -ep

  write errors to the standard error output.

-oeq or -eq

  do not send notification of errors to the sender. This only works for mail delivered locally.

-oi

  is equivalent to -i.

-om

  indicate that the sender of a message should receive a copy of the message if the sender appears in an alias expansion. Ignored if aliases are not supported.

-t

  read the message to obtain recipients from the To:, Cc:, and Bcc: headers in the message instead of from the command arguments. If a Bcc: header is present, it is removed from the message unless there is no To: or Cc: header, in which case a Bcc: header with no data is created, in accordance with [RFC 2822:Internet Message Format](#ID_STD_46_RFC2822).

If there are any operands, the recipients list is unspecified.

This option may be ignored when not in *-bm* mode (the default).

**Note:** It is recommended that applications use as few options as necessary, none if possible.

##### Exit status

The **sendmail** command returns an exit status indicating the results of the operation. The exit codes are as defined in <sysexits.h>.

EX\_OK

  successful completion on all addresses. Note this does not necessarily indicate successful delivery.

EX\_NOUSER

  The user specified did not exist.

EX\_UNAVAILABLE

  A required service or resource was unavailable.

EX\_USAGE

  Command was invoked incorrectly, such as wrong number of arguments, syntax error in arguments, bad flags.

EX\_SOFTWARE

  An internal sofware error was detected. Includes bad arguments.

EX\_OSERROR

  An operating system error was detected. Includes failure to execute fork() or pipe().

EX\_NOHOST

  The host specified did not exist.

EX\_TEMPFAIL

  Temporary failure. Used to indicate the message could not be sent immediately, perhaps because a connection could not be created, but the request can be retried.

#### seq

##### Name

seq — generate a sequence of numbers

##### Synopsis

**/usr/bin/seq** [-f fmt\_str] [-s sep\_str] [first\_num] [inc\_num] last\_num

##### Description

The **seq** command shall output a sequence of numbers from *first\_num* to *last\_num*, stepping by the increment *inc\_num*. The *first\_num* and *last\_num* parameters may be omitted, and default to 1 even when *first\_num* is greater than *last\_num*. Floating-point values may be specified for *first\_num*, *inc\_num*, and *last\_num*.

The *fmt\_str* parameter is a floating point format string like the one used for the printf() function in C.

The *sep\_str* parameter string separates the values that are output. The default is a newline character (/n).

**Note:** If *first\_num* is less than *last\_num* and *inc\_num* is negative, or *first\_num* is greater than *last\_num* and *inc\_num* is positive, **seq** shall not generate any output.

##### Standard Options

-f fmt\_str

  Format the numbers in the output sequence according to *fmt\_str*, a floating point format string like the one used for the printf() function in C.

-s sep\_str

  Separate the numbers in the output sequence with *sep\_str*. The default separator string is a newline character (\n).

first\_num

  The first number in the output sequence. Defaults to 1. May be a floating point value.

inc\_num

  The increment for the output sequence. Defaults to 1. May be a floating point value.

last\_num

  The last number in the output sequence. May be a floating point value.

#### sh

##### Name

sh — shell, the standard command language interpreter

##### Description

The **sh** utility shall behave as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but with extensions listed below.

##### Shell Invocation

The shell shall support an additional option, *-l* (the letter *ell*). If the *-l* option is specified, or if the first character of argument zero (the command name) is a '-', this invokation of the shell is a *login shell*.

An interactive shell, as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), that is also a login shell, or any shell if invoked with the *-l* option, shall, prior to reading from the input file, first read and execute commands from the file /etc/profile, if that file exists, and then from a file called ~/.profile, if such a file exists.

**Note:** This specification requires that the **sh** utility shall also read and execute commands in its current execution environment from all the shell scripts in the directory /etc/profile.d. Such scripts are read and executed as a part of reading and executing /etc/profile.

#### shutdown

##### Name

shutdown — shut the system down

##### Synopsis

**/sbin/shutdown** [-t sec] [-h | -r] [-akfF] time [warning-message]**/sbin/shutdown** -c [warning-message]

##### Description

The **shutdown** command shall shut the system down in a secure way (first synopsis), or cancel a pending shutdown (second synopsis). When the shutdown is initiated, all logged-in users shall be notified immediately that the system is going down, and users shall be prevented from logging in to the system. The *time* specifies when the actual shutdown shall commence. See below for details. At the specified time all processes are first notified that the system is going down by the signal SIGTERM. After an interval (see *-t*) all processes shall be sent the signal SIGKILL. If neither the *-h* or the *-r* argument is specified, then the default behavior shall be to take the system to a runlevel where administrative tasks can be run. See also [Run Levels](#ID_RUNLEVELS).

**Note:** This is sometimes referred to as "single user mode".

The *-h* and *-r* options are mutually exclusive. If either the *-h* or *-r* options are specified, the system shall be halted or rebooted respectively.

##### Standard Options

-a

  use access control. See below.

-t sec

  tell the system to wait *sec* seconds between sending processes the warning and the kill signal, before changing to another runlevel. The default period is unspecified.

-k

  do not really shutdown; only send the warning messages to everybody.

-r

  reboot after shutdown.

-h

  halt after shutdown. Actions after halting are unspecified (e.g. power off).

-f

  advise the system to skip file system consistency checks on reboot.

-F

  advise the system to force file system consistency checks on reboot.

-c

  cancel an already running **shutdown**.

time

  specify when to shut down.

The time argument shall have the following format: [now | [+]mins | hh:mm] If the format is hh:mm, hh shall specify the hour (1 or 2 digits) and mm is the minute of the hour (exactly two digits), and the shutdown shall commence at the next occurence of the specified time. If the format is mins (or +mins), where mins is a decimal number, shutdown shall commence in the specified number of minutes. The word now is an alias for +0.

warning-message

  specify a message to send to all users.

##### Access Control

If the **shutdown** utility is invoked with the *-a* option, it shall check that an authorized user is currently logged in on the system console. Authorized users are listed, one per line, in the file /etc/shutdown.allow. Lines in this file that begin with a '#' or are blank shall be ignored.

**Note:** The intent of this scheme is to allow a keyboard sequence entered on the system console (e.g. CTRL-ALT-DEL, or STOP-A) to automatically invoke **shutdown -a**, and can be used to prevent unauthorized users from shutting the system down in this fashion.

#### su

##### Name

su — change user ID

##### Synopsis

**su** [options] [-] [username [ARGS]]

##### Description

The **su** command shall start a shell running with the real and effective user and group IDs of the user *username*. If *username* is not specified, **su** shall default to an unspecified user with all appropriate privileges. If the *-s* or *--shell* is not specified, the shell to be invoked shall be that specified for *username* in the user database (see getpwnam()), or /bin/sh if there is no shell specified in the user database.

If the *-* option is specified, or if the first operand is -, the environment for the shell shall be initialized as if the new shell was a login shell (see [Shell Invocation](#ID_CMD_45_SH_45_INVOKE)).

If the invoking user does not have appropriate privileges, the **su** command shall prompt for a password and validate this before continuing. Invalid passwords shall produce an error message. The **su** command shall log in an unspecified manner all invokations, whether successful or unsuccessful.

Any operands specified after the *username* shall be passed to the invoked shell.

If the option *-* is not specified, and if the first operand is not -, the environemnt for the new shell shall be intialized from the current environment. If none of the *-m*, *-p*, or *--preserve-environment* options are specified, the environment may be modified in unspecified ways before invoking the shell. If any of the *-m*, *-p*, or *--preserve-environment* options are specified, the environment shall not be altered.

**Note:** Although the **su** command shall not alter the environment, the invoked shell may still alter it before it is ready to intepret any commands.

##### Standard Options

-

  the invoked shell shall be a login shell.

-c *command*, --command=*command*

  Invoke the shell with the option -c *command*.

-m, -p, --preserve-environment

  The current environment shall be passed to the invoked shell. If the environment variable SHELL is set, it shall specify the shell to invoke, if it matches an entry in /etc/shells. If there is no matching entry in /etc/shells, this option shall be ignored if the *-* option is also specified, or if the first operand is -.

-s *shell*, --shell=*shell*

  Invoke *shell* as the comamnd interpreter. The shell specified shall be present in /etc/shells.

#### sync

##### Name

sync — flush file system buffers

##### Synopsis

**sync**

##### Description

Force changed blocks to disk, update the super block.

#### tar

##### Name

tar — file archiver

##### Description

**tar** is as specified in [SUSv2](#ID_STD_46_SUSV2), but with differences as listed below.

##### Differences

Some elements of the Pattern Matching Notation are optional; see [Pattern Matching Notation](#ID_LOCALIZATION_45_GLOB).

-h

  doesn't dump symlinks; dumps the files they point to.

-z

  filters the archive through **gzip**.

#### umount

##### Name

umount — unmount file systems

##### Synopsis

**umount** [-hV]**umount** -a [-nrv] [-t vfstype]**umount** [-nrv] device | dir

##### Description

**umount** detaches the file system(s) mentioned from the file hierarchy. A file system is specified by giving the directory where it has been mounted.

##### Standard Options

-v

  invokes verbose mode.

-n

  unmounts without writing in /etc/mtab.

-r

  tries to remount read-only if unmounting fails.

-a

  unmounts all of the file systems described in /etc/mtab except for the proc file system.

-t vfstype

  indicates that the actions should only be taken on file systems of the specified type. More than one type may be specified in a comma separated list. The list of file system types can be prefixed with no to specify the file system types on which no action should be taken.

-f

  forces unmount (in case of an unreachable NFS system).

##### LSB Deprecated Options

The behaviors specified in this section are expected to disappear from a future version of the LSB; applications should only use the non-LSB-deprecated behaviors.

-V

  print version and exits.

#### useradd

##### Name

useradd — create a new user or update default new user information

##### Synopsis

**useradd** [-c comment] [-d home\_dir] [-g initial\_group] [-G group...] [-m [-k skeleton\_dir]] [-p passwd] [-r] [-s shell] [-u uid [-o]] login **useradd** -D [-g default\_group] [-b default\_home] [-s default\_shell]

##### Description

When invoked without the *-D* option, and with appropriate privilege, **useradd** creates a new user account using the values specified on the command line and the default values from the system. The new user account will be entered into the system files as needed, the home directory will be created, and initial files copied, depending on the command line options. If the group named *group* does not exist, or the user names *login* already exists, or the user ID *uid* is already in use and no *-o* option is specified, then **useradd** shall issue a diagnostic message and exit with a non-zero exit status.

When invoked with the *-D* option, **useradd** will either display the current default values, or, with appropriate privilege, update the default values from the command line. If no options are specified, **useradd** displays the current default values.

The **useradd** command is a system administration utility, see [Path For System Administration Utilities](#ID_FHS_45_SYSADMIN).

##### Standard Options

-c comment

  specifies the new user's password file comment field value.

-d home\_dir

  creates the new user using home\_dir as the value for the user's login directory. The default is to append the login name to default\_home and use that as the login directory name.

-g initial\_group

  specifies the group name or number of the user's initial login group. The group name shall exist. A group number shall refer to an already existing group. If *-g* is not specified, the implementation will follow the normal user default for that system. This may create a new group or choose a default group that normal users are placed in. Applications which require control of the groups into which a user is placed should specify *-g*.

-G group[,...]

  specifies a list of supplementary groups which the user is also a member of. Each group is separated from the next by a comma, with no intervening whitespace. The groups are subject to the same restrictions as the group given with the *-g* option. The default is for the user to belong only to the initial group.

-m [-k skeleton\_dir]

  specifies the user's home directory will be created if it does not exist. The files contained in skeleton\_dir will be copied to the home directory if the *-k* option is used, otherwise the files contained in /etc/skel will be used instead. Any directories contained in skeleton\_dir or /etc/skel will be created in the user's home directory as well. The *-k* option is only valid in conjunction with the *-m* option. The default is to not create the directory and to not copy any files.

-p passwd

  is the encrypted password, as returned by crypt(). The default is to disable the account.

-r

  creates a system account, that is, a user with a User ID in the range reserved for system account users. If there is not a User ID free in the reserved range the command will fail.

-s shell

  specifies the name of the user's login shell. The default is to leave this field blank, which causes the system to select the default login shell.

-u uid [-o]

  specifies the numerical value of the user's ID. This value shall be unique, unless the *-o* option is used. The value shall be non-negative. The default is the smallest ID value greater than 499 which is not yet used.

##### Change Default Options

-b default\_home

  specifies the initial path prefix for a new user's home directory. The user's name will be affixed to the end of default\_home to create the new directory name if the -d option is not used when creating a new account.

-g default\_group

  specifies the group name or ID for a new user's initial group. The named group shall exist, and a numerical group ID shall have an existing entry.

-s default\_shell

  specifies the name of the new user's login shell. The named program will be used for all future new user accounts.

-c comment

  specifies the new user's password file comment field value.

##### Application Usage

The *-D* option will typically be used by system administration packages. Most applications should not change defaults which will affect other applications and users.

#### userdel

##### Name

userdel — delete a user account and related files

##### Synopsis

**userdel** [-r] login

##### Description

Delete the user account named *login*. If there is also a group named *login*, this command may delete the group as well, or may leave it alone. If the user named *login* does not exist or is currently logged in, **userdel** shall issue a diagnostic message and exit with a non-zero exit status.

The **userdel** command is a system administration utility, see [Path For System Administration Utilities](#ID_FHS_45_SYSADMIN).

##### Options

-r

  removes files in the user's home directory along with the home directory itself. Files located in other file system will have to be searched for and deleted manually.

#### usermod

##### Name

usermod — modify a user account

##### Synopsis

**usermod** [-c comment] [-d home\_dir [ -m]] [-g initial\_group] [-G group [,...]] [-l login\_name] [-p passwd] [-s shell] [-u uid [ -o]] login

##### Description

The **usermod** command shall modify an entry in the user account database.

The **usermod** command is a system administration utility, see [Path For System Administration Utilities](#ID_FHS_45_SYSADMIN).

##### Options

-c comment

  specifies the new value of the user's password file comment field.

-d home\_dir

  specifies the user's new login directory. If the -m option is given the contents of the current home directory will be moved to the new home directory, which is created if it does not already exist.

-g initial\_group

  specifies the group name or number of the user's new initial login group. The group name shall exist. A group number shall refer to an already existing group.

-G group,[...]

  specifies a list of supplementary groups which the user is also a member of. Each group is separated from the next by a comma, with no intervening whitespace. The groups are subject to the same restrictions as the group given with the -g option. If the user is currently a member of a group which is not listed, the user will be removed from the group.

-l login\_name

  changes the name of the user from login to login\_name. Nothing else is changed. In particular, the user's home directory name should probably be changed to reflect the new login name.

-p passwd

  is the encrypted password, as returned by crypt(3).

-s shell

  specifies the name of the user's new login shell. Setting this field to blank causes the system to select the default login shell.

-u uid [-o]

  specifies the numerical value of the user's ID. This value shall be unique, unless the -o option is used. The value shall be non-negative. Any files which the user owns and which are located in the directory tree rooted at the user's home directory will have the file user ID changed automatically. Files outside of the user's home directory shall be altered manually.

#### xargs

##### Name

xargs — build and execute command lines from standard input

##### Description

**xargs** is as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), but with differences as listed below.

##### Differences

-E

  has unspecified behavior.

-I

  has unspecified behavior.

-L

  has unspecified behavior.

**Note:** These options have been implemented in **findutils-4.2.9**, but this version of the utilities is not in widespread use as of April 2005. However, future versions of this specification will require support for these arguments.

#### zcat

##### Name

zcat — uncompress files to standard output

##### Description

The **zcat** utility shall behave as described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), with differences listed below.

The [Filesystem Hierarchy Standard](#ID_STD_46_FHS) requires that if **zcat** exists, it must be a symbolic or hard link to /bin/gzip. This specification additionally allows **zcat** to be a wrapper script which calls **gzip -c -d**.

##### Differences

The **zcat** utility shall write to standard output the uncompressed form of files that have been compressed using any of the compression methods supported by the **gzip** utility. It is the equivalent of **gzip -c -d**. Input files are not affected.

# **VII Execution Environment**

# **18 File System Hierarchy**

An LSB conforming implementation shall provide the mandatory portions of the file system hierarchy specified in the [Filesystem Hierarchy Standard](#ID_STD_46_FHS) (FHS), together with any additional requirements made in this specification.

An LSB conforming application shall conform to the [Filesystem Hierarchy Standard](#ID_STD_46_FHS).

The FHS allows many components or subsystems to be optional. An application shall check for the existence of an optional component before using it, and should behave in a reasonable manner if the optional component is not present.

The FHS requirement to locate the operating system kernel in either / or /boot does not apply if the operating system kernel does not exist as a file in the file system.

The FHS specifies certain behaviors for a variety of commands if they are present (for example, **ping** or **python**). However, conforming applications shall not rely on any commands beyond those required by this specification. The mere existence of a command may not be used as an indication that the command behaves in any particular way.

The following directories or links need not be present: /etc/X11 /usr/bin/X11 /usr/lib/X11 /proc

## **18.1 /dev: Device Files**

The devices described in Chapter 6. "Operating System Specific Annex", Section 6.1. "Linux", subsection 6.1.3. "/dev: Devices and special files" in the [Filesystem Hierarchy Standard](#ID_STD_46_FHS) are required on an LSB conforming system. Other devices may also exist in /dev. Device names may exist as symbolic links to other device nodes located in /dev or subdirectories of /dev. There is no requirement concerning major/minor number values.

## **18.2 /etc: Host-specific system configuration**

In addition to the requirements for /etc in the [Filesystem Hierarchy Standard](#ID_STD_46_FHS), an LSB conforming system shall also provide the following directories or symbolic links to directories:

/etc/cron.d

  A directory containing extended **crontab** files; see [Cron Jobs](#ID_CRONJOBS).

/etc/cron.daily

  A directory containing shell scripts to be executed once a day; see [Cron Jobs](#ID_CRONJOBS).

/etc/cron.hourly

  A directory containing shell scripts to be executed once per hour; see [Cron Jobs](#ID_CRONJOBS).

/etc/cron.monthly

  A directory containing shell scripts to be executed once per month; see [Cron Jobs](#ID_CRONJOBS).

/etc/cron.weekly

  A directory containing shell scripts to be executed once a week; see [Cron Jobs](#ID_CRONJOBS).

/etc/init.d

  A directory containing system initialization scripts; see [Installation and Removal of Init Scripts](#ID_INITSRCINSTRM).

/etc/profile.d

  A directory containing shell scripts. Script names should follow the same conventions as specified for cron jobs (see [Cron Jobs](#ID_CRONJOBS), but should have the suffix .sh. The behavior is unspecified if a script is installed in this directory that does not have the suffix .sh.

The **sh** utility shall read and execute commands in its current execution environment from all the shell scripts in this directory that have the suffix .sh when invoked as an interactive login shell, or if the *-l* (the letter *ell*) is specified (see [Shell Invocation](#ID_CMD_45_SH_45_INVOKE)).

**Future Directions:** These directories are required at this version of the LSB since there is not yet an agreed method for abstracting the implementation so that applications need not be aware of these locations during installation.

### **18.2.1 File Naming Conventions**

Conforming implementations and applications installing files into any of the above locations under /etc may only use filenames from the following managed namespaces:

• Assigned names. Such names must be chosen from the character set [a-z0-9]. In order to avoid conflicts these names shall be registered. This specification establishes a registry of provider, package and script names which is maintained at the Linux Assigned Names and Numbers Authority (LANANA). See www.lanana.org (http://www.lanana.org) to register names or look up already registered names.

**Note:** Commonly used names should be registered to avoid conflicts and promote name reuse across distributions. Project developers are encouraged to reserve names with the LANANA as early as possible as registration is on a first-come, first-served basis.

• Hierarchical names. Script names in this category take the form: <hier1>-<hier2>-...-<name>, where name is taken from the character set [a-z0-9], and where there may be one or more <hier-*n*> components. <hier1> may either be an LSB provider name registered with the LANANA, or it may be a domain name registered to the provider in the DNS system, containing at least one '.' (e.g. "debian.org", "staroffice.sun.com"). The LSB provider name registered with the LANANA shall only consist of the ASCII characters [a-z0-9].

• Reserved names. Names that begin with the character '\_' are reserved for distribution use only. Names in this form should be used for essential system packages only.

**Note:** As this specification cannot enforce rules for applications which do not choose to conform to it, conforming applications need to be aware that the managed namespaces may have been polluted with unregistered filenames and should check for namespace collisions and take appropriate steps if they occur.

In general, if a package or system script is likely to be used on multiple systems, the package developers or the distribution should register the name through the LANANA, and distributions should strive to use the same name whenever possible. For applications which may not be essential or may not be commonly installed, the hierarchical namespace may be more appropriate. An advantage to the hierarchical namespace is that there is no need to consult with the LANANA before using a specific name.

Short names are highly desirable, since system administrators may wish to manually start and stop services. Given this, they should be standardized on a per-package basis. This is the rationale behind having the LANANA organization assign these names. The LANANA may be called upon to handle other namespace issues, such as package/prerequisites naming.

## **18.3 User Accounting Databases**

The [Filesystem Hierarchy Standard](#ID_STD_46_FHS) specifies two optional locations for user accounting databases used by the getutent(), getutent\_r(), getutxent(), getutxid(), getutxline(), and pututxline() functions. These are /var/run/utmp and /var/run/wtmp.

The LSB does not specify the format or structure of these files, or even if they are files at all. They should be used only as "magic cookies" to the utmpname() function.

## **18.4 Path For System Administration Utilities**

Certain utilities used for system administration (and other privileged commands) may be stored in /sbin, /usr/sbin, and /usr/local/sbin. Applications requiring to use commands identified as system administration utilities should add these directories to their PATH. By default, as described in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), standard utilities shall be found on the PATH returned by **getconf PATH** (or **command -p getconf PATH** to be guaranteed to invoke the correct version of **getconf**).

# **19 Additional Recommendations**

## **19.1 Recommendations for applications on ownership and permissions**

### **19.1.1 Directory Write Permissions**

The application should not depend on having directory write permission in any directory except /tmp, /var/tmp, and the invoking user's home directory.

In addition, the application may store variable data in /var/opt/*package*, (where *package* is the name of the application package), if such a directory is created with appropriate permissions during the package installation.

For these directories the application should be able to work with directory write permissions restricted by the S\_ISVTXT bit, implementing the restricted deletion mode as described for the XSI option for [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

### **19.1.2 File Write Permissions**

The application should not depend on file write permission to any file that it does not itself create.

### **19.1.3 File Read and execute Permissions**

The application should not depend on having read permission to every file and directory.

### **19.1.4 SUID and SGID Permissions**

The application should not depend on the set user ID or set group ID (the S\_ISUID or S\_ISGID permission bits) permissions of a file not packaged with the application. Instead, the distribution is responsible for assuming that all system commands have the required permissions and work correctly.

**Rationale:** In order to implement common security policies it is strongly advisable for applications to use the minimum set of security attributes necessary for correct operation. Applications that require substantial appropriate privilege are likely to cause problems with such security policies.

### **19.1.5 Privileged users**

In general, applications should not depend on running as a privileged user. This specification uses the term "appropriate privilege" throughout to identify operations that cannot be achieved without some special granting of additional privilege.

Applications that have a reason to run with appropriate privilege should outline this reason clearly in their documentation. Users of the application should be informed, that "this application demands security privileges, which could interfere with system security".

The application should not contain binary-only software that requires being run with appropriate privilege, as this makes security auditing harder or even impossible.

### **19.1.6 Changing permissions**

The application shall not change permissions of files and directories that do not belong to its own package. Should an application require that certain files and directories not directly belonging to the package have a particular ownership, the application shall document this requirement, and may fail during installation if the permissions on these files is inappropriate.

### **19.1.7 Removable Media (Cdrom, Floppy, etc.)**

Applications that expect to be runnable from removable media should not depend on logging in as a privileged user, and should be prepared to deal with a restrictive environment. Examples of such restrictions could be default mount options that disable set-user/group-ID attributes, disabling block or character-special files on the medium, or remapping the user and group IDs of files away from any privileged value.

**Rationale:** System vendors and local system administrators want to run applications from removable media, but want the possibility to control what the application can do.

### **19.1.8 Installable applications**

Where the installation of an application needs additional privileges, it must clearly document all files and system databases that are modified outside of those in /opt/*pkg-name*, /etc/opt/*pkg-name*, and /var/opt/*pkg-name*, other than those that may be updated by system logging or auditing activities.

Without this, the local system administrator would have to blindly trust a piece of software, particularly with respect to its security.

# **20 Additional Behaviors**

## **20.1 Mandatory Optional Behaviors**

This section specifies behaviors in which there is optional behavior in one of the standards on which this specification relies, and where this specification requires a specific behavior.

**Note:** This specification does not require the kernel to be Linux; the set of mandated options reflects current existing practice, but may be modified in future releases.

LSB conforming implementations shall support the following options defined within the [*POSIX 1003.1-2008 (ISO/IEC 9945-2009)*](#ID_STD_46_SUSV4):

|  |
| --- |
| \_POSIX\_FSYNC |
| \_POSIX\_MAPPED\_FILES |
| \_POSIX\_MEMLOCK |
| \_POSIX\_MEMLOCK\_RANGE |
| \_POSIX\_MEMORY\_PROTECTION |
| \_POSIX\_PRIORITY\_SCHEDULING |
| \_POSIX\_REALTIME\_SIGNALS |
| \_POSIX\_THREAD\_ATTR\_STACKADDR |
| \_POSIX\_THREAD\_ATTR\_STACKSIZE |
| \_POSIX\_THREAD\_PROCESS\_SHARED |
| \_POSIX\_THREAD\_SAFE\_FUNCTIONS |
| \_POSIX\_THREADS |

The opendir() function shall consume a file descriptor in the same fashion as open(), and therefore may fail with EMFILE or ENFILE.

The START and STOP termios characters shall be changeable, as described as optional behavior in the "General Terminal Interface" section of the [*POSIX 1003.1-2008 (ISO/IEC 9945-2009)*](#ID_STD_46_SUSV4).

The access() function function shall fail with errno set to EINVAL if the *amode* argument contains bits other than those set by the bitwise inclusive OR of R\_OK, W\_OK, X\_OK and F\_OK.

The link() function shall require access to the existing file in order to succeed, as described as optional behavior in the [*POSIX 1003.1-2008 (ISO/IEC 9945-2009)*](#ID_STD_46_SUSV4).

Calling unlink() on a directory shall fail. Calling link() specifying a directory as the first argument shall fail. See also [unlink](#ID_BASELIB_45_UNLINK_45_3).

**Note:** Linux allows rename() on a directory without having write access, but this specification does not require this behavior.

### **20.1.1 Special Requirements**

LSB conforming systems shall enforce certain special additional restrictions above and beyond those required by [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

**Note:** These additional restrictions are required in order to support the testing and certification programs associated with the LSB. In each case, these are values that defined macros must not have; conforming applications that use these values shall trigger a failure in the interface that is otherwise described as a "may fail".

The fcntl() function shall treat the "cmd" value -1 as invalid.

The *whence* value -1 shall be an invalid value for the lseek(), fseek() and fcntl() functions.

The value -5 shall be an invalid signal number.

If the sigaddset() or sigdelset() functions are passed an invalid signal number, they shall return with EINVAL. Implementations are only required to enforce this requirement for signal numbers which are specified to be invalid by this specification (such as the -5 mentioned above).

The mode value -1 to the access() function shall be treated as invalid.

A value of -1 shall be an invalid "\_PC\_..." value for pathconf().

A value of -1 shall be an invalid "\_SC..." value for sysconf().

The *nl\_item* value -1 shall be invalid for nl\_langinfo().

The value -1 shall be an invalid "\_CS\_..." value for confstr().

The value "a" shall be an invalid *mode* argument to popen().

The fcntl() function shall fail and set errno to EDEADLK if the *cmd* argument is F\_SETLKW, and the lock is blocked by a lock from another process already blocked by the current process.

The opendir() function shall consume a file descriptor; the readdir() function shall fail and set errno to EBADF if the underlying file descriptor is closed.

The link() function shall not work across file systems, and shall fail and set errno to EXDEV as described as optional behavior in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4).

## **20.2 Optional Mandatory Behaviors**

This section specifies behaviors that are mandatory in one of the standards on which this specification relies, but which are optional in this specification.

[POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) describes the behavior of the file access time, available as the *st\_atime* field of the stat and stat64 structures. An LSB conforming implementation need not update this information every time a file is accessed.

**Note:** A subsequent edition of the POSIX standard no longer mandates updating of *st\_atime* but the older edition is still the guiding standard for this specification, thus this exception is needed.

## **20.3 Executable Scripts**

An executable script is an executable file of which the first two characters are #! as defined in the portable character set. In [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), this construct is undefined, but reserved for implementations which wish to provide this functionality. LSB conforming implementations shall support executable scripts.

A successful call to a function of the exec family with an executable script as the first parameter shall result in a new process, where the process image started is that of the interpreter. The path name of the interpreter follows the #! characters.

If the executable script has a first line

#! interpreter [arg]

then *interpreter* shall be called with an argument array consisting of an unspecified zeroth argument, followed by *arg* (if present), followed by a path name for the script, followed by the arguments following the zeroth argument in the exec call of the script.

The interpreter shall not perform any operations on the first line of an executable script.

The first line of the executable script shall meet all of the following criteria otherwise the results are unspecified:

1. Is of one of the forms:

#!interpreter

#! interpreter

#!interpreter arg

#! interpreter arg

2. The *interpreter* argument is an absolute pathname of an executable file other than an executable script.

3. Neither the *interpreter* argument nor the *arg* argument, if present, contain any quoting characters.

4. Neither the *interpreter* argument nor the *arg* argument, if present, contain any whitespace characters.

5. The length of the entire line is no longer than 80 bytes.

If the interpreter is required by this specification to be in a specfic named directory, a conforming application must use that path for *interpreter*, as implementations are not prohibited from having other, possibly non-conforming, versions of the same interpreter installed on the system. If the interpreter is a required command in this specification, but does not have a required path, the application should take special measures to insure the appropriate version is selected. If the interpreter is not a required command in this specification, the application must make appropriate provisions that the interpreter is available at the appropriate path.

**Note:** In case the path is not specified, it is recommended that an installation script for executable scripts use the standard PATH returned by a call to the **getconf** command with the argument *PATH*, combined with the **command** command to determine the location of a standard command.

For example to determine the location of the standard **awk** command:

PATH=`getconf PATH` command -v awk

The installation script should ensure that the returned pathname is an absolute pathname prior to use, since a shell builtin might be returned for some utilities.

Use of the common form #!/usr/bin/env interpreter is not recommended as the PATH will be unknown at execution time and an alternative version of *interpreter* might be selected.

# **21 Localization**

## **21.1 Introduction**

In order to install a message catalog, the installation procedure shall supply the message catalog in a format readable by the **msgfmt** command, which shall be invoked to compile the message catalog into an appropriate binary format on the target system.

**Rationale:** The original intent was to allow an application to contain the binary GNU MO format files. However, the format of these files is not officially stable, hence it is necessary to compile these catalogs on the target system. These binary catalogs may differ from architecture to architecture as well.

The resulting binary message catalog shall be located in the package's private area under /opt, and the application may use bindtextdomain() to specify this location.

Implementations shall support the POSIX and C locales as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4). Other locales may be supported.

Implementations may define additional locale categories not defined by that standard.

**Note:** Implementations choosing additional locale categories should be aware of [ISO/IEC TR14652](#ID_STD_46_TR14652) and are advised not to choose names that conflict with that specification. If implementations provide locale categories whose names are part of the FDCC set of [ISO/IEC TR14652](#ID_STD_46_TR14652), they should behave as defined by that specification.

## **21.2 Regular Expressions**

Utilities that process regular expressions shall support Basic Regular Expressions and Extended Regular Expressions as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), with the following exceptions:

Range expression (such as [a-z]) can be based on code point order instead of collating element order.

Equivalence class expression (such as [=a=]) and multi-character collating element expression (such as [.ch.]) are optional.

Handling of a multi-character collating element is optional.

This affects at least the following utilities:

• **awk** (see [awk](#ID_AWK))

• **grep** (see [grep](#ID_GREP)) (including **egrep** , see [egrep](#ID_EGREP))

• **sed** (see [sed](#ID_SED))

It also affects the behavior of interfaces in the base libraries, including at least

• regexec() (see [regexec](#ID_BASELIB_45_REGEXEC_45_2))

## **21.3 Pattern Matching Notation**

Utilities that perform filename pattern matching (also known as Filename Globbing) shall do it as specified in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), Pattern Matching Notation, with the following exceptions:

Pattern bracket expressions (such as [a-z]) can be based on code point order instead of collating element order.

Equivalence class expression (such as [=a=]) and multi-character collating element expression (such as [.ch.]) are optional.

Handling of a multi-character collating element is optional.

This affects at least the following utilities: **cpio** ([cpio](#ID_CPIO)), **find** and **tar** ([tar](#ID_TAR)).

# **VIII System Initialization**

# **22 System Initialization**

## **22.1 Cron Jobs**

In addition to the individual user crontab files specified by [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), which are located in /var/spool/cron as specified by the [Filesystem Hierarchy Standard](#ID_STD_46_FHS) (FHS), the process that executes scheduled commands shall also process the following additional crontab files, which are in a different format (see below). /etc/crontab, /etc/cron.d/\*. The installation of a package shall not modify the crontab file /etc/crontab, and shall not directly modify the user crontab files in /var/spool/cron/crontabs. but may use the **crontab** command to modify the latter.

If a package wishes to install a job that has to be executed periodically, it shall place an executable *cron script* in one of the following directories:

|  |
| --- |
| /etc/cron.hourly |
| /etc/cron.daily |
| /etc/cron.weekly |
| /etc/cron.monthly |

As these directory names suggest, the files within them are executed on a hourly, daily, weekly, or monthly basis, respectively, under the control of an entry in one of the system crontab files, at an unspecified time of day. See below for the rules concerning the names of cron scripts.

**Note:** It is recommended that cron scripts installed in any of these directories be script files rather than compiled binaries so that they may be modified by the local system administrator. Conforming applications may only install cron scripts which use an interpreter required by this specification or provided by this or another conforming application.

This specification does not define the concept of a package *upgrade*. Implementations may do different things when packages are upgraded, including not replacing a cron script if it marked as a configuration file, particularly if the cron script appears to have been modified since installation. In some circumstances, the cron script may not be removed when the package is uninstalled. Applications should design their installation procedure and cron scripts to be robust in the face of such behavior. In particular, cron scripts should not fail obscurely if run in unexpected circumstances. Testing for the existence of application binaries before executing them is suggested.

If a certain task has to be executed at other than the predefined frequencies, the package shall install a file /etc/cron.d/*cron-name*. The file shall have the same format as that described for the **crontab** command in [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4), except that there shall be an additional field, *username*, before the name of the command to execute. For completeness, the seven fields shall be:

1. Minute [0,59]

2. Hour [0,23]

3. Day of the month [1,31]

4. Month of the year [1,12]

5. Day of the week [0,6] (with 0=Sunday)

6. Username

7. command [args ...]

This file shall be processed by the system automatically, with the named command being run at the specified time, as the specified username.

Applications installing files in these directories shall use the LSB naming conventions (see [File Naming Conventions](#ID_FHS_45_NAME_45_RULES)).

## **22.2 Init Script Actions**

Conforming applications which need to execute commands on changes to the system run level (including boot and shutdown), may install one or more *init scripts*. Init scripts provided by conforming applications shall accept a single argument which selects the action:

|  |  |
| --- | --- |
| **start** | start the service |
| **stop** | stop the service |
| **restart** | stop and restart the service if the service is already running, otherwise start the service |
| **try-restart** | restart the service if the service is already running |
| **reload** | cause the configuration of the service to be reloaded without actually stopping and restarting the service |
| **force-reload** | cause the configuration to be reloaded if the service supports this, otherwise restart the service if it is running |
| **status** | print the current status of the service |

The **start**, **stop**, **restart**, **force-reload**, and **status** actions shall be supported by all init scripts; the **reload** and the **try-restart** actions are optional. Other init-script actions may be defined by the init script.

Init scripts shall ensure that they will behave sensibly if invoked with **start** when the service is already running, or with **stop** when not running, and that they do not kill similarly-named user processes. The best way to achieve this is to use the init-script functions provided by /lib/lsb/init-functions (see [Init Script Functions](#ID_INISCRPTFUNC))

If a service reloads its configuration automatically (as in the case of cron, for example), the **reload** action of the init script shall behave as if the configuration was reloaded successfully. The **restart**, **try-restart**, **reload** and **force-reload** actions may be atomic; that is if a service is known not to be operational after a restart or reload, the script may return an error without any further action.

**Note:** This specification does not define the concept of a package *upgrade*. Implementations may do different things when packages are upgraded, including not replacing an init script if it is marked as a configuration file, particularly if the file appears to have been modified since installation. In some circumstances, the init script may not be removed when the package is uninstalled. Applications should design their installation procedure and init scripts to be robust in the face of such behavior. In particular, init scripts should not fail obscurely if run in unexpected circumstances. Testing for the existence of application binaries before executing them is suggested.

If the **status** action is requested, the init script will return the following exit status codes.

|  |  |
| --- | --- |
| 0 | program is running or service is OK |
| 1 | program is dead and /var/run pid file exists |
| 2 | program is dead and /var/lock lock file exists |
| 3 | program is not running |
| 4 | program or service status is unknown |
| 5-99 | reserved for future LSB use |
| 100-149 | reserved for distribution use |
| 150-199 | reserved for application use |
| 200-254 | reserved |

For all other init-script actions, the init script shall return an exit status of zero if the action was successful. Otherwise, the exit status shall be non-zero, as defined below. In addition to straightforward success, the following situations are also to be considered successful:

• restarting a service (instead of reloading it) with the **force-reload** argument

• running **start** on a service already running

• running **stop** on a service already stopped or not running

• running **restart** on a service already stopped or not running

• running **try-restart** on a service already stopped or not running

In case of an error while processing any init-script action except for **status**, the init script shall print an error message and exit with a non-zero status code:

|  |  |
| --- | --- |
| 1 | generic or unspecified error (current practice) |
| 2 | invalid or excess argument(s) |
| 3 | unimplemented feature (for example, "reload") |
| 4 | user had insufficient privilege |
| 5 | program is not installed |
| 6 | program is not configured |
| 7 | program is not running |
| 8-99 | reserved for future LSB use |
| 100-149 | reserved for distribution use |
| 150-199 | reserved for application use |
| 200-254 | reserved |

Error and status messages should be printed with the logging functions (see [Init Script Functions](#ID_INISCRPTFUNC)) log\_success\_msg(), log\_failure\_msg() and log\_warning\_msg(). Scripts may write to standard error or standard output, but implementations need not present text written to standard error/output to the user or do anything else with it.

**Note:** Since init scripts may be run manually by a system administrator with non-standard environment variable values for PATH, USER, LOGNAME, etc., init scripts should not depend on the values of these environment variables. They should set them to some known/default values if they are needed.

## **22.3 Comment Conventions for Init Scripts**

Conforming applications may install one or more init scripts. These init scripts must be activated by invoking the **install\_initd** command. Prior to package removal, the changes applied by **install\_initd** must be undone by invoking **remove\_initd**. See [Installation and Removal of Init Scripts](#ID_INITSRCINSTRM) for more details.

**install\_initd** and **remove\_initd** determine actions to take by decoding a specially formatted block of lines in the script. This block shall be delimited by the lines

### BEGIN INIT INFO

### END INIT INFO

The delimiter lines may contain trailing whitespace, which shall be ignored. All lines inside the block shall begin with a hash character '#' in the first column, so the shell interprets them as comment lines which do not affect operation of the script. The lines shall be of the form:

# {keyword}: arg1 [arg2...]

with exactly one space character between the '#' and the keyword, with a single exception. In lines following a line containing the **Description** keyword, and until the next keyword or block ending delimiter is seen, a line where the '#' is followed by more than one space or a tab character shall be treated as a continuation of the previous line.

The information extracted from the block is used by the installation tool or the init-script system to assure that init scripts are run in the correct order. It is unspecified whether the information is evaluated only when **install\_initd** runs, when the init scripts are executed, or both. The information extracted includes run levels, defined in [Run Levels](#ID_RUNLEVELS), and boot facilities, defined in [Facility Names](#ID_FACILNAME).

The following keywords, with their arguments, are defined:

**Provides:** boot\_facility\_1 [boot\_facility\_2...]

  boot facilities provided by this init script. When an init script is run with a **start** argument, the boot facility or facilities specified by the **Provides** keyword shall be deemed present and hence init scripts which require those boot facilities should be started later. When an init script is run with a **stop** argument, the boot facilities specified by the **Provides** keyword are deemed no longer present.

**Required-Start:** boot\_facility\_1 [boot\_facility\_2...]

  facilities which must be available during startup of this service. The init-script system should insure init scripts which provide the **Required-Start** facilities are started before starting this script.

**Required-Stop:** boot\_facility\_1 [boot\_facility\_2...]

  facilities which must be available during the shutdown of this service. The init-script system should avoid stopping init scripts which provide the **Required-Stop** facilities until this script is stopped.

**Should-Start:** boot\_facility\_1 [boot\_facility\_2...]

  facilities which, if present, should be available during startup of this service. This allows for weak dependencies which do not cause the service to fail if a facility is not available. The service may provide reduced functionality in this situation. Conforming applications should not rely on the existence of this feature.

**Should-Stop:** boot\_facility\_1 [boot\_facility\_2...]

  facilities which should be available during shutdown of this service.

**Default-Start:** run\_level\_1 [run\_level\_2...]

**Default-Stop:** run\_level\_1 [run\_level\_2...]

  which run levels should by default run the init script with a **start** (**stop**) argument to start (stop) the services controlled by the init script.

For example, if a service should run in runlevels 3, 4, and 5 only, specify "Default-Start: 3 4 5" and "Default-Stop: 0 1 2 6".

**Short-Description:** short\_description

  provide a brief description of the actions of the init script. Limited to a single line of text.

**Description:** multiline\_description

  provide a more complete description of the actions of the init script. May span mulitple lines. In a multiline description, each continuation line shall begin with a '#' followed by tab character or a '#' followed by at least two space characters. The multiline description is terminated by the first line that does not match this criteria.

Additional keywords may be defined in future versions of this specification. Also, implementations may define local extensions by using the prefix **X-*implementor***. For example, **X-RedHat-foobardecl**, or **X-Debian-xyzzydecl**.

Example:

### BEGIN INIT INFO

# Provides: lsb-ourdb

# Required-Start: $local\_fs $network $remote\_fs

# Required-Stop: $local\_fs $network $remote\_fs

# Default-Start: 2 3 4 5

# Default-Stop: 0 1 6

# Short-Description: start and stop OurDB

# Description: OurDB is a very fast and reliable database

# engine used for illustrating init scripts

### END INIT INFO

The comment conventions described in this section are only required for init scripts installed by conforming applications. Conforming runtime implementations are not required to use this scheme in their system provided init scripts.

**Note:** This specification does not require, but is designed to allow, the development of a system which runs init scripts in parallel. Hence, enforced-serialization of scripts is avoided unless it is explicitly necessary.

## **22.4 Installation and Removal of Init Scripts**

Conforming applications may install one or more initialization scripts (or *init scripts*). An init script shall be installed in /etc/init.d (which may be a symbolic link to another location), by the package installer.

During the installer's post-install processing phase the program **/usr/lib/lsb/install\_initd** must be called to activate the init script. Activation consists of arranging for the init script to be called in the correct order on system run-level changes (including system boot and shutdown), based on dependencies supplied in the init script (see [Comment Conventions for Init Scripts](#ID_INITSCRCOMCONV)). The **install\_initd** command should be thought of as a wrapper which hides the implementation details; how any given implementation arranges for the init script to be called at the appropriate time is not specified.

Example: if an init script specified "Default-Start: 3 4 5" and "Default-Stop: 0 1 2 6", **install\_initd** might create "start" symbolic links with names starting with 'S' in /etc/rc3.d, /etc/rc4.d and /etc/rc5.d and "stop" symbolic links with names starting with 'K' in /etc/rc0.d, /etc/rc1.d, /etc/rc2.d and /etc/rc6.d. Such a scheme would be similar to the System V Init mechanism, but is by no means the only way this specification could be implemented.

The **install\_initd** command takes a single argument, the full pathname of the installed init script. The init script must already be installed in /etc/init.d. The **install\_initd** command will not copy it there, only activate it once it has been installed. For example:

/usr/lib/lsb/install\_initd /etc/init.d/example.com-coffeed

The **install\_initd** command shall return an exit status of zero if the init-script activation was successful or if the init script was already activated. If the dependencies in the init script (see [Comment Conventions for Init Scripts](#ID_INITSCRCOMCONV)) cannot be met, an exit status of one shall be returned and the init script shall not be activated.

When a software package is removed, **/usr/lib/lsb/remove\_initd** must be called to deactivate the init script. This must occur before the init script itself is removed, as the dependency information in the script may be required for successful completion. Thus the installer's pre-remove processing phase must call **remove\_initd**, and pass the full pathname of the installed init script. The package installer is still responsible for removing the init script. For example:

/usr/lib/lsb/remove\_initd /etc/init.d/example.com-coffeed

The **remove\_initd** program shall return an exit status of zero if the init script has been successfully deactivated or if the init script is not activated. If another init script which depends on a boot facility provided by this init script is activated, an exit status of one shall be returned and the init script shall remain activated. The installer must fail on such an exit code so it does not subsequently remove the init script.

**Note:** This specification does not describe a mechanism for the system administrator to manipulate the run levels at which an init script is started or stopped. There is no assurance that modifying the comment block for this purpose will have the desired effect.

## **22.5 Run Levels**

The following *run levels* are specified for use by the **Default-Start** and **Default-Stop** actions defined in [Comment Conventions for Init Scripts](#ID_INITSCRCOMCONV) as hints to the **install\_initd** command. Conforming implementations are not required to provide these exact run levels or give them the meanings described here, and may map any level described here to a different level which provides the equivalent functionality. Applications may not depend on specific run-level numbers.

|  |  |
| --- | --- |
| 0 | halt |
| 1 | single user mode |
| 2 | multiuser with no network services exported |
| 3 | normal/full multiuser |
| 4 | reserved for local use, default is normal/full multiuser |
| 5 | multiuser with a display manager or equivalent |
| 6 | reboot |

**Note:** These run levels were chosen as reflecting the most frequent existing practice, and in the absence of other considerations, implementors are strongly encouraged to follow this convention to provide consistency for system administrators who need to work with multiple distributions.

## **22.6 Facility Names**

Boot *facilities* are used to indicate dependencies in initialization scripts, as defined in [Comment Conventions for Init Scripts](#ID_INITSCRCOMCONV). Facility names are assigned to scripts by the **Provides:** keyword. Facility names that begin with a dollar sign ('$') are reserved system facility names.

**Note:** Facility names are only recognized in the context of the init script comment block and are not available in the body of the init script. In particular, the use of the leading '$' character does not imply system facility names are subject to shell variable expansion, since they appear inside comments.

Conforming applications shall not provide facilities that begin with a dollar sign. Implementations shall provide the following facility names:

**$local\_fs**

  all local file systems are mounted

**$network**

  basic networking support is available. Example: a server program could listen on a socket.

**$named**

  IP name-to-address translation, using the interfaces described in this specification, are available to the level the system normally provides them. Example: if a DNS query daemon normally provides this facility, then that daemon has been started.

**$portmap**

  daemons providing SunRPC/ONCRPC portmapping service as defined in [RFC 1833: Binding Protocols for ONC RPC Version 2](#ID_STD_46_RFC1833) (if present) are running.

**$remote\_fs**

  all remote file systems are available. In some configurations, file systems such as /usr may be remote. Many applications that require **$local\_fs** will probably also require **$remote\_fs**.

**$syslog**

  system logger is operational.

**$time**

  the system time has been set, for example by using a network-based time program such as **ntp** or **rdate**, or via the hardware Real Time Clock.

Other (non-system) facilities may be defined by other conforming applications. These facilities shall be named using the same conventions defined for naming init scripts (see [Script Names](#ID_SCRPTNAMES)). Commonly, the facility provided by a conforming init script will have the same name as the name assigned to the init script.

## **22.7 Script Names**

Since init scripts live in a single directory, they must share a single namespace. To avoid conflicts, applications installing files in this directories shall use the LSB naming conventions (see [File Naming Conventions](#ID_FHS_45_NAME_45_RULES)).

## **22.8 Init Script Functions**

Each conforming init script shall execute the commands in the file /lib/lsb/init-functions in the current environment (see shell special built-in command **dot**). This file shall cause the following shell script commands to be defined in an unspecified manner.

**Note:** This can be done either by adding a directory to the PATH variable which defines these commands, or by defining shell aliases or functions.

Although the commands made available via this mechanism need not be conforming applications in their own right, applications that use them should only depend on features described in this specification.

Conforming scripts shall not specify the "exit on error" option (i.e. **set -e**) when sourcing this file, or calling any of the commands thus made available.

The **start\_daemon**, **killproc** and **pidofproc** functions shall use the following algorithm for determining the status and the process identifiers of the specified program.

1. If the *-p pidfile* option is specified, and the named pidfile exists, a single line at the start of the pidfile shall be read. If this line contains one or more numeric values, separated by spaces, these values shall be used. If the *-p pidfile* option is specified and the named pidfile does not exist, the functions shall assume that the daemon is not running.

2. Otherwise, /var/run/*basename*.pid shall be read in a similar fashion. If this contains one or more numeric values on the first line, these values shall be used. Optionally, implementations may use unspecified additional methods to locate the process identifiers required.

The method used to determine the status is implementation defined, but should allow for non-binary programs.

**Note:** Commonly used methods check either for the existence of the /proc/pid directory or use /proc/pid/exe and /proc/pid/cmdline. Relying only on /proc/pid/exe is discouraged since this specification does not specify the existence of, or semantics for, /proc. Additionally, using /proc/pid/exe may result in a not-running status for daemons that are written in a script language.

Conforming implementations may use other mechanisms besides those based on pidfiles, unless the *-p pidfile* option has been used. Conforming applications should not rely on such mechanisms and should always use a pidfile. When a program is stopped, it should delete its pidfile. Multiple process identifiers shall be separated by a single space in the pidfile and in the output of **pidofproc**.

**start\_daemon** [-f] [-n nicelevel] [-p pidfile] pathname [args...]

  runs the specified program as a daemon. The **start\_daemon** function shall check if the program is already running using the algorithm given above. If so, it shall not start another copy of the daemon unless the *-f* option is given. The *-n* option specifies a nice level. See **nice**. **start\_daemon** shall return the LSB defined exit status codes. It shall return 0 if the program has been successfully started or is running and not 0 otherwise.

**killproc** [-p pidfile] pathname [signal]

  The **killproc** function shall stop the specified program. The program is found using the algorithm given above. If a signal is specified, using the *-signal\_name* or *-signal\_number* syntaxes as specified by the **kill** command, the program is sent that signal. Otherwise, a SIGTERM followed by a SIGKILL after an unspecified number of seconds shall be sent. If a program has been terminated, the pidfile should be removed if the terminated process has not already done so. The **killproc** function shall return the LSB defined exit status codes. If called without a signal, it shall return 0 if the program has been stopped or is not running and not 0 otherwise. If a signal is given, it shall return 0 only if the program is running.

**pidofproc** [-p pidfile] pathname

  The **pidofproc** function shall return one or more process identifiers for a particular daemon using the algorithm given above. Only process identifiers of running processes should be returned. Multiple process identifiers shall be separated by a single space.

**Note:** A process may exit between **pidofproc** discovering its identity and the caller of **pidofproc** being able to act on that identity. As a result, no test assertion can be made that the process identifiers returned by **pidofproc** *shall* be running processes.

The **pidofproc** function shall return the LSB defined exit status codes for "status". It shall return 0 if the program is running and not 0 otherwise.

**log\_success\_msg** message

  The **log\_success\_msg** function shall cause the system to write a success message to an unspecified log file. The format of the message is unspecified. The **log\_success\_msg** function may also write a message to the standard output.

**Note:** The message should be relatively short; no more than 60 characters is highly desirable.

**log\_failure\_msg** message

  The **log\_failure\_msg** function shall cause the system to write a failure message to an unspecified log file. The format of the message is unspecified. The **log\_failure\_msg** function may also write a message to the standard output.

**Note:** The message should be relatively short; no more than 60 characters is highly desirable.

**log\_warning\_msg** message

  The **log\_warning\_msg** function shall cause the system to write a warning message to an unspecified log file. The format of the message is unspecified. The **log\_warning\_msg** function may also write a message to the standard output.

**Note:** The message should be relatively short; no more than 60 characters is highly desirable.

# **IX Users & Groups**

# **23 Users & Groups**

## **23.1 User and Group Database**

The format of the User and Group databases is not specified. Programs may only read these databases using the provided API. Changes to these databases should be made using the provided commands.

## **23.2 User & Group Names**

[Table 23-1](#ID_TBL_45_REQUIREDUSERS) describes required mnemonic user and group names. This specification makes no attempt to numerically assign user or group identity numbers, with the exception that both the User ID and Group ID for the user root shall be equal to 0.

**Table 23-1 Required User & Group Names**

| **User** | **Group** | **Comments** |
| --- | --- | --- |
| root | root | Administrative user with all appropriate privileges |
| bin | bin | Legacy User ID/Group IDa |
| daemon | daemon | Legacy User ID/Group IDb |
| Notes:  a The bin User ID/Group ID is included for compatibility with legacy applications. New applications should no longer use the bin User ID/Group ID.  b The daemon User ID/Group ID was used as an unprivileged User ID/Group ID for daemons to execute under in order to limit their access to the system. Generally daemons should now run under individual User ID/Group IDs in order to further partition daemons from one another. | | |

[Table 23-2](#ID_TBL_45_OPTUSERS) is a table of optional mnemonic user and group names. This specification makes no attempt to numerically assign uid or gid numbers. If the username exists on a system, then they should be in the suggested corresponding group. These user and group names are for use by distributions, not by applications.

**Table 23-2 Optional User & Group Names**

| **User** | **Group** | **Comments** |
| --- | --- | --- |
| adm | adm | Administrative special privileges |
| lp | lp | Printer special privileges |
| sync | sync | Login to sync the system |
| shutdown | shutdown | Login to shutdown the system |
| halt | halt | Login to halt the system |
| mail | mail | Mail special privileges |
| news | news | News special privileges |
| uucp | uucp | UUCP special privileges |
| operator | root | Operator special privileges |
| man | man | Man special privileges |
| nobody | nobody | Used by NFS |

Only a minimum working set of "user names" and their corresponding "user groups" are required. Applications cannot assume non system user or group names will be defined.

Applications cannot assume any policy for the default file creation mask (**umask**) or the default directory permissions a user may have. Applications should enforce user only file permissions on private files such as mailboxes. The location of the users home directory is also not defined by policy other than the recommendations of the [Filesystem Hierarchy Standard](#ID_STD_46_FHS) and should be obtained by the getpwnam(), getpwnam\_r(), getpwent(), getpwuid(), and getpwuid\_r() functions.

## **23.3 User ID Ranges**

The system User IDs from 0 to 99 should be statically allocated by the system, and shall not be created by applications.

The system User IDs from 100 to 499 should be reserved for dynamic allocation by system administrators and post install scripts using **useradd**.

## **23.4 Rationale**

The purpose of specifying optional users and groups is to reduce the potential for name conflicts between applications and distributions.

# **X Network Security Services**

# **24 Libraries**

## **24.1 Interfaces for libnspr4**

[Table 24-1](#ID_LIB_45_LIBNSPR4_45_DEF) defines the library name and shared object name for the libnspr4 library

**Table 24-1 libnspr4 Definition**

|  |  |
| --- | --- |
| Library: | libnspr4 |
| SONAME: | libnspr4.so |

The behavior of the interfaces in this library is specified by the following specifications:

|  |
| --- |
| [NSPR] [NSPR Reference](#ID_STD_46_NSPR) |

### **24.1.1 Netscape Portable Runtime**

#### 24.1.1.1 Interfaces for Netscape Portable Runtime

An LSB conforming implementation shall provide the generic functions for Netscape Portable Runtime specified in [Table 24-2](#ID_TBL_45_LIBNSPR4_45_NETSC_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 24-2 libnspr4 - Netscape Portable Runtime Function Interfaces**

|  |  |  |
| --- | --- | --- |
| PR\_Abort [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_Accept [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_AtomicAdd [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |
| PR\_AtomicDecrement [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_AtomicIncrement [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_AtomicSet [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |
| PR\_Bind [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_Calloc [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_Cleanup [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |
| PR\_ClearInterrupt [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_Close [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_Connect [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |
| PR\_ConnectContinue [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_ConvertIPv4AddrToIPv6 [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_CreateIOLayerStub [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |
| PR\_CreatePipe [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_DestroyCondVar [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_DestroyLock [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |
| PR\_EnumerateAddrInfo [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_ExplodeTime [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_FormatTime [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |
| PR\_Free [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_FreeAddrInfo [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_GMTParameters [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |
| PR\_GetAddrInfoByName [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_GetCanonNameFromAddrInfo [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_GetDefaultIOMethods [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |
| PR\_GetDescType [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_GetError [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_GetErrorText [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |
| PR\_GetErrorTextLength [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_GetLayersIdentity [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_GetOSError [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |
| PR\_GetSocketOption [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_GetThreadPrivate [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_GetThreadScope [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |
| PR\_GetThreadState [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_GetUniqueIdentity [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_ImplodeTime [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |
| PR\_ImportTCPSocket [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_Init [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_Initialize [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |
| PR\_InitializeNetAddr [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_Initialized [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_Interrupt [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |
| PR\_IntervalNow [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_IntervalToMicroseconds [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_IntervalToMilliseconds [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |
| PR\_IntervalToSeconds [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_Listen [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_LocalTimeParameters [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |
| PR\_Lock [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_Malloc [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_MicrosecondsToInterval [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |
| PR\_MillisecondsToInterval [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_NetAddrToString [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_NewCondVar [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |
| PR\_NewLock [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_NewThreadPrivateIndex [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_NormalizeTime [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |
| PR\_NotifyAllCondVar [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_NotifyCondVar [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_Now [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |
| PR\_Open [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_OpenTCPSocket [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_OpenUDPSocket [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |
| PR\_ParseTimeString [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_ParseTimeStringToExplodedTime [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_Poll [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |
| PR\_PopIOLayer [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_ProcessExit [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_PushIOLayer [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |
| PR\_Read [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_Realloc [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_Recv [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |
| PR\_RecvFrom [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_SecondsToInterval [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_Send [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |
| PR\_SendTo [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_SetError [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_SetErrorText [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |
| PR\_SetSocketOption [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_SetThreadPrivate [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_Shutdown [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |
| PR\_Sleep [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_StringToNetAddr [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_TicksPerSecond [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |
| PR\_Unlock [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_WaitCondVar [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) | PR\_Write [[NSPR]](#ID_REFSTD_46_LIBNSPR4_46_1) |

## **24.2 Data Definitions for libnspr4**

This section defines global identifiers and their values that are associated with interfaces contained in libnspr4. These definitions are organized into groups that correspond to system headers. This convention is used as a convenience for the reader, and does not imply the existence of these headers, or their content. Where an interface is defined as requiring a particular system header file all of the data definitions for that system header file presented here shall be in effect.

This section gives data definitions to promote binary application portability, not to repeat source interface definitions available elsewhere. System providers and application developers should use this ABI to supplement - not to replace - source interface definition specifications.

This specification uses the [ISO C (1999)](#ID_STD_46_ISOC99) C Language as the reference programming language, and data definitions are specified in ISO C format. The C language is used here as a convenient notation. Using a C language description of these data objects does not preclude their use by other programming languages.

### **24.2.1 nspr4/nspr.h**

#define nspr\_h\_\_\_

### **24.2.2 nspr4/plarena.h**

#define plarena\_h\_\_\_

typedef struct PLArenaPool {

struct PLArena first;

struct PLArena \*current;

PRUint32 arenasize;

PRUword mask;

} PLArenaPool;

struct PLArena {

struct PLArena \*next;

PRUword base;

PRUword limit;

PRUword avail;

};

### **24.2.3 nspr4/plhash.h**

#define plhash\_h\_\_\_

typedef PRUint32 PLHashNumber;

typedef PRIntn(\*PLHashComparator) (const void \*, const void \*);

typedef struct PLHashAllocOps {

void \*(\*allocTable) (void \*, PRSize);

void (\*freeTable) (void \*, void \*);

struct PLHashEntry \*(\*allocEntry) (void \*, const void \*);

void (\*freeEntry) (void \*, struct PLHashEntry \*, PRUintn);

} PLHashAllocOps;

typedef PLHashNumber(\*PLHashFunction) (const void \*);

struct PLHashEntry {

struct PLHashEntry \*next;

PLHashNumber keyHash;

const void \*key;

void \*value;

};

struct PLHashTable {

struct PLHashEntry \*\*buckets;

PRUint32 nentries;

PRUint32 shift;

PLHashFunction keyHash;

PLHashComparator keyCompare;

PLHashComparator valueCompare;

const PLHashAllocOps \*allocOps;

void \*allocPriv;

};

### **24.2.4 nspr4/pratom.h**

#define PR\_ATOMIC\_ADD(val) PR\_AtomicAdd(val)

#define PR\_ATOMIC\_DECREMENT(val) PR\_AtomicDecrement(val)

#define PR\_ATOMIC\_INCREMENT(val) PR\_AtomicIncrement(val)

#define PR\_ATOMIC\_SET(val) PR\_AtomicSet(val)

extern PRInt32 PR\_AtomicAdd(PRInt32 \* ptr, PRInt32 val);

extern PRInt32 PR\_AtomicDecrement(PRInt32 \* val);

extern PRInt32 PR\_AtomicIncrement(PRInt32 \* val);

extern PRInt32 PR\_AtomicSet(PRInt32 \* val, PRInt32 newval);

### **24.2.5 nspr4/prclist.h**

#define prclist\_h\_\_\_

typedef struct PRCListStr {

PRCList \*next;

PRCList \*prev;

} PRCList;

### **24.2.6 nspr4/prcvar.h**

typedef struct PRCondVar PRCondVar;

extern void PR\_DestroyCondVar(PRCondVar \* cvar);

extern PRCondVar \*PR\_NewCondVar(PRLock \* lock);

extern PRStatus PR\_NotifyAllCondVar(PRCondVar \* cvar);

extern PRStatus PR\_NotifyCondVar(PRCondVar \* cvar);

extern PRStatus PR\_WaitCondVar(PRCondVar \* cvar, PRIntervalTime timeout);

### **24.2.7 nspr4/prerror.h**

#define prerror\_h\_\_\_

typedef PRInt32 PRErrorCode;

extern PRErrorCode PR\_GetError(void);

extern PRInt32 PR\_GetErrorText(char \*text);

extern PRInt32 PR\_GetErrorTextLength(void);

extern PRInt32 PR\_GetOSError(void);

extern void PR\_SetError(PRErrorCode errorCode, PRInt32 oserr);

extern void PR\_SetErrorText(PRIntn textLength, const char \*text);

### **24.2.8 nspr4/prinit.h**

#define prinit\_h\_\_\_

typedef PRIntn(\*PRPrimordialFn) (PRIntn argc, char \*\*argv);

typedef PRStatus(\*PRCallOnceFN) (void);

typedef PRStatus(\*PRCallOnceWithArgFN) (void \*arg);

extern void PR\_Abort(void);

extern PRStatus PR\_Cleanup(void);

extern void PR\_Init(PRThreadType type, PRThreadPriority priority,

PRUintn maxPTDs);

extern PRIntn PR\_Initialize(PRPrimordialFn prmain, PRIntn argc,

char \*\*argv, PRUintn maxPTDs);

extern PRBool PR\_Initialized(void);

extern void PR\_ProcessExit(PRIntn status);

### **24.2.9 nspr4/prinrval.h**

#define prinrval\_h

#define PR\_INTERVAL\_NO\_WAIT 0UL

#define PR\_INTERVAL\_NO\_TIMEOUT 0xffffffffUL

typedef PRUint32 PRIntervalTime;

extern PRIntervalTime PR\_IntervalNow(void);

extern PRUint32 PR\_IntervalToMicroseconds(PRIntervalTime ticks);

extern PRUint32 PR\_IntervalToMilliseconds(PRIntervalTime ticks);

extern PRUint32 PR\_IntervalToSeconds(PRIntervalTime ticks);

extern PRIntervalTime PR\_MicrosecondsToInterval(PRUint32 micro);

extern PRIntervalTime PR\_MillisecondsToInterval(PRUint32 milli);

extern PRIntervalTime PR\_SecondsToInterval(PRUint32 seconds);

extern PRUint32 PR\_TicksPerSecond(void);

### **24.2.10 nspr4/prio.h**

#define prio\_h\_\_\_

#define PR\_RDONLY 0x01

#define PR\_WRONLY 0x02

#define PR\_RDWR 0x04

#define PR\_CREATE\_FILE 0x08

#define PR\_APPEND 0x10

#define PR\_TRUNCATE 0x20

#define PR\_SYNC 0x40

#define PR\_EXCL 0x80

typedef enum PRDescType {

PR\_DESC\_FILE = 1,

PR\_DESC\_SOCKET\_TCP = 2,

PR\_DESC\_SOCKET\_UDP = 3,

PR\_DESC\_LAYERED = 4,

PR\_DESC\_PIPE = 5

} PRDescType;

typedef struct PRIPv6Addr {

union {

PRUint8 \_S6\_u8[15];

PRUint16 \_S6\_u16[7];

PRUint32 \_S6\_u32[3];

PRUint64 \_S6\_u64[1];

} \_S6\_un;

} PRIPv6Addr;

typedef enum PRTransmitFileFlags {

PR\_TRANSMITFILE\_KEEP\_OPEN = 0,

PR\_TRANSMITFILE\_CLOSE\_SOCKET = 1

} PRTransmitFileFlags;

typedef struct PRLinger {

PRBool polarity;

PRIntervalTime linger;

} PRLinger;

typedef struct PRFilePrivate PRFilePrivate;

typedef struct PRFileDesc {

const struct PRIOMethods \*methods;

PRFilePrivate \*secret;

PRFileDesc \*lower;

PRFileDesc \*higher;

void (\*dtor) (PRFileDesc \*);

PRDescIdentity identity;

} PRFileDesc;

typedef union PRNetAddr {

struct {

PRUint16 family;

char data[14];

} raw;

struct {

PRUint16 family;

PRUint16 port;

PRUint32 ip;

char pad[7];

} inet;

struct {

PRUint16 family;

PRUint16 port;

PRUint32 flowinfo;

PRIPv6Addr ip;

PRUint32 scope\_id;

} ipv6;

struct {

PRUint16 family;

char path[103];

} local;

} PRNetAddr;

typedef struct PRMcastRequest {

union PRNetAddr mcaddr;

union PRNetAddr ifaddr;

} PRMcastRequest;

typedef struct PRIOVec {

char \*iov\_base;

int iov\_len;

} PRIOVec;

typedef struct PRSocketOptionData {

PRSockOption option;

union {

PRUintn ip\_ttl;

PRUintn mcast\_ttl;

PRUintn tos;

PRBool non\_blocking;

PRBool reuse\_addr;

PRBool keep\_alive;

PRBool mcast\_loopback;

PRBool no\_delay;

PRBool broadcast;

PRSize max\_segment;

PRSize recv\_buffer\_size;

PRSize send\_buffer\_size;

PRLinger linger;

PRMcastRequest add\_member;

PRMcastRequest drop\_member;

union PRNetAddr mcast\_if;

} value;

} PRSocketOptionData;

typedef PRStatus(\*PRFsyncFN) (PRFileDesc \* fd);

typedef PRStatus(\*PRListenFN) (PRFileDesc \* fd, PRIntn how);

typedef enum PRSeekWhence {

PR\_SEEK\_SET = 0,

PR\_SEEK\_CUR = 1,

PR\_SEEK\_END = 2

} PRSeekWhence;

typedef PRInt32(\*PRAcceptreadFN) (PRFileDesc \* sd, PRFileDesc \* \*nd,

PRNetAddr \* \*raddr, void \*buf,

PRInt32 amount, PRIntervalTime t);

typedef PRStatus(\*PRCloseFN) (PRFileDesc \* fd);

typedef PRInt32(\*PRTransmitfileFN) (PRFileDesc \* sd, PRFileDesc \* fd,

const void \*headers, PRInt32 hlen,

PRTransmitFileFlags flags,

PRIntervalTime t);

typedef enum PRSockOption {

PR\_SockOpt\_Nonblocking,

PR\_SockOpt\_Linger = 1,

PR\_SockOpt\_Reuseaddr = 2,

PR\_SockOpt\_Keepalive = 3,

PR\_SockOpt\_RecvBufferSize = 4,

PR\_SockOpt\_SendBufferSize = 5,

PR\_SockOpt\_IpTimeToLive = 6,

PR\_SockOpt\_IpTypeOfService = 7,

PR\_SockOpt\_AddMember = 8,

PR\_SockOpt\_DropMember = 9,

PR\_SockOpt\_McastInterface = 10,

PR\_SockOpt\_McastTimeToLive = 11,

PR\_SockOpt\_McastLoopback = 12,

PR\_SockOpt\_NoDelay = 13,

PR\_SockOpt\_MaxSegment = 14,

PR\_SockOpt\_Broadcast = 15,

PR\_SockOpt\_Last = 16

} PRSockOption;

typedef PRFileDesc \*(\*PRAcceptFN) (PRFileDesc \* fd, PRNetAddr \* addr,

PRIntervalTime timeout);

typedef PRStatus(\*PRConnectcontinueFN) (PRFileDesc \* fd,

PRInt16 out\_flags);

typedef PRInt32(\*PRReadFN) (PRFileDesc \* fd, void \*buf, PRInt32 amount);

typedef struct PRFileInfo64 {

PRFileType type;

PROffset64 size;

PRTime creationTime;

PRTime modifyTime;

} PRFileInfo64;

typedef PRStatus(\*PRGetsocketoptionFN) (PRFileDesc \* fd,

PRSocketOptionData \* data);

typedef PRInt32(\*PRSendtoFN) (PRFileDesc \* fd, const void \*buf,

PRInt32 amount, PRIntn flags,

const PRNetAddr \* addr,

PRIntervalTime timeout);

typedef PRStatus(\*PRGetsocknameFN) (PRFileDesc \* fd, PRNetAddr \* addr);

typedef PRInt32(\*PRSendFN) (PRFileDesc \* fd, const void \*buf,

PRInt32 amount, PRIntn flags,

PRIntervalTime timeout);

typedef PROffset32(\*PRSeekFN) (PRFileDesc \* fd, PROffset32 offset,

PRSeekWhence how);

typedef PRInt64(\*PRAvailable64FN) (PRFileDesc \* fd);

typedef PRInt32(\*PRAvailableFN) (PRFileDesc \* fd);

typedef struct PRFileInfo {

PRFileType type;

PROffset32 size;

PRTime creationTime;

PRTime modifyTime;

} PRFileInfo;

typedef PROffset64(\*PRSeek64FN) (PRFileDesc \* fd, PROffset64 offset,

PRSeekWhence how);

typedef PRStatus(\*PRSetsocketoptionFN) (PRFileDesc \* fd,

const PRSocketOptionData \* data);

typedef PRInt32(\*PRRecvFN) (PRFileDesc \* fd, void \*buf, PRInt32 amount,

PRIntn flags, PRIntervalTime timeout);

typedef struct PRSendFileData {

PRFileDesc \*fd;

PRUint32 file\_offset;

PRSize file\_nbytes;

const void \*header;

PRInt32 hlen;

const void \*trailer;

PRInt32 tlen;

} PRSendFileData;

typedef PRIntn PRDescIdentity;

typedef PRStatus(\*PRConnectFN) (PRFileDesc \* fd, const PRNetAddr \* addr,

PRIntervalTime timeout);

typedef PRInt32(\*PRSendfileFN) (PRFileDesc \* networkSocket,

PRSendFileData \* sendData,

PRTransmitFileFlags flags,

PRIntervalTime timeout);

typedef PRInt32(\*PRRecvfromFN) (PRFileDesc \* fd, void \*buf, PRInt32 amount,

PRIntn flags, PRNetAddr \* addr,

PRIntervalTime timeout);

typedef struct PRPollDesc {

PRFileDesc \*fd;

PRInt16 in\_flags;

PRInt16 out\_flags;

} PRPollDesc;

typedef PRInt32(\*PRWriteFN) (PRFileDesc \* fd, const void \*buf,

PRInt32 amount);

typedef PRStatus(\*PRFileInfo64FN) (PRFileDesc \* fd, PRFileInfo64 \* info);

typedef PRStatus(\*PRShutdownFN) (PRFileDesc \* fd, PRIntn how);

typedef PRIntn(\*PRReservedFN) (PRFileDesc \* fd);

typedef PRStatus(\*PRFileInfoFN) (PRFileDesc \* fd, PRFileInfo \* info);

typedef PRInt32(\*PRWritevFN) (PRFileDesc \* fd, const PRIOVec \* iov,

PRInt32 iov\_size, PRIntervalTime timeout);

typedef enum PRFileType {

PR\_FILE\_FILE = 1,

PR\_FILE\_DIRECTORY = 2,

PR\_FILE\_OTHER = 3

} PRFileType;

typedef PRStatus(\*PRBindFN) (PRFileDesc \* fd, const PRNetAddr \* addr);

typedef PRInt16(\*PRPollFN) (PRFileDesc \* fd, PRInt16 in\_flags,

PRInt16 \* out\_flags);

struct PRIOMethods {

PRDescType file\_type;

PRCloseFN close;

PRReadFN read;

PRWriteFN write;

PRAvailableFN available;

PRAvailable64FN available64;

PRFsyncFN fsync;

PRSeekFN seek;

PRSeek64FN seek64;

PRFileInfoFN fileInfo;

PRFileInfo64FN fileInfo64;

PRWritevFN writev;

PRConnectFN connect;

PRAcceptFN accept;

PRBindFN bind;

PRListenFN listen;

PRShutdownFN shutdown;

PRRecvFN recv;

PRSendFN send;

PRRecvfromFN recvfrom;

PRSendtoFN sendto;

PRPollFN poll;

PRAcceptreadFN acceptread;

PRTransmitfileFN transmitfile;

PRGetsocknameFN getsockname;

PRGetpeernameFN getpeername;

PRReservedFN reserved\_fn\_6;

PRReservedFN reserved\_fn\_5;

PRGetsocketoptionFN getsocketoption;

PRSetsocketoptionFN setsocketoption;

PRSendfileFN sendfile;

PRConnectcontinueFN connectcontinue;

PRReservedFN reserved\_fn\_3;

PRReservedFN reserved\_fn\_2;

PRReservedFN reserved\_fn\_1;

PRReservedFN reserved\_fn\_0;

};

typedef PRStatus(\*PRGetpeernameFN) (PRFileDesc \* fd, PRNetAddr \* addr);

typedef enum PRShutdownHow {

PR\_SHUTDOWN\_RCV = 0,

PR\_SHUTDOWN\_SEND = 1,

PR\_SHUTDOWN\_BOTH = 2

} PRShutdownHow;

extern PRFileDesc \*PR\_Accept(PRFileDesc \* fd, PRNetAddr \* addr,

PRIntervalTime timeout);

extern PRStatus PR\_Bind(PRFileDesc \* fd, const PRNetAddr \* addr);

extern PRStatus PR\_Close(PRFileDesc \* fd);

extern PRStatus PR\_Connect(PRFileDesc \* fd, const PRNetAddr \* addr,

PRIntervalTime timeout);

extern PRStatus PR\_ConnectContinue(PRFileDesc \* fd, PRInt16 out\_flags);

extern PRFileDesc \*PR\_CreateIOLayerStub(PRDescIdentity ident,

const struct PRIOMethods \*methods);

extern PRStatus PR\_CreatePipe(PRFileDesc \* \*readPipe,

PRFileDesc \* \*writePipe);

extern const struct PRIOMethods \*PR\_GetDefaultIOMethods(void);

extern PRDescType PR\_GetDescType(PRFileDesc \* file);

extern PRDescIdentity PR\_GetLayersIdentity(PRFileDesc \* fd);

extern PRStatus PR\_GetSocketOption(PRFileDesc \* fd,

PRSocketOptionData \* data);

extern PRDescIdentity PR\_GetUniqueIdentity(const char \*layer\_name);

extern PRStatus PR\_Listen(PRFileDesc \* fd, PRIntn backlog);

extern PRFileDesc \*PR\_Open(const char \*name, PRIntn flags, PRIntn mode);

extern PRFileDesc \*PR\_OpenTCPSocket(PRIntn af);

extern PRFileDesc \*PR\_OpenUDPSocket(PRIntn af);

extern PRInt32 PR\_Poll(PRPollDesc \* pds, PRIntn npds,

PRIntervalTime timeout);

extern PRFileDesc \*PR\_PopIOLayer(PRFileDesc \* fd\_stack, PRDescIdentity id);

extern PRStatus PR\_PushIOLayer(PRFileDesc \* fd\_stack, PRDescIdentity id,

PRFileDesc \* layer);

extern PRInt32 PR\_Read(PRFileDesc \* fd, void \*buf, PRInt32 amount);

extern PRInt32 PR\_Recv(PRFileDesc \* fd, void \*buf, PRInt32 amount,

PRIntn flags, PRIntervalTime timeout);

extern PRInt32 PR\_RecvFrom(PRFileDesc \* fd, void \*buf, PRInt32 amount,

PRIntn flags, PRNetAddr \* addr,

PRIntervalTime timeout);

extern PRInt32 PR\_Send(PRFileDesc \* fd, const void \*buf, PRInt32 amount,

PRIntn flags, PRIntervalTime timeout);

extern PRInt32 PR\_SendTo(PRFileDesc \* fd, const void \*buf, PRInt32 amount,

PRIntn flags, const PRNetAddr \* addr,

PRIntervalTime timeout);

extern PRStatus PR\_SetSocketOption(PRFileDesc \* fd,

const PRSocketOptionData \* data);

extern PRStatus PR\_Shutdown(PRFileDesc \* fd, PRShutdownHow how);

extern PRInt32 PR\_Write(PRFileDesc \* fd, const void \*buf, PRInt32 amount);

### **24.2.11 nspr4/private/pprio.h**

#define pprio\_h\_\_\_

typedef PRInt32 PROsfd;

extern PRFileDesc \*PR\_ImportTCPSocket(PROsfd osfd);

### **24.2.12 nspr4/prlock.h**

#define prlock\_h\_\_\_

typedef struct PRLock PRLock;

extern void PR\_DestroyLock(PRLock \* lock);

extern void PR\_Lock(PRLock \* lock);

extern PRLock \*PR\_NewLock(void);

extern PRStatus PR\_Unlock(PRLock \* lock);

### **24.2.13 nspr4/prmem.h**

#define PR\_NEW(\_struct) ((\_struct \*) PR\_MALLOC(sizeof(\_struct)))

#define PR\_NEWZAP(\_struct) ((\_struct\*)PR\_Calloc(1, sizeof(\_struct)))

#define PR\_CALLOC(\_size) (PR\_Calloc(1, (\_size)))

#define PR\_MALLOC(\_bytes) (PR\_Malloc(\_bytes))

#define PR\_REALLOC(\_ptr, \_size) (PR\_Realloc((\_ptr), (\_size)))

#define PR\_FREEIF(\_ptr) if (\_ptr) PR\_DELETE(\_ptr)

#define PR\_DELETE(\_ptr) { PR\_Free(\_ptr); (\_ptr) = NULL; }

extern void \*PR\_Calloc(PRUint32 nelem, PRUint32 elsize);

extern void PR\_Free(void \*ptr);

extern void \*PR\_Malloc(PRUint32 size);

extern void \*PR\_Realloc(void \*ptr, PRUint32 size);

### **24.2.14 nspr4/prmon.h**

#define prmon\_h\_\_\_

typedef struct PRMonitor PRMonitor;

### **24.2.15 nspr4/prnetdb.h**

#define prnetdb\_h\_\_\_

#define PR\_NetAddrFamily(addr) ((addr)->raw.family)

#define PR\_NetAddrInetPort(addr) \

((addr)->raw.family == PR\_AF\_INET6 ? (addr)->ipv6.port : (addr)->inet.port)

typedef struct PRHostEnt {

char \*h\_name;

char \*\*h\_aliases;

PRInt32 h\_addrtype;

PRInt32 h\_length;

char \*\*h\_addr\_list;

} PRHostEnt;

typedef struct PRAddrInfo PRAddrInfo;

typedef enum PRNetAddrValue {

PR\_IpAddrNull,

PR\_IpAddrAny,

PR\_IpAddrLoopback,

PR\_IpAddrV4Mapped

} PRNetAddrValue;

extern void PR\_ConvertIPv4AddrToIPv6(PRUint32 v4addr, PRIPv6Addr \* v6addr);

extern void \*PR\_EnumerateAddrInfo(void \*enumPtr,

const PRAddrInfo \* addrInfo,

PRUint16 port, PRNetAddr \* result);

extern void PR\_FreeAddrInfo(PRAddrInfo \* addrInfo);

extern PRAddrInfo \*PR\_GetAddrInfoByName(const char \*hostname, PRUint16 af,

PRIntn flags);

extern const char \*PR\_GetCanonNameFromAddrInfo(const PRAddrInfo \*

addrInfo);

extern PRStatus PR\_InitializeNetAddr(PRNetAddrValue val, PRUint16 port,

PRNetAddr \* addr);

extern PRStatus PR\_NetAddrToString(const PRNetAddr \* addr, char \*string,

PRUint32 size);

extern PRStatus PR\_StringToNetAddr(const char \*string, PRNetAddr \* addr);

### **24.2.16 nspr4/prthread.h**

#define prthread\_h\_\_\_

typedef struct PRThread PRThread;

typedef enum PRThreadType {

PR\_USER\_THREAD,

PR\_SYSTEM\_THREAD

} PRThreadType;

typedef enum PRThreadScope {

PR\_LOCAL\_THREAD,

PR\_GLOBAL\_THREAD,

PR\_GLOBAL\_BOUND\_THREAD

} PRThreadScope;

typedef enum PRThreadState {

PR\_JOINABLE\_THREAD,

PR\_UNJOINABLE\_THREAD

} PRThreadState;

typedef enum PRThreadPriority {

PR\_PRIORITY\_FIRST = 0,

PR\_PRIORITY\_LOW = 0,

PR\_PRIORITY\_NORMAL = 1,

PR\_PRIORITY\_HIGH = 2,

PR\_PRIORITY\_URGENT = 3,

PR\_PRIORITY\_LAST = 3

} PRThreadPriority;

typedef void (\*PRThreadPrivateDTOR) (void \*priv);

extern PRStatus PR\_ClearInterrupt(void);

extern void \*PR\_GetThreadPrivate(PRUintn tpdIndex);

extern PRThreadScope PR\_GetThreadScope(const PRThread \* thread);

extern PRThreadState PR\_GetThreadState(const PRThread \* thread);

extern PRStatus PR\_Interrupt(PRThread \* thread);

extern PRStatus PR\_NewThreadPrivateIndex(PRUintn \* newIndex,

PRThreadPrivateDTOR destructor);

extern PRStatus PR\_SetThreadPrivate(PRUintn tpIndex, void \*priv);

extern PRStatus PR\_Sleep(PRIntervalTime ticks);

### **24.2.17 nspr4/prtime.h**

#define prtime\_h\_\_\_

typedef PRInt64 PRTime;

typedef struct PRTimeParameters {

PRInt32 tp\_gmt\_offset;

PRInt32 tp\_dst\_offset;

} PRTimeParameters;

typedef struct PRExplodedTime {

PRInt32 tm\_usec;

PRInt32 tm\_sec;

PRInt32 tm\_min;

PRInt32 tm\_hour;

PRInt32 tm\_mday;

PRInt32 tm\_month;

PRInt16 tm\_year;

PRInt8 tm\_wday;

PRInt16 tm\_yday;

PRTimeParameters tm\_params;

} PRExplodedTime;

typedef PRTimeParameters(\*PRTimeParamFn) (const PRExplodedTime \* gmt);

extern void PR\_ExplodeTime(PRTime usecs, PRTimeParamFn params,

PRExplodedTime \* expoded);

extern PRUint32 PR\_FormatTime(char \*buf, int buflen, const char \*fmt,

const PRExplodedTime \* tm);

extern PRTimeParameters PR\_GMTParameters(const PRExplodedTime \* gmt);

extern PRTime PR\_ImplodeTime(const PRExplodedTime \* exploded);

extern PRTimeParameters PR\_LocalTimeParameters(const PRExplodedTime \* gmt);

extern void PR\_NormalizeTime(PRExplodedTime \* exploded,

PRTimeParamFn params);

extern PRTime PR\_Now(void);

extern PRStatus PR\_ParseTimeString(const char \*string,

PRBool default\_to\_gmt, PRTime \* result);

extern PRStatus PR\_ParseTimeStringToExplodedTime(const char \*string,

PRBool default\_to\_gmt,

PRExplodedTime \* result);

### **24.2.18 nspr4/prtypes.h**

#define prtypes\_h\_\_\_

typedef int PRInt32;

typedef unsigned long int PRUword;

typedef int PRIntn;

typedef unsigned long int PRUint64;

typedef unsigned char PRUint8;

typedef short int PRInt16;

typedef long int PRInt64;

typedef PRIntn PRBool;

typedef unsigned short PRUint16;

typedef unsigned int PRUint32;

typedef size\_t PRSize;

typedef unsigned int PRUintn;

typedef PRInt64 PROffset64;

typedef PRInt32 PROffset32;

typedef enum {

PR\_FAILURE = -1,

PR\_SUCCESS = 0

} PRStatus;

typedef signed char PRInt8;

## **24.3 Interfaces for libnss3**

[Table 24-3](#ID_LIB_45_LIBNSS3_45_DEF) defines the library name and shared object name for the libnss3 library

**Table 24-3 libnss3 Definition**

|  |  |
| --- | --- |
| Library: | libnss3 |
| SONAME: | libnss3.so |

The behavior of the interfaces in this library is specified by the following specifications:

|  |
| --- |
| [NSS SSL] [Mozilla's NSS SSL Reference](#ID_STD_46_NSS_46_SSL) |

### **24.3.1 NSS Utility**

#### 24.3.1.1 Interfaces for NSS Utility

An LSB conforming implementation shall provide the generic functions for NSS Utility specified in [Table 24-4](#ID_TBL_45_LIBNSS3_45_NSS_45_U_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 24-4 libnss3 - NSS Utility Function Interfaces**

|  |  |  |
| --- | --- | --- |
| CERT\_CheckCertValidTimes(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBNSS3_46_1) | CERT\_DestroyCertificate(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBNSS3_46_1) | CERT\_DupCertificate(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBNSS3_46_1) |
| CERT\_FreeNicknames(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBNSS3_46_1) | CERT\_GetCertNicknames(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBNSS3_46_1) | CERT\_GetDefaultCertDB(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBNSS3_46_1) |
| CERT\_VerifyCertName(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBNSS3_46_1) | CERT\_VerifyCertNow(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBNSS3_46_1) | NSS\_Init(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBNSS3_46_1) |
| NSS\_InitReadWrite(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBNSS3_46_1) | NSS\_NoDB\_Init(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBNSS3_46_1) | NSS\_Shutdown(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBNSS3_46_1) |
| PK11\_FindCertFromNickname(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBNSS3_46_1) | PK11\_FindKeyByAnyCert(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBNSS3_46_1) | PK11\_GetSlotName(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBNSS3_46_1) |
| PK11\_GetTokenName(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBNSS3_46_1) | PK11\_IsHW(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBNSS3_46_1) | PK11\_IsPresent(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBNSS3_46_1) |
| PK11\_IsReadOnly(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBNSS3_46_1) | PK11\_SetPasswordFunc(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBNSS3_46_1) | SECKEY\_DestroyPrivateKey(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBNSS3_46_1) |

## **24.4 Data Definitions for libnss3**

This section defines global identifiers and their values that are associated with interfaces contained in libnss3. These definitions are organized into groups that correspond to system headers. This convention is used as a convenience for the reader, and does not imply the existence of these headers, or their content. Where an interface is defined as requiring a particular system header file all of the data definitions for that system header file presented here shall be in effect.

This section gives data definitions to promote binary application portability, not to repeat source interface definitions available elsewhere. System providers and application developers should use this ABI to supplement - not to replace - source interface definition specifications.

This specification uses the [ISO C (1999)](#ID_STD_46_ISOC99) C Language as the reference programming language, and data definitions are specified in ISO C format. The C language is used here as a convenient notation. Using a C language description of these data objects does not preclude their use by other programming languages.

### **24.4.1 nss3/blapit.h**

#define \_BLAPIT\_H\_

#define PQG\_PBITS\_TO\_INDEX(bits) \

(((bits) < 512 || (bits) > 1024 || (bits) % 64) ? -1 : \

(int)((bits)-512)/64)

#define PQG\_INDEX\_TO\_PBITS(j) \

(((unsigned)(j) > 8) ? -1 : (512 + 64 \* (j)))

#define NSS\_AES 0

#define NSS\_DES 0

#define NSS\_RC2 0

#define NSS\_RC5 0

#define NSS\_AES\_CBC 1

#define NSS\_DES\_CBC 1

#define NSS\_RC2\_CBC 1

#define NSS\_RC5\_CBC 1

#define DSA\_MAX\_P\_BITS 1024

#define DH\_MIN\_P\_BITS 128

#define RSA\_MIN\_MODULUS\_BITS 128

#define AES\_BLOCK\_SIZE 16

#define DSA\_Q\_BITS 160

#define NSS\_DES\_EDE3 2

#define DSA\_SUBPRIME\_LEN 20

#define NSS\_FREEBL\_DEFAULT\_CHUNKSIZE 2048

#define DH\_MAX\_P\_BITS 2236

#define NSS\_DES\_EDE3\_CBC 3

#define DSA\_SIGNATURE\_LEN 40

#define DSA\_MIN\_P\_BITS 512

#define AES\_KEY\_WRAP\_BLOCK\_SIZE 8

#define AES\_KEY\_WRAP\_IV\_BYTES 8

#define DES\_KEY\_LENGTH 8

typedef struct PQGParamsStr {

PLArenaPool \*arena;

SECItem prime;

SECItem subPrime;

SECItem base;

} PQGParams;

typedef struct PQGVerifyStr {

PLArenaPool \*arena;

unsigned int counter;

SECItem seed;

SECItem h;

} PQGVerify;

### **24.4.2 nss3/cert.h**

#define \_CERT\_H\_

extern SECCertTimeValidity CERT\_CheckCertValidTimes(CERTCertificate \* cert,

PRTime t,

PRBool allowOverride);

extern void CERT\_DestroyCertificate(CERTCertificate \* cert);

extern CERTCertificate \*CERT\_DupCertificate(CERTCertificate \* c);

extern void CERT\_FreeNicknames(CERTCertNicknames \* nicknames);

extern CERTCertNicknames \*CERT\_GetCertNicknames(CERTCertDBHandle \* handle,

int what, void \*wincx);

extern CERTCertDBHandle \*CERT\_GetDefaultCertDB(void);

extern SECStatus CERT\_VerifyCertName(CERTCertificate \* cert,

const char \*hostname);

extern SECStatus CERT\_VerifyCertNow(CERTCertDBHandle \* handle,

CERTCertificate \* cert,

PRBool checkSig,

SECCertUsage certUsage, void \*wincx);

### **24.4.3 nss3/certt.h**

#define \_CERTT\_H\_

#define NS\_CERT\_TYPE\_CA \

( NS\_CERT\_TYPE\_SSL\_CA | NS\_CERT\_TYPE\_EMAIL\_CA | \

NS\_CERT\_TYPE\_OBJECT\_SIGNING\_CA | EXT\_KEY\_USAGE\_STATUS\_RESPONDER )

#define NS\_CERT\_TYPE\_APP \

( NS\_CERT\_TYPE\_SSL\_CLIENT | NS\_CERT\_TYPE\_SSL\_SERVER | \

NS\_CERT\_TYPE\_EMAIL | NS\_CERT\_TYPE\_OBJECT\_SIGNING )

#define SEC\_GET\_TRUST\_FLAGS(trust,type) \

(((type)==trustSSL)?((trust)->sslFlags): \

(((type)==trustEmail)?((trust)->emailFlags): \

(((type)==trustObjectSigning)?((trust)->objectSigningFlags):0)))

#define KU\_ALL \

(KU\_DIGITAL\_SIGNATURE | KU\_NON\_REPUDIATION | KU\_KEY\_ENCIPHERMENT | \

KU\_DATA\_ENCIPHERMENT | KU\_KEY\_AGREEMENT | KU\_KEY\_CERT\_SIGN | \

KU\_CRL\_SIGN)

#define CERT\_LIST\_END(n,l) (((void \*)n) == ((void \*)&l->list))

#define CERT\_LIST\_NEXT(n) ((CERTCertListNode \*)n->links.next)

#define CERT\_LIST\_HEAD(l) ((CERTCertListNode \*)PR\_LIST\_HEAD(&l->list))

#define certificateUsageSSLClient (0x0001)

#define certificateUsageSSLServer (0x0002)

#define certificateUsageSSLServerWithStepUp (0x0004)

#define certificateUsageSSLCA (0x0008)

#define certificateUsageEmailSigner (0x0010)

#define certificateUsageEmailRecipient (0x0020)

#define certificateUsageObjectSigner (0x0040)

#define certificateUsageUserCertImport (0x0080)

#define NS\_CERT\_TYPE\_OBJECT\_SIGNING\_CA (0x01)

#define certificateUsageVerifyCA (0x0100)

#define KU\_CRL\_SIGN (0x02)

#define NS\_CERT\_TYPE\_EMAIL\_CA (0x02)

#define RF\_CERTIFICATE\_HOLD (0x02)

#define certificateUsageProtectedObjectSigner (0x0200)

#define KU\_KEY\_CERT\_SIGN (0x04)

#define NS\_CERT\_TYPE\_SSL\_CA (0x04)

#define RF\_CESSATION\_OF\_OPERATION (0x04)

#define certificateUsageStatusResponder (0x0400)

#define KU\_KEY\_AGREEMENT (0x08)

#define NS\_CERT\_TYPE\_RESERVED (0x08)

#define RF\_SUPERSEDED (0x08)

#define certificateUsageAnyCA (0x0800)

#define KU\_DATA\_ENCIPHERMENT (0x10)

#define NS\_CERT\_TYPE\_OBJECT\_SIGNING (0x10)

#define RF\_AFFILIATION\_CHANGED (0x10)

#define KU\_KEY\_ENCIPHERMENT (0x20)

#define NS\_CERT\_TYPE\_EMAIL (0x20)

#define RF\_CA\_COMPROMISE (0x20)

#define KU\_NON\_REPUDIATION (0x40)

#define NS\_CERT\_TYPE\_SSL\_SERVER (0x40)

#define RF\_KEY\_COMPROMISE (0x40)

#define EXT\_KEY\_USAGE\_STATUS\_RESPONDER (0x4000)

#define KU\_KEY\_AGREEMENT\_OR\_ENCIPHERMENT (0x4000)

#define KU\_DIGITAL\_SIGNATURE (0x80)

#define NS\_CERT\_TYPE\_SSL\_CLIENT (0x80)

#define RF\_UNUSED (0x80)

#define EXT\_KEY\_USAGE\_TIME\_STAMP (0x8000)

#define KU\_NS\_GOVT\_APPROVED (0x8000)

#define CERT\_UNLIMITED\_PATH\_CONSTRAINT -2

#define SEC\_CERTIFICATE\_REQUEST\_VERSION 0

#define SEC\_CERTIFICATE\_VERSION\_1 0

#define SEC\_CRL\_VERSION\_1 0

#define SEC\_CERTIFICATE\_VERSION\_2 1

#define SEC\_CERT\_CLASS\_CA 1

#define SEC\_CERT\_NICKNAMES\_ALL 1

#define SEC\_CRL\_VERSION\_2 1

#define SEC\_CERTIFICATE\_VERSION\_3 2

#define SEC\_CERT\_CLASS\_SERVER 2

#define SEC\_CERT\_NICKNAMES\_USER 2

#define CERT\_MAX\_CERT\_CHAIN 20

#define SEC\_CERT\_CLASS\_USER 3

#define SEC\_CERT\_NICKNAMES\_SERVER 3

#define SEC\_CERT\_CLASS\_EMAIL 4

#define SEC\_CERT\_NICKNAMES\_CA 4

#define certificateUsageHighest certificateUsageAnyCA

#define CERT\_LIST\_EMPTY(l) CERT\_LIST\_END(CERT\_LIST\_HEAD(l), l)

typedef struct CERTAVAStr {

SECItem type;

SECItem value;

} CERTAVA;

typedef struct CERTAttributeStr {

SECItem attrType;

SECItem \*\*attrValue;

} CERTAttribute;

typedef struct CERTAuthInfoAccessStr {

SECItem method;

SECItem derLocation;

CERTGeneralName \*location;

} CERTAuthInfoAccess;

typedef struct CERTAuthKeyIDStr {

SECItem keyID;

CERTGeneralName \*authCertIssuer;

SECItem authCertSerialNumber;

SECItem \*\*DERAuthCertIssuer;

} CERTAuthKeyID;

typedef struct CERTBasicConstraintsStr {

PRBool isCA;

int pathLenConstraint;

} CERTBasicConstraints;

typedef struct NSSTrustDomainStr CERTCertDBHandle;

typedef struct CERTCertExtensionStr {

SECItem id;

SECItem critical;

SECItem value;

} CERTCertExtension;

typedef struct CERTCertListStr {

PRCList list;

PLArenaPool \*arena;

} CERTCertList;

typedef struct CERTCertListNodeStr {

PRCList links;

CERTCertificate \*cert;

void \*appData;

} CERTCertListNode;

typedef struct CERTCertNicknamesStr {

PLArenaPool \*arena;

void \*head;

int numnicknames;

char \*\*nicknames;

int what;

int totallen;

} CERTCertNicknames;

typedef struct CERTCertTrustStr {

unsigned int sslFlags;

unsigned int emailFlags;

unsigned int objectSigningFlags;

} CERTCertTrust;

typedef struct CERTSignedDataStr {

SECItem data;

SECAlgorithmID signatureAlgorithm;

SECItem signature;

} CERTSignedData;

typedef struct CERTCertificateListStr {

SECItem \*certs;

int len;

PLArenaPool \*arena;

} CERTCertificateList;

typedef struct CERTNameStr {

PLArenaPool \*arena;

CERTRDN \*\*rdns;

} CERTName;

typedef struct CERTCrlStr {

PLArenaPool \*arena;

SECItem version;

SECAlgorithmID signatureAlg;

SECItem derName;

CERTName name;

SECItem lastUpdate;

SECItem nextUpdate;

CERTCrlEntry \*\*entries;

CERTCertExtension \*\*extensions;

} CERTCrl;

typedef struct CERTCrlDistributionPointsStr {

CRLDistributionPoint \*\*distPoints;

} CERTCrlDistributionPoints;

typedef struct CERTCrlEntryStr {

SECItem serialNumber;

SECItem revocationDate;

CERTCertExtension \*\*extensions;

} CERTCrlEntry;

typedef struct CERTCrlHeadNodeStr {

PLArenaPool \*arena;

CERTCertDBHandle \*dbhandle;

CERTCrlNode \*first;

CERTCrlNode \*last;

} CERTCrlHeadNode;

typedef struct CERTCrlNodeStr {

CERTCrlNode \*next;

int type;

CERTSignedCrl \*crl;

} CERTCrlNode;

typedef struct CERTDistNamesStr {

PLArenaPool \*arena;

int nnames;

SECItem \*names;

void \*head;

} CERTDistNames;

typedef struct OtherNameStr {

SECItem name;

SECItem oid;

} OtherName;

typedef struct CERTGeneralNameListStr {

PLArenaPool \*arena;

CERTGeneralName \*name;

int refCount;

int len;

PRLock \*lock;

} CERTGeneralNameList;

typedef struct CERTIssuerAndSNStr {

SECItem derIssuer;

CERTName issuer;

SECItem serialNumber;

} CERTIssuerAndSN;

typedef struct CERTSubjectPublicKeyInfoStr {

PLArenaPool \*arena;

SECAlgorithmID algorithm;

SECItem subjectPublicKey;

} CERTSubjectPublicKeyInfo;

typedef struct CERTGeneralNameStr {

CERTGeneralNameType type;

union {

CERTName directoryName;

OtherName OthName;

SECItem other;

} name;

SECItem derDirectoryName;

PRCList l;

} CERTGeneralName;

typedef struct CERTNameConstraintsStr {

CERTNameConstraint \*permited;

CERTNameConstraint \*excluded;

SECItem \*\*DERPermited;

SECItem \*\*DERExcluded;

} CERTNameConstraints;

typedef struct CERTOKDomainNameStr {

CERTOKDomainName \*next;

char name[1];

} CERTOKDomainName;

typedef struct CERTPrivKeyUsagePeriodStr {

SECItem notBefore;

SECItem notAfter;

PLArenaPool \*arena;

} CERTPrivKeyUsagePeriod;

typedef struct CERTRDNStr {

CERTAVA \*\*avas;

} CERTRDN;

typedef struct CERTSignedCrlStr {

PLArenaPool \*arena;

CERTCrl crl;

void \*reserved1;

PRBool reserved2;

PRBool isperm;

PRBool istemp;

int referenceCount;

CERTCertDBHandle \*dbhandle;

CERTSignedData signatureWrap;

char \*url;

SECItem \*derCrl;

PK11SlotInfo \*slot;

CK\_OBJECT\_HANDLE pkcs11ID;

void \*opaque;

} CERTSignedCrl;

typedef struct CERTValidityStr {

PLArenaPool \*arena;

SECItem notBefore;

SECItem notAfter;

} CERTValidity;

typedef struct CERTStatusConfigStr {

CERTStatusChecker statusChecker;

CERTStatusDestroy statusDestroy;

void \*statusContext;

} CERTStatusConfig;

typedef struct CERTSubjectListStr {

PLArenaPool \*arena;

int ncerts;

char \*emailAddr;

CERTSubjectNode \*head;

CERTSubjectNode \*tail;

void \*entry;

} CERTSubjectList;

typedef struct CERTSubjectNodeStr {

struct CERTSubjectNodeStr \*next;

struct CERTSubjectNodeStr \*prev;

SECItem certKey;

SECItem keyID;

} CERTSubjectNode;

typedef struct CERTCertificateRequestStr {

PLArenaPool \*arena;

SECItem version;

CERTName subject;

CERTSubjectPublicKeyInfo subjectPublicKeyInfo;

CERTAttribute \*\*attributes;

} CERTCertificateRequest;

typedef struct CERTCertificateStr {

PLArenaPool \*arena;

char \*subjectName;

char \*issuerName;

CERTSignedData signatureWrap;

SECItem derCert;

SECItem derIssuer;

SECItem derSubject;

SECItem derPublicKey;

SECItem certKey;

SECItem version;

SECItem serialNumber;

SECAlgorithmID signature;

CERTName issuer;

CERTValidity validity;

CERTName subject;

CERTSubjectPublicKeyInfo subjectPublicKeyInfo;

SECItem issuerID;

SECItem subjectID;

CERTCertExtension \*\*extensions;

char \*emailAddr;

CERTCertDBHandle \*dbhandle;

SECItem subjectKeyID;

PRBool keyIDGenerated;

unsigned int keyUsage;

unsigned int rawKeyUsage;

PRBool keyUsagePresent;

PRUint32 nsCertType;

PRBool keepSession;

PRBool timeOK;

CERTOKDomainName \*domainOK;

PRBool isperm;

PRBool istemp;

char \*nickname;

char \*dbnickname;

struct NSSCertificateStr \*nssCertificate;

CERTCertTrust \*trust;

int referenceCount;

CERTSubjectList \*subjectList;

CERTAuthKeyID \*authKeyID;

PRBool isRoot;

union {

void \*apointer;

struct {

unsigned int hasUnsupportedCriticalExt;

} bits;

} options;

int series;

PK11SlotInfo \*slot;

CK\_OBJECT\_HANDLE pkcs11ID;

PRBool ownSlot;

} CERTCertificate;

typedef struct CERTVerifyLogStr {

PLArenaPool \*arena;

unsigned int count;

struct CERTVerifyLogNodeStr \*head;

struct CERTVerifyLogNodeStr \*tail;

} CERTVerifyLog;

typedef struct CRLDistributionPointStr {

DistributionPointTypes distPointType;

union {

CERTGeneralName \*fullName;

CERTRDN relativeName;

} distPoint;

SECItem reasons;

CERTGeneralName \*crlIssuer;

SECItem derDistPoint;

SECItem derRelativeName;

SECItem \*\*derCrlIssuer;

SECItem \*\*derFullName;

SECItem bitsmap;

} CRLDistributionPoint;

typedef enum SECCertUsageEnum {

certUsageSSLClient,

certUsageSSLServer = 1,

certUsageSSLServerWithStepUp = 2,

certUsageSSLCA = 3,

certUsageEmailSigner = 4,

certUsageEmailRecipient = 5,

certUsageObjectSigner = 6,

certUsageUserCertImport = 7,

certUsageVerifyCA = 8,

certUsageProtectedObjectSigner = 9,

certUsageStatusResponder = 10,

certUsageAnyCA = 11

} SECCertUsage;

typedef PRInt64 SECCertificateUsage;

typedef enum SECCertTimeValidityEnum {

secCertTimeValid,

secCertTimeExpired = 1,

secCertTimeNotValidYet = 2,

secCertTimeUndetermined = 3

} SECCertTimeValidity;

typedef enum CERTCompareValidityStatusEnum {

certValidityUndetermined,

certValidityChooseB = 1,

certValidityEqual = 2,

certValidityChooseA = 3

} CERTCompareValidityStatus;

typedef enum CERTGeneralNameTypeEnum {

certOtherName = 1,

certRFC822Name = 2,

certDNSName = 3,

certX400Address = 4,

certDirectoryName = 5,

certEDIPartyName = 6,

certURI = 7,

certIPAddress = 8,

certRegisterID = 9

} CERTGeneralNameType;

typedef struct CERTNameConstraintStr {

CERTGeneralName name;

SECItem DERName;

SECItem min;

SECItem max;

PRCList l;

} CERTNameConstraint;

typedef enum DistributionPointTypesEnum {

generalName = 1,

relativeDistinguishedName = 2

} DistributionPointTypes;

struct CERTVerifyLogNodeStr {

CERTCertificate \*cert;

long int error;

unsigned int depth;

void \*arg;

struct CERTVerifyLogNodeStr \*next;

struct CERTVerifyLogNodeStr \*prev;

};

typedef SECStatus(\*CERTStatusChecker) (CERTCertDBHandle \*,

CERTCertificate \*, PRInt64, void \*);

typedef SECStatus(\*CERTStatusDestroy) (CERTStatusConfig \*);

typedef struct {

SECOidTag oid;

SECItem qualifierID;

SECItem qualifierValue;

} CERTPolicyQualifier;

typedef struct {

SECOidTag oid;

SECItem policyID;

CERTPolicyQualifier \*\*policyQualifiers;

} CERTPolicyInfo;

typedef struct {

PLArenaPool \*arena;

CERTPolicyInfo \*\*policyInfos;

} CERTCertificatePolicies;

typedef struct {

SECItem organization;

SECItem \*\*noticeNumbers;

} CERTNoticeReference;

typedef struct {

PLArenaPool \*arena;

CERTNoticeReference noticeReference;

SECItem derNoticeReference;

SECItem displayText;

} CERTUserNotice;

typedef struct {

PLArenaPool \*arena;

SECItem \*\*oids;

} CERTOidSequence;

### **24.4.4 nss3/cmsreclist.h**

#define \_CMSRECLIST\_H

typedef struct NSSCMSRecipientStr {

int riIndex;

int subIndex;

enum {

RLIssuerSN,

RLSubjKeyID = 1

} kind;

union {

CERTIssuerAndSN \*issuerAndSN;

SECItem \*subjectKeyID;

} id;

CERTCertificate \*cert;

SECKEYPrivateKey \*privkey;

PK11SlotInfo \*slot;

} NSSCMSRecipient;

### **24.4.5 nss3/cryptoht.h**

#define \_CRYPTOHT\_H\_

typedef struct SGNContextStr SGNContext;

typedef struct VFYContextStr VFYContext;

### **24.4.6 nss3/hasht.h**

#define \_HASHT\_H\_

#define MD2\_LENGTH 16

#define MD5\_LENGTH 16

#define SHA1\_LENGTH 20

#define SHA256\_LENGTH 32

#define SHA384\_LENGTH 48

#define SHA512\_LENGTH 64

#define HASH\_LENGTH\_MAX SHA512\_LENGTH

typedef struct SECHashObjectStr {

unsigned int length;

void \*(\*create) (void);

void \*(\*clone) (void \*);

void (\*destroy) (void \*, PRBool);

void (\*begin) (void \*);

void (\*update) (void \*, const unsigned char \*, unsigned int);

void (\*end) (void \*, unsigned char \*, unsigned int \*, unsigned int);

unsigned int blocklength;

HASH\_HashType type;

} SECHashObject;

typedef struct HASHContextStr {

const struct SECHashObjectStr \*hashobj;

void \*hash\_context;

} HASHContext;

typedef enum {

HASH\_AlgNULL,

HASH\_AlgMD2 = 1,

HASH\_AlgMD5 = 2,

HASH\_AlgSHA1 = 3,

HASH\_AlgSHA256 = 4,

HASH\_AlgSHA384 = 5,

HASH\_AlgSHA512 = 6,

HASH\_AlgTOTAL = 7

} HASH\_HashType;

### **24.4.7 nss3/key.h**

#define \_KEY\_H\_

### **24.4.8 nss3/keyhi.h**

#define \_KEYHI\_H\_

extern void SECKEY\_DestroyPrivateKey(SECKEYPrivateKey \* key);

### **24.4.9 nss3/keyt.h**

#define \_KEYT\_H\_

### **24.4.10 nss3/keythi.h**

#define \_KEYTHI\_H\_

typedef enum {

nullKey,

rsaKey = 1,

dsaKey = 2,

fortezzaKey = 3,

dhKey = 4,

keaKey = 5,

ecKey = 6

} KeyType;

typedef struct SECKEYRSAPublicKeyStr {

PLArenaPool \*arena;

SECItem modulus;

SECItem publicExponent;

} SECKEYRSAPublicKey;

typedef struct SECKEYPQGParamsStr {

PLArenaPool \*arena;

SECItem prime;

SECItem subPrime;

SECItem base;

} SECKEYPQGParams;

typedef struct SECKEYDSAPublicKeyStr {

SECKEYPQGParams params;

SECItem publicValue;

} SECKEYDSAPublicKey;

typedef struct SECKEYDHParamsStr {

PLArenaPool \*arena;

SECItem prime;

SECItem base;

} SECKEYDHParams;

typedef struct SECKEYDHPublicKeyStr {

PLArenaPool \*arena;

SECItem prime;

SECItem base;

SECItem publicValue;

} SECKEYDHPublicKey;

typedef SECItem SECKEYECParams;

typedef struct SECKEYECPublicKeyStr {

SECKEYECParams DEREncodedParams;

int size;

SECItem publicValue;

} SECKEYECPublicKey;

typedef struct SECKEYFortezzaPublicKeyStr {

int KEAversion;

int DSSversion;

unsigned char KMID[8];

SECItem clearance;

SECItem KEApriviledge;

SECItem DSSpriviledge;

SECItem KEAKey;

SECItem DSSKey;

SECKEYPQGParams params;

SECKEYPQGParams keaParams;

} SECKEYFortezzaPublicKey;

typedef struct SECKEYKEAParamsStr {

PLArenaPool \*arena;

SECItem hash;

} SECKEYKEAParams;

typedef struct SECKEYKEAPublicKeyStr {

SECKEYKEAParams params;

SECItem publicValue;

} SECKEYKEAPublicKey;

typedef struct SECKEYPublicKeyStr {

PLArenaPool \*arena;

KeyType keyType;

PK11SlotInfo \*pkcs11Slot;

CK\_OBJECT\_HANDLE pkcs11ID;

union {

SECKEYRSAPublicKey rsa;

SECKEYDSAPublicKey dsa;

SECKEYDHPublicKey dh;

SECKEYKEAPublicKey kea;

SECKEYFortezzaPublicKey fortezza;

SECKEYECPublicKey ec;

} u;

} SECKEYPublicKey;

typedef struct SECKEYPrivateKeyStr {

PLArenaPool \*arena;

KeyType keyType;

PK11SlotInfo \*pkcs11Slot;

CK\_OBJECT\_HANDLE pkcs11ID;

PRBool pkcs11IsTemp;

void \*wincx;

PRUint32 staticflags;

} SECKEYPrivateKey;

typedef struct {

PRCList links;

SECKEYPrivateKey \*key;

} SECKEYPrivateKeyListNode;

typedef struct {

PRCList list;

PLArenaPool \*arena;

} SECKEYPrivateKeyList;

typedef struct {

PRCList list;

PLArenaPool \*arena;

} SECKEYPublicKeyList;

### **24.4.11 nss3/nss.h**

#define \_\_nss\_h\_

#define NSS\_VERSION "3.11.4"

#define NSS\_INIT\_READONLY 0x1

#define NSS\_INIT\_NOROOTINIT 0x10

#define NSS\_INIT\_NOPK11FINALIZE 0x100

#define NSS\_INIT\_NOCERTDB 0x2

#define NSS\_INIT\_OPTIMIZESPACE 0x20

#define NSS\_INIT\_RESERVED 0x200

#define NSS\_INIT\_NOMODDB 0x4

#define NSS\_INIT\_PK11THREADSAFE 0x40

#define NSS\_INIT\_FORCEOPEN 0x8

#define NSS\_INIT\_PK11RELOAD 0x80

#define NSS\_VMINOR 11

#define NSS\_VMAJOR 3

#define NSS\_VPATCH 4

#define NSS\_INIT\_COOPERATE NSS\_INIT\_PK11THREADSAFE | NSS\_INIT\_PK11RELOAD | NSS\_INIT\_NOPK11FINALIZE | NSS\_INIT\_RESERVED

#define SECMOD\_DB "secmod.db"

extern SECStatus NSS\_Init(const char \*configdir);

extern SECStatus NSS\_InitReadWrite(const char \*configdir);

extern SECStatus NSS\_NoDB\_Init(const char \*configdir);

extern SECStatus NSS\_Shutdown(void);

### **24.4.12 nss3/nssb64.h**

#define \_NSSB64\_H\_

### **24.4.13 nss3/nssb64t.h**

#define \_NSSB64T\_H\_

typedef struct NSSBase64DecoderStr NSSBase64Decoder;

typedef struct NSSBase64EncoderStr NSSBase64Encoder;

### **24.4.14 nss3/nssilckt.h**

#define \_NSSILCKT\_H\_

typedef enum {

nssILockArena,

nssILockSession = 1,

nssILockObject = 2,

nssILockRefLock = 3,

nssILockCert = 4,

nssILockCertDB = 5,

nssILockDBM = 6,

nssILockCache = 7,

nssILockSSL = 8,

nssILockList = 9,

nssILockSlot = 10,

nssILockFreelist = 11,

nssILockOID = 12,

nssILockAttribute = 13,

nssILockPK11cxt = 14,

nssILockRWLock = 15,

nssILockOther = 16,

nssILockSelfServ = 17,

nssILockKeyDB = 18,

nssILockLast = 19

} nssILockType;

### **24.4.15 nss3/nssrwlkt.h**

#define nssrwlkt\_h\_\_\_

typedef struct nssRWLockStr NSSRWLock;

### **24.4.16 nss3/ocspt.h**

#define \_OCSPT\_H\_

typedef struct CERTOCSPRequestStr CERTOCSPRequest;

typedef struct CERTOCSPResponseStr CERTOCSPResponse;

typedef struct CERTOCSPCertIDStr CERTOCSPCertID;

typedef struct CERTOCSPSingleResponseStr CERTOCSPSingleResponse;

### **24.4.17 nss3/pk11pub.h**

#define \_PK11PUB\_H\_

extern CERTCertificate \*PK11\_FindCertFromNickname(const char \*nickname,

void \*wincx);

extern SECKEYPrivateKey \*PK11\_FindKeyByAnyCert(CERTCertificate \* cert,

void \*wincx);

extern char \*PK11\_GetSlotName(PK11SlotInfo \* slot);

extern char \*PK11\_GetTokenName(PK11SlotInfo \* slot);

extern PRBool PK11\_IsHW(PK11SlotInfo \* slot);

extern PRBool PK11\_IsPresent(PK11SlotInfo \* slot);

extern PRBool PK11\_IsReadOnly(PK11SlotInfo \* slot);

extern void PK11\_SetPasswordFunc(PK11PasswordFunc func);

### **24.4.18 nss3/pkcs11t.h**

#define \_PKCS11T\_H\_

typedef unsigned char CK\_BYTE;

typedef CK\_BYTE CK\_CHAR;

typedef CK\_BYTE CK\_UTF8CHAR;

typedef unsigned long int CK\_ULONG;

typedef CK\_ULONG CK\_FLAGS;

typedef void \*CK\_VOID\_PTR;

typedef struct CK\_VERSION {

CK\_BYTE major;

CK\_BYTE minor;

} CK\_VERSION;

typedef struct CK\_INFO {

CK\_VERSION cryptokiVersion;

CK\_UTF8CHAR manufacturerID[31];

CK\_FLAGS flags;

CK\_UTF8CHAR libraryDescription[31];

CK\_VERSION libraryVersion;

} CK\_INFO;

typedef CK\_ULONG CK\_SLOT\_ID;

typedef struct CK\_SLOT\_INFO {

CK\_UTF8CHAR slotDescription[63];

CK\_UTF8CHAR manufacturerID[31];

CK\_FLAGS flags;

CK\_VERSION hardwareVersion;

CK\_VERSION firmwareVersion;

} CK\_SLOT\_INFO;

typedef struct CK\_TOKEN\_INFO {

CK\_UTF8CHAR label[31];

CK\_UTF8CHAR manufacturerID[31];

CK\_UTF8CHAR model[15];

CK\_CHAR serialNumber[15];

CK\_FLAGS flags;

CK\_ULONG ulMaxSessionCount;

CK\_ULONG ulSessionCount;

CK\_ULONG ulMaxRwSessionCount;

CK\_ULONG ulRwSessionCount;

CK\_ULONG ulMaxPinLen;

CK\_ULONG ulMinPinLen;

CK\_ULONG ulTotalPublicMemory;

CK\_ULONG ulFreePublicMemory;

CK\_ULONG ulTotalPrivateMemory;

CK\_ULONG ulFreePrivateMemory;

CK\_VERSION hardwareVersion;

CK\_VERSION firmwareVersion;

CK\_CHAR utcTime[15];

} CK\_TOKEN\_INFO;

typedef CK\_ULONG CK\_SESSION\_HANDLE;

typedef CK\_ULONG CK\_OBJECT\_HANDLE;

typedef CK\_ULONG CK\_OBJECT\_CLASS;

typedef CK\_ULONG CK\_KEY\_TYPE;

typedef CK\_ULONG CK\_ATTRIBUTE\_TYPE;

typedef struct CK\_ATTRIBUTE {

CK\_ATTRIBUTE\_TYPE type;

CK\_VOID\_PTR pValue;

CK\_ULONG ulValueLen;

} CK\_ATTRIBUTE;

typedef CK\_ATTRIBUTE \*CK\_ATTRIBUTE\_PTR;

typedef CK\_ULONG CK\_MECHANISM\_TYPE;

typedef struct CK\_MECHANISM {

CK\_MECHANISM\_TYPE mechanism;

CK\_VOID\_PTR pParameter;

CK\_ULONG ulParameterLen;

} CK\_MECHANISM;

typedef CK\_MECHANISM \*CK\_MECHANISM\_PTR;

typedef CK\_ULONG CK\_RV;

### **24.4.19 nss3/pkcs7t.h**

#define \_PKCS7T\_H\_

typedef struct SEC\_PKCS7RecipientInfoStr {

SECItem version;

CERTIssuerAndSN \*issuerAndSN;

SECAlgorithmID keyEncAlg;

SECItem encKey;

CERTCertificate \*cert;

} SEC\_PKCS7RecipientInfo;

### **24.4.20 nss3/secasn1t.h**

#define \_SECASN1T\_H\_

typedef struct sec\_ASN1Template\_struct {

unsigned long int kind;

unsigned long int offset;

const void \*sub;

unsigned int size;

} SEC\_ASN1Template;

typedef struct sec\_DecoderContext\_struct SEC\_ASN1DecoderContext;

typedef struct sec\_EncoderContext\_struct SEC\_ASN1EncoderContext;

typedef enum {

SEC\_ASN1\_Identifier,

SEC\_ASN1\_Length = 1,

SEC\_ASN1\_Contents = 2,

SEC\_ASN1\_EndOfContents = 3

} SEC\_ASN1EncodingPart;

typedef void (\*SEC\_ASN1NotifyProc) (void \*, PRBool, void \*, int);

typedef void (\*SEC\_ASN1WriteProc) (void \*, const char \*, unsigned long int,

int, SEC\_ASN1EncodingPart);

### **24.4.21 nss3/seccomon.h**

#define \_SECCOMMON\_H\_

typedef enum {

siBuffer,

siClearDataBuffer = 1,

siCipherDataBuffer = 2,

siDERCertBuffer = 3,

siEncodedCertBuffer = 4,

siDERNameBuffer = 5,

siEncodedNameBuffer = 6,

siAsciiNameString = 7,

siAsciiString = 8,

siDEROID = 9,

siUnsignedInteger = 10,

siUTCTime = 11,

siGeneralizedTime = 12,

siVisibleString = 13,

siUTF8String = 14,

siBMPString = 15

} SECItemType;

typedef struct SECItemStr {

SECItemType type;

unsigned char \*data;

unsigned int len;

} SECItem;

typedef enum \_SECStatus {

SECWouldBlock = -2,

SECFailure = -1,

SECSuccess

} SECStatus;

typedef enum \_SECComparison {

SECLessThan = -1,

SECEqual,

SECGreaterThan = 1

} SECComparison;

### **24.4.22 nss3/secdert.h**

#define \_SECDERT\_H\_

typedef struct DERTemplateStr {

unsigned long int kind;

unsigned int offset;

DERTemplate \*sub;

unsigned long int arg;

} DERTemplate;

### **24.4.23 nss3/secdigt.h**

#define \_SECDIGT\_H\_

typedef struct SGNDigestInfoStr {

PLArenaPool \*arena;

SECAlgorithmID digestAlgorithm;

SECItem digest;

} SGNDigestInfo;

### **24.4.24 nss3/secmodt.h**

#define \_SECMODT\_H\_

#define SECMOD\_MAKE\_NSS\_FLAGS(fips,slot) \

"Flags=internal,critical"fips" \

slotparams=("#slot"={"SECMOD\_SLOT\_FLAGS"})"

#define SECMOD\_FIPS\_NAME "NSS Internal FIPS PKCS #11 Module"

#define SECMOD\_INT\_NAME "NSS Internal PKCS #11 Module"

#define SECMOD\_SLOT\_FLAGS "slotFlags=[RSA,DSA,DH,RC2,RC4,DES,RANDOM,SHA1,MD5,MD2,SSL,TLS,AES,SHA256,SHA512]"

#define SECMOD\_EXTERNAL 0

#define CRL\_IMPORT\_DEFAULT\_OPTIONS 0x00000000

#define CRL\_IMPORT\_BYPASS\_CHECKS 0x00000001

#define PK11\_ATTR\_TOKEN 0x00000001L

#define SECMOD\_RSA\_FLAG 0x00000001L

#define PK11\_ATTR\_SESSION 0x00000002L

#define SECMOD\_DSA\_FLAG 0x00000002L

#define PK11\_ATTR\_PRIVATE 0x00000004L

#define SECMOD\_RC2\_FLAG 0x00000004L

#define PK11\_ATTR\_PUBLIC 0x00000008L

#define SECMOD\_RC4\_FLAG 0x00000008L

#define PK11\_ATTR\_MODIFIABLE 0x00000010L

#define SECMOD\_DES\_FLAG 0x00000010L

#define PK11\_ATTR\_UNMODIFIABLE 0x00000020L

#define SECMOD\_DH\_FLAG 0x00000020L

#define PK11\_ATTR\_SENSITIVE 0x00000040L

#define SECMOD\_FORTEZZA\_FLAG 0x00000040L

#define PK11\_ATTR\_INSENSITIVE 0x00000080L

#define SECMOD\_RC5\_FLAG 0x00000080L

#define PK11\_ATTR\_EXTRACTABLE 0x00000100L

#define SECMOD\_SHA1\_FLAG 0x00000100L

#define PK11\_ATTR\_UNEXTRACTABLE 0x00000200L

#define SECMOD\_MD5\_FLAG 0x00000200L

#define SECMOD\_MD2\_FLAG 0x00000400L

#define SECMOD\_SSL\_FLAG 0x00000800L

#define SECMOD\_TLS\_FLAG 0x00001000L

#define SECMOD\_AES\_FLAG 0x00002000L

#define SECMOD\_SHA256\_FLAG 0x00004000L

#define SECMOD\_SHA512\_FLAG 0x00008000L

#define SECMOD\_END\_WAIT 0x01

#define SECMOD\_WAIT\_SIMULATED\_EVENT 0x02

#define SECMOD\_WAIT\_PKCS11\_EVENT 0x04

#define SECMOD\_RESERVED\_FLAG 0X08000000L

#define SECMOD\_FRIENDLY\_FLAG 0x10000000L

#define PK11\_OWN\_PW\_DEFAULTS 0x20000000L

#define PK11\_DISABLE\_FLAG 0x40000000L

#define SECMOD\_RANDOM\_FLAG 0x80000000L

#define CKM\_FAKE\_RANDOM 0x80000efeL

#define CKM\_INVALID\_MECHANISM 0xffffffffL

#define SECMOD\_INTERNAL 1

#define SECMOD\_FIPS 2

#define PK11\_PW\_AUTHENTICATED "AUTH"

#define PK11\_PW\_RETRY "RETRY"

#define SECMOD\_INT\_FLAGS SECMOD\_MAKE\_NSS\_FLAGS("",1)

#define SECMOD\_FIPS\_FLAGS SECMOD\_MAKE\_NSS\_FLAGS(",fips",3)

#define PK11\_PW\_TRY "TRY"

typedef struct SECMODModuleStr {

PLArenaPool \*arena;

PRBool internal;

PRBool loaded;

PRBool isFIPS;

char \*dllName;

char \*commonName;

void \*library;

void \*functionList;

PRLock \*refLock;

int refCount;

PK11SlotInfo \*\*slots;

int slotCount;

PK11PreSlotInfo \*slotInfo;

int slotInfoCount;

SECMODModuleID moduleID;

PRBool isThreadSafe;

unsigned long int ssl[1];

char \*libraryParams;

void \*moduleDBFunc;

SECMODModule \*parent;

PRBool isCritical;

PRBool isModuleDB;

PRBool moduleDBOnly;

int trustOrder;

int cipherOrder;

unsigned long int evControlMask;

CK\_VERSION cryptokiVersion;

} SECMODModule;

typedef struct SECMODModuleListStr {

SECMODModuleList \*next;

SECMODModule \*module;

} SECMODModuleList;

typedef NSSRWLock SECMODListLock;

typedef struct PK11SlotInfoStr PK11SlotInfo;

typedef struct PK11PreSlotInfoStr PK11PreSlotInfo;

typedef struct PK11SymKeyStr PK11SymKey;

typedef struct PK11ContextStr PK11Context;

typedef struct PK11SlotListStr PK11SlotList;

typedef struct PK11SlotListElementStr PK11SlotListElement;

typedef unsigned long int SECMODModuleID;

typedef struct PK11DefaultArrayEntryStr PK11DefaultArrayEntry;

typedef struct PK11GenericObjectStr PK11GenericObject;

typedef void (\*PK11FreeDataFunc) (void \*);

typedef enum {

PK11CertListUnique,

PK11CertListUser = 1,

PK11CertListRootUnique = 2,

PK11CertListCA = 3,

PK11CertListCAUnique = 4,

PK11CertListUserUnique = 5,

PK11CertListAll = 6

} PK11CertListType;

typedef PRUint32 PK11AttrFlags;

typedef enum {

PK11\_OriginNULL,

PK11\_OriginDerive = 1,

PK11\_OriginGenerated = 2,

PK11\_OriginFortezzaHack = 3,

PK11\_OriginUnwrap = 4

} PK11Origin;

typedef enum {

PK11\_DIS\_NONE,

PK11\_DIS\_USER\_SELECTED = 1,

PK11\_DIS\_COULD\_NOT\_INIT\_TOKEN = 2,

PK11\_DIS\_TOKEN\_VERIFY\_FAILED = 3,

PK11\_DIS\_TOKEN\_NOT\_PRESENT = 4

} PK11DisableReasons;

typedef enum {

PK11\_TypeGeneric,

PK11\_TypePrivKey = 1,

PK11\_TypePubKey = 2,

PK11\_TypeCert = 3,

PK11\_TypeSymKey = 4

} PK11ObjectType;

typedef char \*(\*PK11PasswordFunc) (PK11SlotInfo \*, PRBool, void \*);

typedef struct SECKEYAttributeStr {

SECItem attrType;

SECItem \*\*attrValue;

} SECKEYAttribute;

typedef struct SECKEYPrivateKeyInfoStr {

PLArenaPool \*arena;

SECItem version;

SECAlgorithmID algorithm;

SECItem privateKey;

SECKEYAttribute \*\*attributes;

} SECKEYPrivateKeyInfo;

typedef struct SECKEYEncryptedPrivateKeyInfoStr {

PLArenaPool \*arena;

SECAlgorithmID algorithm;

SECItem encryptedData;

} SECKEYEncryptedPrivateKeyInfo;

typedef enum {

PK11TokenNotRemovable,

PK11TokenPresent = 1,

PK11TokenChanged = 2,

PK11TokenRemoved = 3

} PK11TokenStatus;

typedef enum {

PK11TokenRemovedOrChangedEvent,

PK11TokenPresentEvent = 1

} PK11TokenEvent;

### **24.4.25 nss3/secoidt.h**

#define \_SECOIDT\_H\_

typedef struct SECOidDataStr {

SECItem oid;

SECOidTag offset;

const char \*desc;

unsigned long int mechanism;

SECSupportExtenTag supportedExtension;

} SECOidData;

typedef struct SECAlgorithmIDStr {

SECItem algorithm;

SECItem parameters;

} SECAlgorithmID;

typedef enum {

SEC\_OID\_UNKNOWN,

SEC\_OID\_MD2 = 1,

SEC\_OID\_MD4 = 2,

SEC\_OID\_MD5 = 3,

SEC\_OID\_SHA1 = 4,

SEC\_OID\_RC2\_CBC = 5,

SEC\_OID\_RC4 = 6,

SEC\_OID\_DES\_EDE3\_CBC = 7,

SEC\_OID\_RC5\_CBC\_PAD = 8,

SEC\_OID\_DES\_ECB = 9,

SEC\_OID\_DES\_CBC = 10,

SEC\_OID\_DES\_OFB = 11,

SEC\_OID\_DES\_CFB = 12,

SEC\_OID\_DES\_MAC = 13,

SEC\_OID\_DES\_EDE = 14,

SEC\_OID\_ISO\_SHA\_WITH\_RSA\_SIGNATURE = 15,

SEC\_OID\_PKCS1\_RSA\_ENCRYPTION = 16,

SEC\_OID\_PKCS1\_MD2\_WITH\_RSA\_ENCRYPTION = 17,

SEC\_OID\_PKCS1\_MD4\_WITH\_RSA\_ENCRYPTION = 18,

SEC\_OID\_PKCS1\_MD5\_WITH\_RSA\_ENCRYPTION = 19,

SEC\_OID\_PKCS1\_SHA1\_WITH\_RSA\_ENCRYPTION = 20,

SEC\_OID\_PKCS5\_PBE\_WITH\_MD2\_AND\_DES\_CBC = 21,

SEC\_OID\_PKCS5\_PBE\_WITH\_MD5\_AND\_DES\_CBC = 22,

SEC\_OID\_PKCS5\_PBE\_WITH\_SHA1\_AND\_DES\_CBC = 23,

SEC\_OID\_PKCS7 = 24,

SEC\_OID\_PKCS7\_DATA = 25,

SEC\_OID\_PKCS7\_SIGNED\_DATA = 26,

SEC\_OID\_PKCS7\_ENVELOPED\_DATA = 27,

SEC\_OID\_PKCS7\_SIGNED\_ENVELOPED\_DATA = 28,

SEC\_OID\_PKCS7\_DIGESTED\_DATA = 29,

SEC\_OID\_PKCS7\_ENCRYPTED\_DATA = 30,

SEC\_OID\_PKCS9\_EMAIL\_ADDRESS = 31,

SEC\_OID\_PKCS9\_UNSTRUCTURED\_NAME = 32,

SEC\_OID\_PKCS9\_CONTENT\_TYPE = 33,

SEC\_OID\_PKCS9\_MESSAGE\_DIGEST = 34,

SEC\_OID\_PKCS9\_SIGNING\_TIME = 35,

SEC\_OID\_PKCS9\_COUNTER\_SIGNATURE = 36,

SEC\_OID\_PKCS9\_CHALLENGE\_PASSWORD = 37,

SEC\_OID\_PKCS9\_UNSTRUCTURED\_ADDRESS = 38,

SEC\_OID\_PKCS9\_EXTENDED\_CERTIFICATE\_ATTRIBUTES = 39,

SEC\_OID\_PKCS9\_SMIME\_CAPABILITIES = 40,

SEC\_OID\_AVA\_COMMON\_NAME = 41,

SEC\_OID\_AVA\_COUNTRY\_NAME = 42,

SEC\_OID\_AVA\_LOCALITY = 43,

SEC\_OID\_AVA\_STATE\_OR\_PROVINCE = 44,

SEC\_OID\_AVA\_ORGANIZATION\_NAME = 45,

SEC\_OID\_AVA\_ORGANIZATIONAL\_UNIT\_NAME = 46,

SEC\_OID\_AVA\_DN\_QUALIFIER = 47,

SEC\_OID\_AVA\_DC = 48,

SEC\_OID\_NS\_TYPE\_GIF = 49,

SEC\_OID\_NS\_TYPE\_JPEG = 50,

SEC\_OID\_NS\_TYPE\_URL = 51,

SEC\_OID\_NS\_TYPE\_HTML = 52,

SEC\_OID\_NS\_TYPE\_CERT\_SEQUENCE = 53,

SEC\_OID\_MISSI\_KEA\_DSS\_OLD = 54,

SEC\_OID\_MISSI\_DSS\_OLD = 55,

SEC\_OID\_MISSI\_KEA\_DSS = 56,

SEC\_OID\_MISSI\_DSS = 57,

SEC\_OID\_MISSI\_KEA = 58,

SEC\_OID\_MISSI\_ALT\_KEA = 59,

SEC\_OID\_NS\_CERT\_EXT\_NETSCAPE\_OK = 60,

SEC\_OID\_NS\_CERT\_EXT\_ISSUER\_LOGO = 61,

SEC\_OID\_NS\_CERT\_EXT\_SUBJECT\_LOGO = 62,

SEC\_OID\_NS\_CERT\_EXT\_CERT\_TYPE = 63,

SEC\_OID\_NS\_CERT\_EXT\_BASE\_URL = 64,

SEC\_OID\_NS\_CERT\_EXT\_REVOCATION\_URL = 65,

SEC\_OID\_NS\_CERT\_EXT\_CA\_REVOCATION\_URL = 66,

SEC\_OID\_NS\_CERT\_EXT\_CA\_CRL\_URL = 67,

SEC\_OID\_NS\_CERT\_EXT\_CA\_CERT\_URL = 68,

SEC\_OID\_NS\_CERT\_EXT\_CERT\_RENEWAL\_URL = 69,

SEC\_OID\_NS\_CERT\_EXT\_CA\_POLICY\_URL = 70,

SEC\_OID\_NS\_CERT\_EXT\_HOMEPAGE\_URL = 71,

SEC\_OID\_NS\_CERT\_EXT\_ENTITY\_LOGO = 72,

SEC\_OID\_NS\_CERT\_EXT\_USER\_PICTURE = 73,

SEC\_OID\_NS\_CERT\_EXT\_SSL\_SERVER\_NAME = 74,

SEC\_OID\_NS\_CERT\_EXT\_COMMENT = 75,

SEC\_OID\_NS\_CERT\_EXT\_LOST\_PASSWORD\_URL = 76,

SEC\_OID\_NS\_CERT\_EXT\_CERT\_RENEWAL\_TIME = 77,

SEC\_OID\_NS\_KEY\_USAGE\_GOVT\_APPROVED = 78,

SEC\_OID\_X509\_SUBJECT\_DIRECTORY\_ATTR = 79,

SEC\_OID\_X509\_SUBJECT\_KEY\_ID = 80,

SEC\_OID\_X509\_KEY\_USAGE = 81,

SEC\_OID\_X509\_PRIVATE\_KEY\_USAGE\_PERIOD = 82,

SEC\_OID\_X509\_SUBJECT\_ALT\_NAME = 83,

SEC\_OID\_X509\_ISSUER\_ALT\_NAME = 84,

SEC\_OID\_X509\_BASIC\_CONSTRAINTS = 85,

SEC\_OID\_X509\_NAME\_CONSTRAINTS = 86,

SEC\_OID\_X509\_CRL\_DIST\_POINTS = 87,

SEC\_OID\_X509\_CERTIFICATE\_POLICIES = 88,

SEC\_OID\_X509\_POLICY\_MAPPINGS = 89,

SEC\_OID\_X509\_POLICY\_CONSTRAINTS = 90,

SEC\_OID\_X509\_AUTH\_KEY\_ID = 91,

SEC\_OID\_X509\_EXT\_KEY\_USAGE = 92,

SEC\_OID\_X509\_AUTH\_INFO\_ACCESS = 93,

SEC\_OID\_X509\_CRL\_NUMBER = 94,

SEC\_OID\_X509\_REASON\_CODE = 95,

SEC\_OID\_X509\_INVALID\_DATE = 96,

SEC\_OID\_X500\_RSA\_ENCRYPTION = 97,

SEC\_OID\_RFC1274\_UID = 98,

SEC\_OID\_RFC1274\_MAIL = 99,

SEC\_OID\_PKCS12 = 100,

SEC\_OID\_PKCS12\_MODE\_IDS = 101,

SEC\_OID\_PKCS12\_ESPVK\_IDS = 102,

SEC\_OID\_PKCS12\_BAG\_IDS = 103,

SEC\_OID\_PKCS12\_CERT\_BAG\_IDS = 104,

SEC\_OID\_PKCS12\_OIDS = 105,

SEC\_OID\_PKCS12\_PBE\_IDS = 106,

SEC\_OID\_PKCS12\_SIGNATURE\_IDS = 107,

SEC\_OID\_PKCS12\_ENVELOPING\_IDS = 108,

SEC\_OID\_PKCS12\_PKCS8\_KEY\_SHROUDING = 109,

SEC\_OID\_PKCS12\_KEY\_BAG\_ID = 110,

SEC\_OID\_PKCS12\_CERT\_AND\_CRL\_BAG\_ID = 111,

SEC\_OID\_PKCS12\_SECRET\_BAG\_ID = 112,

SEC\_OID\_PKCS12\_X509\_CERT\_CRL\_BAG = 113,

SEC\_OID\_PKCS12\_SDSI\_CERT\_BAG = 114,

SEC\_OID\_PKCS12\_PBE\_WITH\_SHA1\_AND\_128\_BIT\_RC4 = 115,

SEC\_OID\_PKCS12\_PBE\_WITH\_SHA1\_AND\_40\_BIT\_RC4 = 116,

SEC\_OID\_PKCS12\_PBE\_WITH\_SHA1\_AND\_TRIPLE\_DES\_CBC = 117,

SEC\_OID\_PKCS12\_PBE\_WITH\_SHA1\_AND\_128\_BIT\_RC2\_CBC = 118,

SEC\_OID\_PKCS12\_PBE\_WITH\_SHA1\_AND\_40\_BIT\_RC2\_CBC = 119,

SEC\_OID\_PKCS12\_RSA\_ENCRYPTION\_WITH\_128\_BIT\_RC4 = 120,

SEC\_OID\_PKCS12\_RSA\_ENCRYPTION\_WITH\_40\_BIT\_RC4 = 121,

SEC\_OID\_PKCS12\_RSA\_ENCRYPTION\_WITH\_TRIPLE\_DES = 122,

SEC\_OID\_PKCS12\_RSA\_SIGNATURE\_WITH\_SHA1\_DIGEST = 123,

SEC\_OID\_ANSIX9\_DSA\_SIGNATURE = 124,

SEC\_OID\_ANSIX9\_DSA\_SIGNATURE\_WITH\_SHA1\_DIGEST = 125,

SEC\_OID\_BOGUS\_DSA\_SIGNATURE\_WITH\_SHA1\_DIGEST = 126,

SEC\_OID\_VERISIGN\_USER\_NOTICES = 127,

SEC\_OID\_PKIX\_CPS\_POINTER\_QUALIFIER = 128,

SEC\_OID\_PKIX\_USER\_NOTICE\_QUALIFIER = 129,

SEC\_OID\_PKIX\_OCSP = 130,

SEC\_OID\_PKIX\_OCSP\_BASIC\_RESPONSE = 131,

SEC\_OID\_PKIX\_OCSP\_NONCE = 132,

SEC\_OID\_PKIX\_OCSP\_CRL = 133,

SEC\_OID\_PKIX\_OCSP\_RESPONSE = 134,

SEC\_OID\_PKIX\_OCSP\_NO\_CHECK = 135,

SEC\_OID\_PKIX\_OCSP\_ARCHIVE\_CUTOFF = 136,

SEC\_OID\_PKIX\_OCSP\_SERVICE\_LOCATOR = 137,

SEC\_OID\_PKIX\_REGCTRL\_REGTOKEN = 138,

SEC\_OID\_PKIX\_REGCTRL\_AUTHENTICATOR = 139,

SEC\_OID\_PKIX\_REGCTRL\_PKIPUBINFO = 140,

SEC\_OID\_PKIX\_REGCTRL\_PKI\_ARCH\_OPTIONS = 141,

SEC\_OID\_PKIX\_REGCTRL\_OLD\_CERT\_ID = 142,

SEC\_OID\_PKIX\_REGCTRL\_PROTOCOL\_ENC\_KEY = 143,

SEC\_OID\_PKIX\_REGINFO\_UTF8\_PAIRS = 144,

SEC\_OID\_PKIX\_REGINFO\_CERT\_REQUEST = 145,

SEC\_OID\_EXT\_KEY\_USAGE\_SERVER\_AUTH = 146,

SEC\_OID\_EXT\_KEY\_USAGE\_CLIENT\_AUTH = 147,

SEC\_OID\_EXT\_KEY\_USAGE\_CODE\_SIGN = 148,

SEC\_OID\_EXT\_KEY\_USAGE\_EMAIL\_PROTECT = 149,

SEC\_OID\_EXT\_KEY\_USAGE\_TIME\_STAMP = 150,

SEC\_OID\_OCSP\_RESPONDER = 151,

SEC\_OID\_NETSCAPE\_SMIME\_KEA = 152,

SEC\_OID\_FORTEZZA\_SKIPJACK = 153,

SEC\_OID\_PKCS12\_V2\_PBE\_WITH\_SHA1\_AND\_128\_BIT\_RC4 = 154,

SEC\_OID\_PKCS12\_V2\_PBE\_WITH\_SHA1\_AND\_40\_BIT\_RC4 = 155,

SEC\_OID\_PKCS12\_V2\_PBE\_WITH\_SHA1\_AND\_3KEY\_TRIPLE\_DES\_CBC = 156,

SEC\_OID\_PKCS12\_V2\_PBE\_WITH\_SHA1\_AND\_2KEY\_TRIPLE\_DES\_CBC = 157,

SEC\_OID\_PKCS12\_V2\_PBE\_WITH\_SHA1\_AND\_128\_BIT\_RC2\_CBC = 158,

SEC\_OID\_PKCS12\_V2\_PBE\_WITH\_SHA1\_AND\_40\_BIT\_RC2\_CBC = 159,

SEC\_OID\_PKCS12\_SAFE\_CONTENTS\_ID = 160,

SEC\_OID\_PKCS12\_PKCS8\_SHROUDED\_KEY\_BAG\_ID = 161,

SEC\_OID\_PKCS12\_V1\_KEY\_BAG\_ID = 162,

SEC\_OID\_PKCS12\_V1\_PKCS8\_SHROUDED\_KEY\_BAG\_ID = 163,

SEC\_OID\_PKCS12\_V1\_CERT\_BAG\_ID = 164,

SEC\_OID\_PKCS12\_V1\_CRL\_BAG\_ID = 165,

SEC\_OID\_PKCS12\_V1\_SECRET\_BAG\_ID = 166,

SEC\_OID\_PKCS12\_V1\_SAFE\_CONTENTS\_BAG\_ID = 167,

SEC\_OID\_PKCS9\_X509\_CERT = 168,

SEC\_OID\_PKCS9\_SDSI\_CERT = 169,

SEC\_OID\_PKCS9\_X509\_CRL = 170,

SEC\_OID\_PKCS9\_FRIENDLY\_NAME = 171,

SEC\_OID\_PKCS9\_LOCAL\_KEY\_ID = 172,

SEC\_OID\_BOGUS\_KEY\_USAGE = 173,

SEC\_OID\_X942\_DIFFIE\_HELMAN\_KEY = 174,

SEC\_OID\_NETSCAPE\_NICKNAME = 175,

SEC\_OID\_NETSCAPE\_RECOVERY\_REQUEST = 176,

SEC\_OID\_CERT\_RENEWAL\_LOCATOR = 177,

SEC\_OID\_NS\_CERT\_EXT\_SCOPE\_OF\_USE = 178,

SEC\_OID\_CMS\_EPHEMERAL\_STATIC\_DIFFIE\_HELLMAN = 179,

SEC\_OID\_CMS\_3DES\_KEY\_WRAP = 180,

SEC\_OID\_CMS\_RC2\_KEY\_WRAP = 181,

SEC\_OID\_SMIME\_ENCRYPTION\_KEY\_PREFERENCE = 182,

SEC\_OID\_AES\_128\_ECB = 183,

SEC\_OID\_AES\_128\_CBC = 184,

SEC\_OID\_AES\_192\_ECB = 185,

SEC\_OID\_AES\_192\_CBC = 186,

SEC\_OID\_AES\_256\_ECB = 187,

SEC\_OID\_AES\_256\_CBC = 188,

SEC\_OID\_SDN702\_DSA\_SIGNATURE = 189,

SEC\_OID\_MS\_SMIME\_ENCRYPTION\_KEY\_PREFERENCE = 190,

SEC\_OID\_SHA256 = 191,

SEC\_OID\_SHA384 = 192,

SEC\_OID\_SHA512 = 193,

SEC\_OID\_PKCS1\_SHA256\_WITH\_RSA\_ENCRYPTION = 194,

SEC\_OID\_PKCS1\_SHA384\_WITH\_RSA\_ENCRYPTION = 195,

SEC\_OID\_PKCS1\_SHA512\_WITH\_RSA\_ENCRYPTION = 196,

SEC\_OID\_AES\_128\_KEY\_WRAP = 197,

SEC\_OID\_AES\_192\_KEY\_WRAP = 198,

SEC\_OID\_AES\_256\_KEY\_WRAP = 199,

SEC\_OID\_ANSIX962\_EC\_PUBLIC\_KEY = 200,

SEC\_OID\_ANSIX962\_ECDSA\_SHA1\_SIGNATURE = 201,

SEC\_OID\_ANSIX962\_EC\_PRIME192V1 = 202,

SEC\_OID\_ANSIX962\_EC\_PRIME192V2 = 203,

SEC\_OID\_ANSIX962\_EC\_PRIME192V3 = 204,

SEC\_OID\_ANSIX962\_EC\_PRIME239V1 = 205,

SEC\_OID\_ANSIX962\_EC\_PRIME239V2 = 206,

SEC\_OID\_ANSIX962\_EC\_PRIME239V3 = 207,

SEC\_OID\_ANSIX962\_EC\_PRIME256V1 = 208,

SEC\_OID\_SECG\_EC\_SECP112R1 = 209,

SEC\_OID\_SECG\_EC\_SECP112R2 = 210,

SEC\_OID\_SECG\_EC\_SECP128R1 = 211,

SEC\_OID\_SECG\_EC\_SECP128R2 = 212,

SEC\_OID\_SECG\_EC\_SECP160K1 = 213,

SEC\_OID\_SECG\_EC\_SECP160R1 = 214,

SEC\_OID\_SECG\_EC\_SECP160R2 = 215,

SEC\_OID\_SECG\_EC\_SECP192K1 = 216,

SEC\_OID\_SECG\_EC\_SECP224K1 = 217,

SEC\_OID\_SECG\_EC\_SECP224R1 = 218,

SEC\_OID\_SECG\_EC\_SECP256K1 = 219,

SEC\_OID\_SECG\_EC\_SECP384R1 = 220,

SEC\_OID\_SECG\_EC\_SECP521R1 = 221,

SEC\_OID\_ANSIX962\_EC\_C2PNB163V1 = 222,

SEC\_OID\_ANSIX962\_EC\_C2PNB163V2 = 223,

SEC\_OID\_ANSIX962\_EC\_C2PNB163V3 = 224,

SEC\_OID\_ANSIX962\_EC\_C2PNB176V1 = 225,

SEC\_OID\_ANSIX962\_EC\_C2TNB191V1 = 226,

SEC\_OID\_ANSIX962\_EC\_C2TNB191V2 = 227,

SEC\_OID\_ANSIX962\_EC\_C2TNB191V3 = 228,

SEC\_OID\_ANSIX962\_EC\_C2ONB191V4 = 229,

SEC\_OID\_ANSIX962\_EC\_C2ONB191V5 = 230,

SEC\_OID\_ANSIX962\_EC\_C2PNB208W1 = 231,

SEC\_OID\_ANSIX962\_EC\_C2TNB239V1 = 232,

SEC\_OID\_ANSIX962\_EC\_C2TNB239V2 = 233,

SEC\_OID\_ANSIX962\_EC\_C2TNB239V3 = 234,

SEC\_OID\_ANSIX962\_EC\_C2ONB239V4 = 235,

SEC\_OID\_ANSIX962\_EC\_C2ONB239V5 = 236,

SEC\_OID\_ANSIX962\_EC\_C2PNB272W1 = 237,

SEC\_OID\_ANSIX962\_EC\_C2PNB304W1 = 238,

SEC\_OID\_ANSIX962\_EC\_C2TNB359V1 = 239,

SEC\_OID\_ANSIX962\_EC\_C2PNB368W1 = 240,

SEC\_OID\_ANSIX962\_EC\_C2TNB431R1 = 241,

SEC\_OID\_SECG\_EC\_SECT113R1 = 242,

SEC\_OID\_SECG\_EC\_SECT113R2 = 243,

SEC\_OID\_SECG\_EC\_SECT131R1 = 244,

SEC\_OID\_SECG\_EC\_SECT131R2 = 245,

SEC\_OID\_SECG\_EC\_SECT163K1 = 246,

SEC\_OID\_SECG\_EC\_SECT163R1 = 247,

SEC\_OID\_SECG\_EC\_SECT163R2 = 248,

SEC\_OID\_SECG\_EC\_SECT193R1 = 249,

SEC\_OID\_SECG\_EC\_SECT193R2 = 250,

SEC\_OID\_SECG\_EC\_SECT233K1 = 251,

SEC\_OID\_SECG\_EC\_SECT233R1 = 252,

SEC\_OID\_SECG\_EC\_SECT239K1 = 253,

SEC\_OID\_SECG\_EC\_SECT283K1 = 254,

SEC\_OID\_SECG\_EC\_SECT283R1 = 255,

SEC\_OID\_SECG\_EC\_SECT409K1 = 256,

SEC\_OID\_SECG\_EC\_SECT409R1 = 257,

SEC\_OID\_SECG\_EC\_SECT571K1 = 258,

SEC\_OID\_SECG\_EC\_SECT571R1 = 259,

SEC\_OID\_NETSCAPE\_AOLSCREENNAME = 260,

SEC\_OID\_AVA\_SURNAME = 261,

SEC\_OID\_AVA\_SERIAL\_NUMBER = 262,

SEC\_OID\_AVA\_STREET\_ADDRESS = 263,

SEC\_OID\_AVA\_TITLE = 264,

SEC\_OID\_AVA\_POSTAL\_ADDRESS = 265,

SEC\_OID\_AVA\_POSTAL\_CODE = 266,

SEC\_OID\_AVA\_POST\_OFFICE\_BOX = 267,

SEC\_OID\_AVA\_GIVEN\_NAME = 268,

SEC\_OID\_AVA\_INITIALS = 269,

SEC\_OID\_AVA\_GENERATION\_QUALIFIER = 270,

SEC\_OID\_AVA\_HOUSE\_IDENTIFIER = 271,

SEC\_OID\_AVA\_PSEUDONYM = 272,

SEC\_OID\_PKIX\_CA\_ISSUERS = 273,

SEC\_OID\_PKCS9\_EXTENSION\_REQUEST = 274,

SEC\_OID\_ANSIX962\_ECDSA\_SIGNATURE\_RECOMMENDED\_DIGEST = 275,

SEC\_OID\_ANSIX962\_ECDSA\_SIGNATURE\_SPECIFIED\_DIGEST = 276,

SEC\_OID\_ANSIX962\_ECDSA\_SHA224\_SIGNATURE = 277,

SEC\_OID\_ANSIX962\_ECDSA\_SHA256\_SIGNATURE = 278,

SEC\_OID\_ANSIX962\_ECDSA\_SHA384\_SIGNATURE = 279,

SEC\_OID\_ANSIX962\_ECDSA\_SHA512\_SIGNATURE = 280,

SEC\_OID\_X509\_HOLD\_INSTRUCTION\_CODE = 281,

SEC\_OID\_X509\_DELTA\_CRL\_INDICATOR = 282,

SEC\_OID\_X509\_ISSUING\_DISTRIBUTION\_POINT = 283,

SEC\_OID\_X509\_CERT\_ISSUER = 284,

SEC\_OID\_X509\_FRESHEST\_CRL = 285,

SEC\_OID\_X509\_INHIBIT\_ANY\_POLICY = 286,

SEC\_OID\_X509\_SUBJECT\_INFO\_ACCESS = 287,

SEC\_OID\_CAMELLIA\_128\_CBC = 288,

SEC\_OID\_CAMELLIA\_192\_CBC = 289,

SEC\_OID\_CAMELLIA\_256\_CBC = 290,

SEC\_OID\_PKCS5\_PBKDF2 = 291,

SEC\_OID\_PKCS5\_PBES2 = 292,

SEC\_OID\_PKCS5\_PBMAC1 = 293,

SEC\_OID\_HMAC\_SHA1 = 294,

SEC\_OID\_HMAC\_SHA224 = 295,

SEC\_OID\_HMAC\_SHA256 = 296,

SEC\_OID\_HMAC\_SHA384 = 297,

SEC\_OID\_HMAC\_SHA512 = 298,

SEC\_OID\_PKIX\_TIMESTAMPING = 299,

SEC\_OID\_PKIX\_CA\_REPOSITORY = 300,

SEC\_OID\_ISO\_SHA1\_WITH\_RSA\_SIGNATURE = 301,

SEC\_OID\_TOTAL = 302

} SECOidTag;

typedef enum {

INVALID\_CERT\_EXTENSION,

UNSUPPORTED\_CERT\_EXTENSION = 1,

SUPPORTED\_CERT\_EXTENSION = 2

} SECSupportExtenTag;

### **24.4.26 nss3/secpkcs5.h**

#define \_SECPKCS5\_H\_

typedef enum {

pbeBitGenIDNull,

pbeBitGenCipherKey = 1,

pbeBitGenCipherIV = 2,

pbeBitGenIntegrityKey = 3

} PBEBitGenID;

typedef struct PBEBitGenContextStr PBEBitGenContext;

### **24.4.27 nss3/secport.h**

#define \_SECPORT\_H\_

typedef PRBool(\*PORTCharConversionWSwapFunc) (PRBool, unsigned char \*,

unsigned int,

unsigned char \*,

unsigned int, unsigned int \*,

PRBool);

typedef PRBool(\*PORTCharConversionFunc) (PRBool, unsigned char \*,

unsigned int, unsigned char \*,

unsigned int, unsigned int \*);

## **24.5 Interfaces for libssl3**

[Table 24-5](#ID_LIB_45_LIBSSL3_45_DEF) defines the library name and shared object name for the libssl3 library

**Table 24-5 libssl3 Definition**

|  |  |
| --- | --- |
| Library: | libssl3 |
| SONAME: | libssl3.so |

The behavior of the interfaces in this library is specified by the following specifications:

|  |
| --- |
| [NSS SSL] [Mozilla's NSS SSL Reference](#ID_STD_46_NSS_46_SSL) |

### **24.5.1 NSS SSL**

#### 24.5.1.1 Interfaces for NSS SSL

An LSB conforming implementation shall provide the generic functions for NSS SSL specified in [Table 24-6](#ID_TBL_45_LIBSSL3_45_NSS_45_S_45_INTS), with the full mandatory functionality as described in the referenced underlying specification.

**Table 24-6 libssl3 - NSS SSL Function Interfaces**

|  |  |  |
| --- | --- | --- |
| NSS\_CmpCertChainWCANames(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) | NSS\_FindCertKEAType(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) | NSS\_GetClientAuthData(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) |
| SSL\_AuthCertificate(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) | SSL\_AuthCertificateHook(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) | SSL\_BadCertHook(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) |
| SSL\_CipherPolicyGet(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) | SSL\_CipherPolicySet(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) | SSL\_CipherPrefGet(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) |
| SSL\_CipherPrefGetDefault(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) | SSL\_CipherPrefSet(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) | SSL\_CipherPrefSetDefault(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) |
| SSL\_ClearSessionCache(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) | SSL\_ConfigMPServerSIDCache(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) | SSL\_ConfigSecureServer(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) |
| SSL\_ConfigServerSessionIDCache(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) | SSL\_DataPending(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) | SSL\_ForceHandshake(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) |
| SSL\_GetClientAuthDataHook(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) | SSL\_GetSessionID(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) | SSL\_HandshakeCallback(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) |
| SSL\_ImportFD(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) | SSL\_InheritMPServerSIDCache(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) | SSL\_InvalidateSession(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) |
| SSL\_OptionGet(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) | SSL\_OptionGetDefault(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) | SSL\_OptionSet(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) |
| SSL\_OptionSetDefault(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) | SSL\_PeerCertificate(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) | SSL\_ReHandshake(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) |
| SSL\_ResetHandshake(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) | SSL\_RevealPinArg(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) | SSL\_RevealURL(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) |
| SSL\_SecurityStatus(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) | SSL\_SetPKCS11PinArg(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) | SSL\_SetSockPeerID(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) |
| SSL\_SetURL(NSS\_3.2) [[NSS SSL]](#ID_REFSTD_46_LIBSSL3_46_1) |  |  |

## **24.6 Data Definitions for libssl3**

This section defines global identifiers and their values that are associated with interfaces contained in libssl3. These definitions are organized into groups that correspond to system headers. This convention is used as a convenience for the reader, and does not imply the existence of these headers, or their content. Where an interface is defined as requiring a particular system header file all of the data definitions for that system header file presented here shall be in effect.

This section gives data definitions to promote binary application portability, not to repeat source interface definitions available elsewhere. System providers and application developers should use this ABI to supplement - not to replace - source interface definition specifications.

This specification uses the [ISO C (1999)](#ID_STD_46_ISOC99) C Language as the reference programming language, and data definitions are specified in ISO C format. The C language is used here as a convenient notation. Using a C language description of these data objects does not preclude their use by other programming languages.

### **24.6.1 nss3/ecl-exp.h**

#define \_\_ecl\_exp\_h\_

#define ECCurve\_SECG\_CHAR2\_163R2 ECCurve\_NIST\_B163

#define ECCurve\_SECG\_CHAR2\_233R1 ECCurve\_NIST\_B233

#define ECCurve\_WTLS\_11 ECCurve\_NIST\_B233

#define ECCurve\_SECG\_CHAR2\_283R1 ECCurve\_NIST\_B283

#define ECCurve\_SECG\_CHAR2\_409R1 ECCurve\_NIST\_B409

#define ECCurve\_SECG\_CHAR2\_571R1 ECCurve\_NIST\_B571

#define ECCurve\_SECG\_CHAR2\_163K1 ECCurve\_NIST\_K163

#define ECCurve\_WTLS\_3 ECCurve\_NIST\_K163

#define ECCurve\_SECG\_CHAR2\_233K1 ECCurve\_NIST\_K233

#define ECCurve\_WTLS\_10 ECCurve\_NIST\_K233

#define ECCurve\_SECG\_CHAR2\_283K1 ECCurve\_NIST\_K283

#define ECCurve\_SECG\_CHAR2\_409K1 ECCurve\_NIST\_K409

#define ECCurve\_SECG\_CHAR2\_571K1 ECCurve\_NIST\_K571

#define ECCurve\_SECG\_PRIME\_192R1 ECCurve\_NIST\_P192

#define ECCurve\_X9\_62\_PRIME\_192V1 ECCurve\_NIST\_P192

#define ECCurve\_SECG\_PRIME\_224R1 ECCurve\_NIST\_P224

#define ECCurve\_WTLS\_12 ECCurve\_NIST\_P224

#define ECCurve\_SECG\_PRIME\_256R1 ECCurve\_NIST\_P256

#define ECCurve\_X9\_62\_PRIME\_256V1 ECCurve\_NIST\_P256

#define ECCurve\_SECG\_PRIME\_384R1 ECCurve\_NIST\_P384

#define ECCurve\_SECG\_PRIME\_521R1 ECCurve\_NIST\_P521

#define ECCurve\_WTLS\_4 ECCurve\_SECG\_CHAR2\_113R1

#define ECCurve\_WTLS\_6 ECCurve\_SECG\_PRIME\_112R1

#define ECCurve\_WTLS\_7 ECCurve\_SECG\_PRIME\_160R1

#define ECCurve\_WTLS\_5 ECCurve\_X9\_62\_CHAR2\_PNB163V1

enum ECField {

ECField\_GFp = 0,

ECField\_GF2m = 1

};

typedef struct ECCurveParamsStr {

char \*text;

enum ECField field;

unsigned int size;

char \*irr;

char \*curvea;

char \*curveb;

char \*genx;

char \*geny;

char \*order;

int cofactor;

} ECCurveParams;

enum ECCurveName {

ECCurve\_noName = 0,

ECCurve\_NIST\_P192 = 1,

ECCurve\_NIST\_P224 = 2,

ECCurve\_NIST\_P256 = 3,

ECCurve\_NIST\_P384 = 4,

ECCurve\_NIST\_P521 = 5,

ECCurve\_NIST\_K163 = 6,

ECCurve\_NIST\_B163 = 7,

ECCurve\_NIST\_K233 = 8,

ECCurve\_NIST\_B233 = 9,

ECCurve\_NIST\_K283 = 10,

ECCurve\_NIST\_B283 = 11,

ECCurve\_NIST\_K409 = 12,

ECCurve\_NIST\_B409 = 13,

ECCurve\_NIST\_K571 = 14,

ECCurve\_NIST\_B571 = 15,

ECCurve\_X9\_62\_PRIME\_192V2 = 16,

ECCurve\_X9\_62\_PRIME\_192V3 = 17,

ECCurve\_X9\_62\_PRIME\_239V1 = 18,

ECCurve\_X9\_62\_PRIME\_239V2 = 19,

ECCurve\_X9\_62\_PRIME\_239V3 = 20,

ECCurve\_X9\_62\_CHAR2\_PNB163V1 = 21,

ECCurve\_X9\_62\_CHAR2\_PNB163V2 = 22,

ECCurve\_X9\_62\_CHAR2\_PNB163V3 = 23,

ECCurve\_X9\_62\_CHAR2\_PNB176V1 = 24,

ECCurve\_X9\_62\_CHAR2\_TNB191V1 = 25,

ECCurve\_X9\_62\_CHAR2\_TNB191V2 = 26,

ECCurve\_X9\_62\_CHAR2\_TNB191V3 = 27,

ECCurve\_X9\_62\_CHAR2\_PNB208W1 = 28,

ECCurve\_X9\_62\_CHAR2\_TNB239V1 = 29,

ECCurve\_X9\_62\_CHAR2\_TNB239V2 = 30,

ECCurve\_X9\_62\_CHAR2\_TNB239V3 = 31,

ECCurve\_X9\_62\_CHAR2\_PNB272W1 = 32,

ECCurve\_X9\_62\_CHAR2\_PNB304W1 = 33,

ECCurve\_X9\_62\_CHAR2\_TNB359V1 = 34,

ECCurve\_X9\_62\_CHAR2\_PNB368W1 = 35,

ECCurve\_X9\_62\_CHAR2\_TNB431R1 = 36,

ECCurve\_SECG\_PRIME\_112R1 = 37,

ECCurve\_SECG\_PRIME\_112R2 = 38,

ECCurve\_SECG\_PRIME\_128R1 = 39,

ECCurve\_SECG\_PRIME\_128R2 = 40,

ECCurve\_SECG\_PRIME\_160K1 = 41,

ECCurve\_SECG\_PRIME\_160R1 = 42,

ECCurve\_SECG\_PRIME\_160R2 = 43,

ECCurve\_SECG\_PRIME\_192K1 = 44,

ECCurve\_SECG\_PRIME\_224K1 = 45,

ECCurve\_SECG\_PRIME\_256K1 = 46,

ECCurve\_SECG\_CHAR2\_113R1 = 47,

ECCurve\_SECG\_CHAR2\_113R2 = 48,

ECCurve\_SECG\_CHAR2\_131R1 = 49,

ECCurve\_SECG\_CHAR2\_131R2 = 50,

ECCurve\_SECG\_CHAR2\_163R1 = 51,

ECCurve\_SECG\_CHAR2\_193R1 = 52,

ECCurve\_SECG\_CHAR2\_193R2 = 53,

ECCurve\_SECG\_CHAR2\_239K1 = 54,

ECCurve\_WTLS\_1 = 55,

ECCurve\_WTLS\_8 = 56,

ECCurve\_WTLS\_9 = 57,

ECCurve\_pastLastCurve = 58

};

### **24.6.2 nss3/ssl.h**

#define \_\_ssl\_h\_

#define SSL\_IS\_SSL2\_CIPHER(which) (((which) & 0xfff0) == 0xff00)

#define SSL\_REQUIRE\_NEVER ((PRBool)0)

#define SSL\_REQUIRE\_ALWAYS ((PRBool)1)

#define SSL\_REQUIRE\_FIRST\_HANDSHAKE ((PRBool)2)

#define SSL\_REQUIRE\_NO\_ERROR ((PRBool)3)

#define SSL\_SECURITY\_STATUS\_NOOPT -1

#define SSL\_NOT\_ALLOWED 0

#define SSL\_SECURITY\_STATUS\_OFF 0

#define SSL\_ALLOWED 1

#define SSL\_SECURITY 1

#define SSL\_SECURITY\_STATUS\_ON\_HIGH 1

#define SSL\_REQUIRE\_CERTIFICATE 10

#define SSL\_ENABLE\_FDX 11

#define SSL\_V2\_COMPATIBLE\_HELLO 12

#define SSL\_ENABLE\_TLS 13

#define SSL\_ROLLBACK\_DETECTION 14

#define SSL\_NO\_STEP\_DOWN 15

#define SSL\_BYPASS\_PKCS11 16

#define SSL\_NO\_LOCKS 17

#define SSL\_RESTRICTED 2

#define SSL\_SECURITY\_STATUS\_ON\_LOW 2

#define SSL\_SOCKS 2

#define SSL\_REQUEST\_CERTIFICATE 3

#define SSL\_HANDSHAKE\_AS\_CLIENT 5

#define SSL\_HANDSHAKE\_AS\_SERVER 6

#define SSL\_ENABLE\_SSL2 7

#define SSL\_ENABLE\_SSL3 8

#define SSL\_NO\_CACHE 9

#define SSL\_ENV\_VAR\_NAME "SSL\_INHERITANCE"

typedef SECStatus(\*SSLAuthCertificate) (void \*, PRFileDesc \*, PRBool,

PRBool);

typedef SECStatus(\*SSLGetClientAuthData) (void \*, PRFileDesc \*,

CERTDistNames \*,

CERTCertificate \* \*,

SECKEYPrivateKey \* \*);

typedef SECStatus(\*SSLBadCertHandler) (void \*, PRFileDesc \*);

typedef void (\*SSLHandshakeCallback) (PRFileDesc \*, void \*);

extern SECStatus NSS\_CmpCertChainWCANames(CERTCertificate \* cert,

CERTDistNames \* caNames);

extern SSLKEAType NSS\_FindCertKEAType(CERTCertificate \* cert);

extern SECStatus NSS\_GetClientAuthData(void \*arg, PRFileDesc \* socket,

struct CERTDistNamesStr \*caNames,

struct CERTCertificateStr

\*\*pRetCert,

struct SECKEYPrivateKeyStr

\*\*pRetKey);

extern SECStatus SSL\_AuthCertificate(void \*arg, PRFileDesc \* fd,

PRBool checkSig, PRBool isServer);

extern SECStatus SSL\_AuthCertificateHook(PRFileDesc \* fd,

SSLAuthCertificate f, void \*arg);

extern SECStatus SSL\_BadCertHook(PRFileDesc \* fd, SSLBadCertHandler f,

void \*arg);

extern SECStatus SSL\_CipherPolicyGet(PRInt32 cipher, PRInt32 \* policy);

extern SECStatus SSL\_CipherPolicySet(PRInt32 cipher, PRInt32 policy);

extern SECStatus SSL\_CipherPrefGet(PRFileDesc \* fd, PRInt32 cipher,

PRBool \* enabled);

extern SECStatus SSL\_CipherPrefGetDefault(PRInt32 cipher,

PRBool \* enabled);

extern SECStatus SSL\_CipherPrefSet(PRFileDesc \* fd, PRInt32 cipher,

PRBool enabled);

extern SECStatus SSL\_CipherPrefSetDefault(PRInt32 cipher, PRBool enabled);

extern void SSL\_ClearSessionCache(void);

extern SECStatus SSL\_ConfigMPServerSIDCache(int maxCacheEntries,

PRUint32 timeout,

PRUint32 ssl3\_timeout,

const char \*directory);

extern SECStatus SSL\_ConfigSecureServer(PRFileDesc \* fd,

CERTCertificate \* cert,

SECKEYPrivateKey \* key,

SSLKEAType kea);

extern SECStatus SSL\_ConfigServerSessionIDCache(int maxCacheEntries,

PRUint32 timeout,

PRUint32 ssl3\_timeout,

const char \*directory);

extern int SSL\_DataPending(PRFileDesc \* fd);

extern SECStatus SSL\_ForceHandshake(PRFileDesc \* fd);

extern SECStatus SSL\_GetClientAuthDataHook(PRFileDesc \* fd,

SSLGetClientAuthData f,

void \*a);

extern SECItem \*SSL\_GetSessionID(PRFileDesc \* fd);

extern SECStatus SSL\_HandshakeCallback(PRFileDesc \* fd,

SSLHandshakeCallback cb,

void \*client\_data);

extern PRFileDesc \*SSL\_ImportFD(PRFileDesc \* model, PRFileDesc \* fd);

extern SECStatus SSL\_InheritMPServerSIDCache(const char \*envString);

extern SECStatus SSL\_InvalidateSession(PRFileDesc \* fd);

extern SECStatus SSL\_OptionGet(PRFileDesc \* fd, PRInt32 option,

PRBool \* on);

extern SECStatus SSL\_OptionGetDefault(PRInt32 option, PRBool \* on);

extern SECStatus SSL\_OptionSet(PRFileDesc \* fd, PRInt32 option, PRBool on);

extern SECStatus SSL\_OptionSetDefault(PRInt32 option, PRBool on);

extern CERTCertificate \*SSL\_PeerCertificate(PRFileDesc \* fd);

extern SECStatus SSL\_ReHandshake(PRFileDesc \* fd, PRBool flushCache);

extern SECStatus SSL\_ResetHandshake(PRFileDesc \* fd, PRBool asServer);

extern void \*SSL\_RevealPinArg(PRFileDesc \* socket);

extern char \*SSL\_RevealURL(PRFileDesc \* socket);

extern SECStatus SSL\_SecurityStatus(PRFileDesc \* fd, int \*on,

char \*\*cipher, int \*keySize,

int \*secretKeySize, char \*\*issuer,

char \*\*subject);

extern SECStatus SSL\_SetPKCS11PinArg(PRFileDesc \* fd, void \*a);

extern SECStatus SSL\_SetSockPeerID(PRFileDesc \* fd, const char \*peerID);

extern SECStatus SSL\_SetURL(PRFileDesc \* fd, const char \*url);

### **24.6.3 nss3/sslerr.h**

#define \_\_SSL\_ERR\_H\_

#define IS\_SSL\_ERROR(code) \

(((code) >= SSL\_ERROR\_BASE) && ((code) < SSL\_ERROR\_LIMIT))

#define SSL\_ERROR\_BASE (-0x3000)

#define SSL\_ERROR\_LIMIT (SSL\_ERROR\_BASE + 1000)

typedef enum {

SSL\_ERROR\_EXPORT\_ONLY\_SERVER = (SSL\_ERROR\_BASE + 0),

SSL\_ERROR\_US\_ONLY\_SERVER = (SSL\_ERROR\_BASE + 1),

SSL\_ERROR\_NO\_CYPHER\_OVERLAP = (SSL\_ERROR\_BASE + 2),

SSL\_ERROR\_NO\_CERTIFICATE = (SSL\_ERROR\_BASE + 3),

SSL\_ERROR\_BAD\_CERTIFICATE = (SSL\_ERROR\_BASE + 4),

SSL\_ERROR\_BAD\_CLIENT = (SSL\_ERROR\_BASE + 6),

SSL\_ERROR\_BAD\_SERVER = (SSL\_ERROR\_BASE + 7),

SSL\_ERROR\_UNSUPPORTED\_CERTIFICATE\_TYPE = (SSL\_ERROR\_BASE + 8),

SSL\_ERROR\_UNSUPPORTED\_VERSION = (SSL\_ERROR\_BASE + 9),

SSL\_ERROR\_WRONG\_CERTIFICATE = (SSL\_ERROR\_BASE + 11),

SSL\_ERROR\_BAD\_CERT\_DOMAIN = (SSL\_ERROR\_BASE + 12),

SSL\_ERROR\_POST\_WARNING = (SSL\_ERROR\_BASE + 13),

SSL\_ERROR\_SSL2\_DISABLED = (SSL\_ERROR\_BASE + 14),

SSL\_ERROR\_BAD\_MAC\_READ = (SSL\_ERROR\_BASE + 15),

SSL\_ERROR\_BAD\_MAC\_ALERT = (SSL\_ERROR\_BASE + 16),

SSL\_ERROR\_BAD\_CERT\_ALERT = (SSL\_ERROR\_BASE + 17),

SSL\_ERROR\_REVOKED\_CERT\_ALERT = (SSL\_ERROR\_BASE + 18),

SSL\_ERROR\_EXPIRED\_CERT\_ALERT = (SSL\_ERROR\_BASE + 19),

SSL\_ERROR\_SSL\_DISABLED = (SSL\_ERROR\_BASE + 20),

SSL\_ERROR\_FORTEZZA\_PQG = (SSL\_ERROR\_BASE + 21),

SSL\_ERROR\_UNKNOWN\_CIPHER\_SUITE = (SSL\_ERROR\_BASE + 22),

SSL\_ERROR\_NO\_CIPHERS\_SUPPORTED = (SSL\_ERROR\_BASE + 23),

SSL\_ERROR\_BAD\_BLOCK\_PADDING = (SSL\_ERROR\_BASE + 24),

SSL\_ERROR\_RX\_RECORD\_TOO\_LONG = (SSL\_ERROR\_BASE + 25),

SSL\_ERROR\_TX\_RECORD\_TOO\_LONG = (SSL\_ERROR\_BASE + 26),

SSL\_ERROR\_RX\_MALFORMED\_HELLO\_REQUEST = (SSL\_ERROR\_BASE + 27),

SSL\_ERROR\_RX\_MALFORMED\_CLIENT\_HELLO = (SSL\_ERROR\_BASE + 28),

SSL\_ERROR\_RX\_MALFORMED\_SERVER\_HELLO = (SSL\_ERROR\_BASE + 29),

SSL\_ERROR\_RX\_MALFORMED\_CERTIFICATE = (SSL\_ERROR\_BASE + 30),

SSL\_ERROR\_RX\_MALFORMED\_SERVER\_KEY\_EXCH = (SSL\_ERROR\_BASE + 31),

SSL\_ERROR\_RX\_MALFORMED\_CERT\_REQUEST = (SSL\_ERROR\_BASE + 32),

SSL\_ERROR\_RX\_MALFORMED\_HELLO\_DONE = (SSL\_ERROR\_BASE + 33),

SSL\_ERROR\_RX\_MALFORMED\_CERT\_VERIFY = (SSL\_ERROR\_BASE + 34),

SSL\_ERROR\_RX\_MALFORMED\_CLIENT\_KEY\_EXCH = (SSL\_ERROR\_BASE + 35),

SSL\_ERROR\_RX\_MALFORMED\_FINISHED = (SSL\_ERROR\_BASE + 36),

SSL\_ERROR\_RX\_MALFORMED\_CHANGE\_CIPHER = (SSL\_ERROR\_BASE + 37),

SSL\_ERROR\_RX\_MALFORMED\_ALERT = (SSL\_ERROR\_BASE + 38),

SSL\_ERROR\_RX\_MALFORMED\_HANDSHAKE = (SSL\_ERROR\_BASE + 39),

SSL\_ERROR\_RX\_MALFORMED\_APPLICATION\_DATA = (SSL\_ERROR\_BASE + 40),

SSL\_ERROR\_RX\_UNEXPECTED\_HELLO\_REQUEST = (SSL\_ERROR\_BASE + 41),

SSL\_ERROR\_RX\_UNEXPECTED\_CLIENT\_HELLO = (SSL\_ERROR\_BASE + 42),

SSL\_ERROR\_RX\_UNEXPECTED\_SERVER\_HELLO = (SSL\_ERROR\_BASE + 43),

SSL\_ERROR\_RX\_UNEXPECTED\_CERTIFICATE = (SSL\_ERROR\_BASE + 44),

SSL\_ERROR\_RX\_UNEXPECTED\_SERVER\_KEY\_EXCH = (SSL\_ERROR\_BASE + 45),

SSL\_ERROR\_RX\_UNEXPECTED\_CERT\_REQUEST = (SSL\_ERROR\_BASE + 46),

SSL\_ERROR\_RX\_UNEXPECTED\_HELLO\_DONE = (SSL\_ERROR\_BASE + 47),

SSL\_ERROR\_RX\_UNEXPECTED\_CERT\_VERIFY = (SSL\_ERROR\_BASE + 48),

SSL\_ERROR\_RX\_UNEXPECTED\_CLIENT\_KEY\_EXCH = (SSL\_ERROR\_BASE + 49),

SSL\_ERROR\_RX\_UNEXPECTED\_FINISHED = (SSL\_ERROR\_BASE + 50),

SSL\_ERROR\_RX\_UNEXPECTED\_CHANGE\_CIPHER = (SSL\_ERROR\_BASE + 51),

SSL\_ERROR\_RX\_UNEXPECTED\_ALERT = (SSL\_ERROR\_BASE + 52),

SSL\_ERROR\_RX\_UNEXPECTED\_HANDSHAKE = (SSL\_ERROR\_BASE + 53),

SSL\_ERROR\_RX\_UNEXPECTED\_APPLICATION\_DATA = (SSL\_ERROR\_BASE + 54),

SSL\_ERROR\_RX\_UNKNOWN\_RECORD\_TYPE = (SSL\_ERROR\_BASE + 55),

SSL\_ERROR\_RX\_UNKNOWN\_HANDSHAKE = (SSL\_ERROR\_BASE + 56),

SSL\_ERROR\_RX\_UNKNOWN\_ALERT = (SSL\_ERROR\_BASE + 57),

SSL\_ERROR\_CLOSE\_NOTIFY\_ALERT = (SSL\_ERROR\_BASE + 58),

SSL\_ERROR\_HANDSHAKE\_UNEXPECTED\_ALERT = (SSL\_ERROR\_BASE + 59),

SSL\_ERROR\_DECOMPRESSION\_FAILURE\_ALERT = (SSL\_ERROR\_BASE + 60),

SSL\_ERROR\_HANDSHAKE\_FAILURE\_ALERT = (SSL\_ERROR\_BASE + 61),

SSL\_ERROR\_ILLEGAL\_PARAMETER\_ALERT = (SSL\_ERROR\_BASE + 62),

SSL\_ERROR\_UNSUPPORTED\_CERT\_ALERT = (SSL\_ERROR\_BASE + 63),

SSL\_ERROR\_CERTIFICATE\_UNKNOWN\_ALERT = (SSL\_ERROR\_BASE + 64),

SSL\_ERROR\_GENERATE\_RANDOM\_FAILURE = (SSL\_ERROR\_BASE + 65),

SSL\_ERROR\_SIGN\_HASHES\_FAILURE = (SSL\_ERROR\_BASE + 66),

SSL\_ERROR\_EXTRACT\_PUBLIC\_KEY\_FAILURE = (SSL\_ERROR\_BASE + 67),

SSL\_ERROR\_SERVER\_KEY\_EXCHANGE\_FAILURE = (SSL\_ERROR\_BASE + 68),

SSL\_ERROR\_CLIENT\_KEY\_EXCHANGE\_FAILURE = (SSL\_ERROR\_BASE + 69),

SSL\_ERROR\_ENCRYPTION\_FAILURE = (SSL\_ERROR\_BASE + 70),

SSL\_ERROR\_DECRYPTION\_FAILURE = (SSL\_ERROR\_BASE + 71),

SSL\_ERROR\_SOCKET\_WRITE\_FAILURE = (SSL\_ERROR\_BASE + 72),

SSL\_ERROR\_MD5\_DIGEST\_FAILURE = (SSL\_ERROR\_BASE + 73),

SSL\_ERROR\_SHA\_DIGEST\_FAILURE = (SSL\_ERROR\_BASE + 74),

SSL\_ERROR\_MAC\_COMPUTATION\_FAILURE = (SSL\_ERROR\_BASE + 75),

SSL\_ERROR\_SYM\_KEY\_CONTEXT\_FAILURE = (SSL\_ERROR\_BASE + 76),

SSL\_ERROR\_SYM\_KEY\_UNWRAP\_FAILURE = (SSL\_ERROR\_BASE + 77),

SSL\_ERROR\_PUB\_KEY\_SIZE\_LIMIT\_EXCEEDED = (SSL\_ERROR\_BASE + 78),

SSL\_ERROR\_IV\_PARAM\_FAILURE = (SSL\_ERROR\_BASE + 79),

SSL\_ERROR\_INIT\_CIPHER\_SUITE\_FAILURE = (SSL\_ERROR\_BASE + 80),

SSL\_ERROR\_SESSION\_KEY\_GEN\_FAILURE = (SSL\_ERROR\_BASE + 81),

SSL\_ERROR\_NO\_SERVER\_KEY\_FOR\_ALG = (SSL\_ERROR\_BASE + 82),

SSL\_ERROR\_TOKEN\_INSERTION\_REMOVAL = (SSL\_ERROR\_BASE + 83),

SSL\_ERROR\_TOKEN\_SLOT\_NOT\_FOUND = (SSL\_ERROR\_BASE + 84),

SSL\_ERROR\_NO\_COMPRESSION\_OVERLAP = (SSL\_ERROR\_BASE + 85),

SSL\_ERROR\_HANDSHAKE\_NOT\_COMPLETED = (SSL\_ERROR\_BASE + 86),

SSL\_ERROR\_BAD\_HANDSHAKE\_HASH\_VALUE = (SSL\_ERROR\_BASE + 87),

SSL\_ERROR\_CERT\_KEA\_MISMATCH = (SSL\_ERROR\_BASE + 88),

SSL\_ERROR\_NO\_TRUSTED\_SSL\_CLIENT\_CA = (SSL\_ERROR\_BASE + 89),

SSL\_ERROR\_SESSION\_NOT\_FOUND = (SSL\_ERROR\_BASE + 90),

SSL\_ERROR\_DECRYPTION\_FAILED\_ALERT = (SSL\_ERROR\_BASE + 91),

SSL\_ERROR\_RECORD\_OVERFLOW\_ALERT = (SSL\_ERROR\_BASE + 92),

SSL\_ERROR\_UNKNOWN\_CA\_ALERT = (SSL\_ERROR\_BASE + 93),

SSL\_ERROR\_ACCESS\_DENIED\_ALERT = (SSL\_ERROR\_BASE + 94),

SSL\_ERROR\_DECODE\_ERROR\_ALERT = (SSL\_ERROR\_BASE + 95),

SSL\_ERROR\_DECRYPT\_ERROR\_ALERT = (SSL\_ERROR\_BASE + 96),

SSL\_ERROR\_EXPORT\_RESTRICTION\_ALERT = (SSL\_ERROR\_BASE + 97),

SSL\_ERROR\_PROTOCOL\_VERSION\_ALERT = (SSL\_ERROR\_BASE + 98),

SSL\_ERROR\_INSUFFICIENT\_SECURITY\_ALERT = (SSL\_ERROR\_BASE + 99),

SSL\_ERROR\_INTERNAL\_ERROR\_ALERT = (SSL\_ERROR\_BASE + 100),

SSL\_ERROR\_USER\_CANCELED\_ALERT = (SSL\_ERROR\_BASE + 101),

SSL\_ERROR\_NO\_RENEGOTIATION\_ALERT = (SSL\_ERROR\_BASE + 102),

SSL\_ERROR\_SERVER\_CACHE\_NOT\_CONFIGURED = (SSL\_ERROR\_BASE + 103),

SSL\_ERROR\_UNSUPPORTED\_EXTENSION\_ALERT = (SSL\_ERROR\_BASE + 104),

SSL\_ERROR\_CERTIFICATE\_UNOBTAINABLE\_ALERT = (SSL\_ERROR\_BASE + 105),

SSL\_ERROR\_UNRECOGNIZED\_NAME\_ALERT = (SSL\_ERROR\_BASE + 106),

SSL\_ERROR\_BAD\_CERT\_STATUS\_RESPONSE\_ALERT = (SSL\_ERROR\_BASE + 107),

SSL\_ERROR\_BAD\_CERT\_HASH\_VALUE\_ALERT = (SSL\_ERROR\_BASE + 108)

} SSLErrorCodes;

### **24.6.4 nss3/sslproto.h**

#define \_\_sslproto\_h\_

#define SSL\_MT\_ERROR 0

#define SSL\_NULL\_WITH\_NULL\_NULL 0x0000

#define SSL\_PE\_NO\_CYPHERS 0x0001

#define SSL\_RSA\_WITH\_NULL\_MD5 0x0001

#define SSL\_LIBRARY\_VERSION\_2 0x0002

#define SSL\_PE\_NO\_CERTIFICATE 0x0002

#define SSL\_RSA\_WITH\_NULL\_SHA 0x0002

#define SSL\_RSA\_EXPORT\_WITH\_RC4\_40\_MD5 0x0003

#define SSL\_PE\_BAD\_CERTIFICATE 0x0004

#define SSL\_RSA\_WITH\_RC4\_128\_MD5 0x0004

#define SSL\_RSA\_WITH\_RC4\_128\_SHA 0x0005

#define SSL\_PE\_UNSUPPORTED\_CERTIFICATE\_TYPE 0x0006

#define SSL\_RSA\_EXPORT\_WITH\_RC2\_CBC\_40\_MD5 0x0006

#define SSL\_RSA\_WITH\_IDEA\_CBC\_SHA 0x0007

#define SSL\_RSA\_EXPORT\_WITH\_DES40\_CBC\_SHA 0x0008

#define SSL\_RSA\_WITH\_DES\_CBC\_SHA 0x0009

#define SSL\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA 0x000a

#define SSL\_DH\_DSS\_EXPORT\_WITH\_DES40\_CBC\_SHA 0x000b

#define SSL\_DH\_DSS\_WITH\_DES\_CBC\_SHA 0x000c

#define SSL\_DH\_DSS\_WITH\_3DES\_EDE\_CBC\_SHA 0x000d

#define SSL\_DH\_RSA\_EXPORT\_WITH\_DES40\_CBC\_SHA 0x000e

#define SSL\_DH\_RSA\_WITH\_DES\_CBC\_SHA 0x000f

#define SSL\_DH\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA 0x0010

#define SSL\_DHE\_DSS\_EXPORT\_WITH\_DES40\_CBC\_SHA 0x0011

#define SSL\_DHE\_DSS\_WITH\_DES\_CBC\_SHA 0x0012

#define SSL\_DHE\_DSS\_WITH\_3DES\_EDE\_CBC\_SHA 0x0013

#define SSL\_DHE\_RSA\_EXPORT\_WITH\_DES40\_CBC\_SHA 0x0014

#define SSL\_DHE\_RSA\_WITH\_DES\_CBC\_SHA 0x0015

#define SSL\_DHE\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA 0x0016

#define SSL\_DH\_ANON\_EXPORT\_WITH\_RC4\_40\_MD5 0x0017

#define SSL\_DH\_ANON\_WITH\_RC4\_128\_MD5 0x0018

#define SSL\_DH\_ANON\_EXPORT\_WITH\_DES40\_CBC\_SHA 0x0019

#define SSL\_DH\_ANON\_WITH\_DES\_CBC\_SHA 0x001a

#define SSL\_DH\_ANON\_WITH\_3DES\_EDE\_CBC\_SHA 0x001b

#define SSL\_FORTEZZA\_DMS\_WITH\_NULL\_SHA 0x001c

#define SSL\_FORTEZZA\_DMS\_WITH\_FORTEZZA\_CBC\_SHA 0x001d

#define SSL\_FORTEZZA\_DMS\_WITH\_RC4\_128\_SHA 0x001e

#define TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA 0x002F

#define TLS\_DH\_DSS\_WITH\_AES\_128\_CBC\_SHA 0x0030

#define TLS\_DH\_RSA\_WITH\_AES\_128\_CBC\_SHA 0x0031

#define TLS\_DHE\_DSS\_WITH\_AES\_128\_CBC\_SHA 0x0032

#define TLS\_DHE\_RSA\_WITH\_AES\_128\_CBC\_SHA 0x0033

#define TLS\_DH\_ANON\_WITH\_AES\_128\_CBC\_SHA 0x0034

#define TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA 0x0035

#define TLS\_DH\_DSS\_WITH\_AES\_256\_CBC\_SHA 0x0036

#define TLS\_DH\_RSA\_WITH\_AES\_256\_CBC\_SHA 0x0037

#define TLS\_DHE\_DSS\_WITH\_AES\_256\_CBC\_SHA 0x0038

#define TLS\_DHE\_RSA\_WITH\_AES\_256\_CBC\_SHA 0x0039

#define TLS\_DH\_ANON\_WITH\_AES\_256\_CBC\_SHA 0x003A

#define TLS\_RSA\_EXPORT1024\_WITH\_DES\_CBC\_SHA 0x0062

#define TLS\_DHE\_DSS\_EXPORT1024\_WITH\_DES\_CBC\_SHA 0x0063

#define TLS\_RSA\_EXPORT1024\_WITH\_RC4\_56\_SHA 0x0064

#define TLS\_DHE\_DSS\_EXPORT1024\_WITH\_RC4\_56\_SHA 0x0065

#define TLS\_DHE\_DSS\_WITH\_RC4\_128\_SHA 0x0066

#define SSL\_AT\_MD5\_WITH\_RSA\_ENCRYPTION 0x01

#define SSL\_CK\_RC4\_128\_WITH\_MD5 0x01

#define SSL\_CT\_X509\_CERTIFICATE 0x01

#define SSL\_CK\_RC4\_128\_EXPORT40\_WITH\_MD5 0x02

#define SSL\_CK\_RC2\_128\_CBC\_WITH\_MD5 0x03

#define SSL\_LIBRARY\_VERSION\_3\_0 0x0300

#define SSL\_LIBRARY\_VERSION\_3\_1\_TLS 0x0301

#define SSL\_CK\_RC2\_128\_CBC\_EXPORT40\_WITH\_MD5 0x04

#define SSL\_CK\_IDEA\_128\_CBC\_WITH\_MD5 0x05

#define SSL\_CK\_DES\_64\_CBC\_WITH\_MD5 0x06

#define SSL\_CK\_DES\_192\_EDE3\_CBC\_WITH\_MD5 0x07

#define TLS\_ECDH\_ECDSA\_WITH\_NULL\_SHA 0xC001

#define TLS\_ECDH\_ECDSA\_WITH\_RC4\_128\_SHA 0xC002

#define TLS\_ECDH\_ECDSA\_WITH\_3DES\_EDE\_CBC\_SHA 0xC003

#define TLS\_ECDH\_ECDSA\_WITH\_AES\_128\_CBC\_SHA 0xC004

#define TLS\_ECDH\_ECDSA\_WITH\_AES\_256\_CBC\_SHA 0xC005

#define TLS\_ECDHE\_ECDSA\_WITH\_NULL\_SHA 0xC006

#define TLS\_ECDHE\_ECDSA\_WITH\_RC4\_128\_SHA 0xC007

#define TLS\_ECDHE\_ECDSA\_WITH\_3DES\_EDE\_CBC\_SHA 0xC008

#define TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA 0xC009

#define TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CBC\_SHA 0xC00A

#define TLS\_ECDH\_RSA\_WITH\_NULL\_SHA 0xC00B

#define TLS\_ECDH\_RSA\_WITH\_RC4\_128\_SHA 0xC00C

#define TLS\_ECDH\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA 0xC00D

#define TLS\_ECDH\_RSA\_WITH\_AES\_128\_CBC\_SHA 0xC00E

#define TLS\_ECDH\_RSA\_WITH\_AES\_256\_CBC\_SHA 0xC00F

#define TLS\_ECDHE\_RSA\_WITH\_NULL\_SHA 0xC010

#define TLS\_ECDHE\_RSA\_WITH\_RC4\_128\_SHA 0xC011

#define TLS\_ECDHE\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA 0xC012

#define TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA 0xC013

#define TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA 0xC014

#define TLS\_ECDH\_anon\_WITH\_NULL\_SHA 0xC015

#define TLS\_ECDH\_anon\_WITH\_RC4\_128\_SHA 0xC016

#define TLS\_ECDH\_anon\_WITH\_3DES\_EDE\_CBC\_SHA 0xC017

#define TLS\_ECDH\_anon\_WITH\_AES\_128\_CBC\_SHA 0xC018

#define TLS\_ECDH\_anon\_WITH\_AES\_256\_CBC\_SHA 0xC019

#define SSL\_RSA\_FIPS\_WITH\_DES\_CBC\_SHA 0xfefe

#define SSL\_RSA\_FIPS\_WITH\_3DES\_EDE\_CBC\_SHA 0xfeff

#define SSL\_RSA\_OLDFIPS\_WITH\_3DES\_EDE\_CBC\_SHA 0xffe0

#define SSL\_RSA\_OLDFIPS\_WITH\_DES\_CBC\_SHA 0xffe1

#define SSL\_HL\_CLIENT\_FINISHED\_HBYTES 1

#define SSL\_HL\_SERVER\_FINISHED\_HBYTES 1

#define SSL\_HL\_SERVER\_VERIFY\_HBYTES 1

#define SSL\_MT\_CLIENT\_HELLO 1

#define SSL\_HL\_CLIENT\_MASTER\_KEY\_HBYTES 10

#define SSL\_HL\_SERVER\_HELLO\_HBYTES 11

#define SSL\_HL\_REQUEST\_CERTIFICATE\_HBYTES 2

#define SSL\_MT\_CLIENT\_MASTER\_KEY 2

#define SSL\_HL\_ERROR\_HBYTES 3

#define SSL\_MT\_CLIENT\_FINISHED 3

#define SSL\_MT\_SERVER\_HELLO 4

#define SSL\_MT\_SERVER\_VERIFY 5

#define SSL\_HL\_CLIENT\_CERTIFICATE\_HBYTES 6

#define SSL\_MT\_SERVER\_FINISHED 6

#define SSL\_MT\_REQUEST\_CERTIFICATE 7

#define SSL\_MT\_CLIENT\_CERTIFICATE 8

#define SSL\_HL\_CLIENT\_HELLO\_HBYTES 9

### **24.6.5 nss3/sslt.h**

#define \_\_sslt\_h\_

typedef enum {

ssl\_kea\_null,

ssl\_kea\_rsa = 1,

ssl\_kea\_dh = 2,

ssl\_kea\_fortezza = 3,

ssl\_kea\_ecdh = 4,

ssl\_kea\_size = 5

} SSLKEAType;

typedef enum {

ssl\_sign\_null,

ssl\_sign\_rsa = 1,

ssl\_sign\_dsa = 2,

ssl\_sign\_ecdsa = 3

} SSLSignType;

typedef enum {

ssl\_auth\_null,

ssl\_auth\_rsa = 1,

ssl\_auth\_dsa = 2,

ssl\_auth\_kea = 3,

ssl\_auth\_ecdsa = 4

} SSLAuthType;

typedef enum {

ssl\_calg\_null,

ssl\_calg\_rc4 = 1,

ssl\_calg\_rc2 = 2,

ssl\_calg\_des = 3,

ssl\_calg\_3des = 4,

ssl\_calg\_idea = 5,

ssl\_calg\_fortezza = 6,

ssl\_calg\_aes = 7,

ssl\_calg\_camellia = 8

} SSLCipherAlgorithm;

typedef enum {

ssl\_mac\_null,

ssl\_mac\_md5 = 1,

ssl\_mac\_sha = 2,

ssl\_hmac\_md5 = 3,

ssl\_hmac\_sha = 4

} SSLMACAlgorithm;

typedef struct SSLChannelInfoStr {

PRUint32 length;

PRUint16 protocolVersion;

PRUint16 cipherSuite;

PRUint32 authKeyBits;

PRUint32 keaKeyBits;

PRUint32 creationTime;

PRUint32 lastAccessTime;

PRUint32 expirationTime;

PRUint32 sessionIDLength;

PRUint8 sessionID[31];

} SSLChannelInfo;

typedef struct SSLCipherSuiteInfoStr {

PRUint16 length;

PRUint16 cipherSuite;

const char \*cipherSuiteName;

const char \*authAlgorithmName;

SSLAuthType authAlgorithm;

const char \*keaTypeName;

SSLKEAType keaType;

const char \*symCipherName;

SSLCipherAlgorithm symCipher;

PRUint16 symKeyBits;

PRUint16 symKeySpace;

PRUint16 effectiveKeyBits;

const char \*macAlgorithmName;

SSLMACAlgorithm macAlgorithm;

PRUint16 macBits;

PRUintn isFIPS:1;

PRUintn isExportable:1;

PRUintn nonStandard:1;

PRUintn reservedBits:29;

} SSLCipherSuiteInfo;

# **XI Package Format and Installation**

# **25 Software Installation**

## **25.1 Introduction**

Applications shall either be packaged in the RPM packaging format as defined in this specification, or supply an installer which is LSB conforming (for example, calls LSB commands and utilities).

**Note:** Supplying an RPM format package is encouraged because it makes systems easier to manage. This specification does not require the implementation to use RPM as the package manager; it only specifies the format of the package file and requires that implementations must have some method of installing conforming packages.

Applications are also encouraged to uninstall cleanly.

A package in the RPM format may include a dependency on the LSB Core and other LSB specifications, as described in [Section 25.6](#ID_PKGDEPEND). Packages that are not in the RPM format may test for the presence of a conforming implementation by means of the **lsb\_release** utility.

Implementations shall provide a mechanism for installing applications in the RPM packaging format with some restrictions listed below.

**Note:** The implementation itself may use a different packaging format for its own packages, and may use any available mechanism for installing conforming packages, including translation into a different format.

## **25.2 Package File Format**

An RPM format file consists of 4 sections, the Lead, Signature, Header, and the Payload. All values are stored in network byte order.

**Table 25-1 RPM File Format**

|  |
| --- |
| Lead |
| Signature |
| Header |
| Payload |

These 4 sections shall exist in the order specified.

The lead section is used to identify the package file.

The signature section is used to verify the integrity, and optionally, the authenticity of the majority of the package file.

The header section contains all available information about the package. Entries such as the package's name, version, and file list, are contained in the header.

The payload section holds the files to be installed.

### **25.2.1 Lead Section**

struct rpmlead {

unsigned char magic[4];

unsigned char major, minor;

short type;

short archnum;

char name[66];

short osnum;

short signature\_type;

char reserved[16];

} ;

*magic*

  Value identifying this file as an RPM format file. This value shall be "\355\253\356\333".

*major*

  Value indicating the major version number of the file format version. This value shall be 3.

*minor*

  Value indicating the minor revision number of file format version. This value shall be 0.

*type*

  Value indicating whether this is a source or binary package. This value shall be 0 to indicate a binary package.

*archnum*

  Value indicating the architecture for which this package is valid. This value is specified in the relevant architecture specific part of the LSB Core Specification.

*name*

  A NUL terminated string that provides the package name. This name shall conform with the Package Naming Conventions section of this specification.

*osnum*

  Value indicating the Operating System for which this package is valid. This value shall be 1.

*signature\_type*

  Value indicating the type of the signature used in the Signature part of the file. This value shall be 5.

*reserved*

  Reserved space. The value is undefined.

### **25.2.2 Header Structure**

The Header structure is used for both the Signature and Header Sections. A Header Structure consists of 3 parts, a Header record, followed by 1 or more Index records, followed by 0 or more bytes of data associated with the Index records. A Header structure shall be aligned to an 8 byte boundary.

**Table 25-2 Signature Format**

|  |
| --- |
| Header Record |
| Array of Index Records |
| Store of Index Values |

#### 25.2.2.1 Header Record

struct rpmheader {

unsigned char magic[4];

unsigned char reserved[4];

int nindex;

int hsize;

};

*magic*

  Value identifying this record as an RPM header record. This value shall be "\216\255\350\001".

*reserved*

  Reserved space. This value shall be "\000\000\000\000".

*nindex*

  The number of Index Records that follow this Header Record. There should be at least 1 Index Record.

*hsize*

  The size in bytes of the storage area for the data pointed to by the Index Records.

#### 25.2.2.2 Index Record

struct rpmhdrindex {

int tag;

int type;

int offset;

int count;

};

*tag*

  Value identifying the purpose of the data associated with this Index Record. The value of this field is dependent on the context in which the Index Record is used, and is defined below and in later sections.

*type*

  Value identifying the type of the data associated with this Index Record. The possible *type* values are defined below.

*offset*

  Location in the Store of the data associated with this Index Record. This value should between 0 and the value contained in the *hsize* of the Header Structure.

*count*

  Size of the data associated with this Index Record. The *count* is the number of elements whose size is defined by the type of this Record.

##### *25.2.2.2.1 Index Type Values*

The possible values for the *type* field are defined in this table.

**Table 25-3 Index Type values**

| **Type** | **Value** | **Size (in bytes)** | **Alignment** |
| --- | --- | --- | --- |
| RPM\_NULL\_TYPE | 0 | Not Implemented. |  |
| RPM\_CHAR\_TYPE | 1 | 1 | 1 |
| RPM\_INT8\_TYPE | 2 | 1 | 1 |
| RPM\_INT16\_TYPE | 3 | 2 | 2 |
| RPM\_INT32\_TYPE | 4 | 4 | 4 |
| RPM\_INT64\_TYPE | 5 | Reserved. |  |
| RPM\_STRING\_TYPE | 6 | variable, NUL terminated | 1 |
| RPM\_BIN\_TYPE | 7 | 1 | 1 |
| RPM\_STRING\_ARRAY\_TYPE | 8 | Variable, sequence of NUL terminated strings | 1 |
| RPM\_I18NSTRING\_TYPE | 9 | variable, sequence of NUL terminated strings | 1 |

The string arrays specified for entries of type RPM\_STRING\_ARRAY\_TYPE and RPM\_I18NSTRING\_TYPE are vectors of strings in a contiguous block of memory, each element separated from its neighbors by a NUL character.

Index records with type RPM\_I18NSTRING\_TYPE shall always have a *count* of 1. The array entries in an index of type RPM\_I18NSTRING\_TYPE correspond to the locale names contained in the RPMTAG\_HDRI18NTABLE index.

##### *25.2.2.2.2 Index Tag Values*

Some values are designated as header private, and may appear in any header structure. These are defined here. Additional values are defined in later sections.

**Table 25-4 Header Private Tag Values**

| **Name** | **Tag Value** | **Type** | **Count** | **Status** |
| --- | --- | --- | --- | --- |
| RPMTAG\_HEADERSIGNATURES | 62 | BIN | 16 | Optional |
| RPMTAG\_HEADERIMMUTABLE | 63 | BIN | 16 | Optional |
| RPMTAG\_HEADERI18NTABLE | 100 | STRING\_ARRAY |  | Optional |

RPMTAG\_HEADERSIGNATURES

  The signature tag differentiates a signature header from a metadata header, and identifies the original contents of the signature header.

RPMTAG\_HEADERIMMUTABLE

  This tag contains an index record which specifies the portion of the Header Record which was used for the calculation of a signature. This data shall be preserved or any header-only signature will be invalidated.

RPMTAG\_HEADERI18NTABLE

  Contains a list of locales for which strings are provided in other parts of the package.

Not all Index records defined here will be present in all packages. Each tag value has a status which is defined here.

Required

  This Index Record shall be present.

Optional

  This Index Record may be present.

Informational

  This Index Record may be present, but does not contribute to the processing of the package.

Deprecated

  This Index Record should not be present.

Obsolete

  This Index Record shall not be present.

Reserved

  This Index Record shall not be present.

#### 25.2.2.3 Header Store

The header store contains the values specified by the Index structures. These values are aligned according to their type and padding is used if needed. The store is located immediately following the Index structures.

### **25.2.3 Signature Section**

The Signature section is implemented using the Header structure. The signature section defines the following additional tag values which may be used in the Index structures.

These values exist to provide additional information about the rest of the package.

**Table 25-5 Signature Tag Values**

| **Name** | **Tag Value** | **Type** | **Count** | **Status** |
| --- | --- | --- | --- | --- |
| RPMSIGTAG\_SIZE | 1000 | INT32 | 1 | Required |
| RPMSIGTAG\_PAYLOADSIZE | 1007 | INT32 | 1 | Optional |

RPMSIGTAG\_SIZE

  This tag specifies the combined size of the Header and Payload sections.

RPMSIGTAG\_PAYLOADSIZE

  This tag specifies the uncompressed size of the Payload archive, including the cpio headers.

These values exist to ensure the integrity of the rest of the package.

**Table 25-6 Signature Digest Tag Values**

| **Name** | **Tag Value** | **Type** | **Count** | **Status** |
| --- | --- | --- | --- | --- |
| RPMSIGTAG\_SHA1 | 269 | STRING | 1 | Optional |
| RPMSIGTAG\_MD5 | 1004 | BIN | 16 | Required |

RPMSIGTAG\_SHA1

  This index contains the SHA1 checksum of the entire Header Section, including the Header Record, Index Records and Header store.

RPMSIGTAG\_MD5

  This tag specifies the 128-bit MD5 checksum of the combined Header and Archive sections.

These values exist to provide authentication of the package.

**Table 25-7 Signature Signing Tag Values**

| **Name** | **Tag Value** | **Type** | **Count** | **Status** |
| --- | --- | --- | --- | --- |
| RPMSIGTAG\_DSA | 267 | BIN | 65 | Optional |
| RPMSIGTAG\_RSA | 268 | BIN | 1 | Optional |
| RPMSIGTAG\_PGP | 1002 | BIN | 1 | Optional |
| RPMSIGTAG\_GPG | 1005 | BIN | 65 | Optional |

RPMSIGTAG\_DSA

  The tag contains the DSA signature of the Header section. The data is formatted as a Version 3 Signature Packet as specified in [RFC 2440: OpenPGP Message Format](#ID_STD_46_RFC2440). If this tag is present, then the SIGTAG\_GPG tag shall also be present.

RPMSIGTAG\_RSA

  The tag contains the RSA signature of the Header section.The data is formatted as a Version 3 Signature Packet as specified in [RFC 2440: OpenPGP Message Format](#ID_STD_46_RFC2440). If this tag is present, then the SIGTAG\_PGP shall also be present.

RPMSIGTAG\_PGP

  This tag specifies the RSA signature of the combined Header and Payload sections. The data is formatted as a Version 3 Signature Packet as specified in [RFC 2440: OpenPGP Message Format](#ID_STD_46_RFC2440).

RPMSIGTAG\_GPG

  The tag contains the DSA signature of the combined Header and Payload sections. The data is formatted as a Version 3 Signature Packet as specified in [RFC 2440: OpenPGP Message Format](#ID_STD_46_RFC2440).

### **25.2.4 Header Section**

The Header section is implemented using the Header structure. The Header section defines the following additional tag values which may be used in the Index structures.

#### 25.2.4.1 Package Information

The following tag values are used to indicate information that describes the package as a whole.

**Table 25-8 Package Info Tag Values**

| **Name** | **Tag Value** | **Type** | **Count** | **Status** |
| --- | --- | --- | --- | --- |
| RPMTAG\_NAME | 1000 | STRING | 1 | Required |
| RPMTAG\_VERSION | 1001 | STRING | 1 | Required |
| RPMTAG\_RELEASE | 1002 | STRING | 1 | Required |
| RPMTAG\_SUMMARY | 1004 | I18NSTRING | 1 | Required |
| RPMTAG\_DESCRIPTION | 1005 | I18NSTRING | 1 | Required |
| RPMTAG\_SIZE | 1009 | INT32 | 1 | Required |
| RPMTAG\_DISTRIBUTION | 1010 | STRING | 1 | Informational |
| RPMTAG\_VENDOR | 1011 | STRING | 1 | Informational |
| RPMTAG\_LICENSE | 1014 | STRING | 1 | Required |
| RPMTAG\_PACKAGER | 1015 | STRING | 1 | Informational |
| RPMTAG\_GROUP | 1016 | I18NSTRING | 1 | Required |
| RPMTAG\_URL | 1020 | STRING | 1 | Informational |
| RPMTAG\_OS | 1021 | STRING | 1 | Required |
| RPMTAG\_ARCH | 1022 | STRING | 1 | Required |
| RPMTAG\_SOURCERPM | 1044 | STRING | 1 | Informational |
| RPMTAG\_ARCHIVESIZE | 1046 | INT32 | 1 | Optional |
| RPMTAG\_RPMVERSION | 1064 | STRING | 1 | Informational |
| RPMTAG\_COOKIE | 1094 | STRING | 1 | Optional |
| RPMTAG\_DISTURL | 1123 | STRING | 1 | Informational |
| RPMTAG\_PAYLOADFORMAT | 1124 | STRING | 1 | Required |
| RPMTAG\_PAYLOADCOMPRESSOR | 1125 | STRING | 1 | Required |
| RPMTAG\_PAYLOADFLAGS | 1126 | STRING | 1 | Required |

RPMTAG\_NAME

  This tag specifies the name of the package.

RPMTAG\_VERSION

  This tag specifies the version of the package.

RPMTAG\_RELEASE

  This tag specifies the release of the package.

RPMTAG\_SUMMARY

  This tag specifies the summary description of the package. The summary value pointed to by this index record contains a one line description of the package.

RPMTAG\_DESCRIPTION

  This tag specifies the description of the package. The description value pointed to by this index record contains a full desription of the package.

RPMTAG\_SIZE

  This tag specifies the sum of the sizes of the regular files in the archive.

RPMTAG\_DISTRIBUTION

  A string containing the name of the distribution on which the package was built.

RPMTAG\_VENDOR

  A string containing the name of the organization that produced the package.

RPMTAG\_LICENSE

  This tag specifies the license which applies to this package.

RPMTAG\_PACKAGER

  A string identifying the tool used to build the package.

RPMTAG\_GROUP

  This tag specifies the administrative group to which this package belongs.

RPMTAG\_URL

  Generic package information URL.

RPMTAG\_OS

  This tag specifies the OS of the package. The OS value pointed to by this index record shall be "linux".

RPMTAG\_ARCH

  This tag specifies the architecture of the package. The architecture value pointed to by this index record is defined in architecture specific LSB specification.

RPMTAG\_SOURCERPM

  This tag specifies the name of the source RPM.

RPMTAG\_ARCHIVESIZE

  This tag specifies the uncompressed size of the Payload archive, including the cpio headers.

RPMTAG\_RPMVERSION

  This tag indicates the version of RPM tool used to build this package. The value is unused.

RPMTAG\_COOKIE

  This tag contains an opaque string whose contents are undefined.

RPMTAG\_DISTURL

  URL for package.

RPMTAG\_PAYLOADFORMAT

  This tag specifies the format of the Archive section. The format value pointed to by this index record shall be 'cpio'.

RPMTAG\_PAYLOADCOMPRESSOR

  This tag specifies the compression used on the Archive section. The compression value pointed to by this index record shall be 'gzip'.

RPMTAG\_PAYLOADFLAGS

  This tag indicates the compression level used for the Payload. This value shall always be '9'.

#### 25.2.4.2 Installation Information

The following tag values are used to provide information needed during the installation of the package.

**Table 25-9 Installation Tag Values**

| **Name** | **Tag Value** | **Type** | **Count** | **Status** |
| --- | --- | --- | --- | --- |
| RPMTAG\_PREIN | 1023 | STRING | 1 | Optional |
| RPMTAG\_POSTIN | 1024 | STRING | 1 | Optional |
| RPMTAG\_PREUN | 1025 | STRING | 1 | Optional |
| RPMTAG\_POSTUN | 1026 | STRING | 1 | Optional |
| RPMTAG\_PREINPROG | 1085 | STRING | 1 | Optional |
| RPMTAG\_POSTINPROG | 1086 | STRING | 1 | Optional |
| RPMTAG\_PREUNPROG | 1087 | STRING | 1 | Optional |
| RPMTAG\_POSTUNPROG | 1088 | STRING | 1 | Optional |

RPMTAG\_PREIN

  This tag specifies the preinstall scriptlet. If present, then RPMTAG\_PREINPROG shall also be present.

RPMTAG\_POSTIN

  This tag specifies the postinstall scriptlet. If present, then RPMTAG\_POSTINPROG shall also be present.

RPMTAG\_PREUN

  his tag specifies the preuninstall scriptlet. If present, then RPMTAG\_PREUNPROG shall also be present.

RPMTAG\_POSTUN

  This tag specified the postuninstall scriptlet. If present, then RPMTAG\_POSTUNPROG shall also be present.

RPMTAG\_PREINPROG

  This tag specifies the name of the intepreter to which the preinstall scriptlet will be passed. The intepreter pointed to by this index record shall be /bin/sh.

RPMTAG\_POSTINPROG

  This tag specifies the name of the intepreter to which the postinstall scriptlet will be passed. The intepreter pointed to by this index record shall be /bin/sh.

RPMTAG\_PREUNPROG

  This tag specifies the name of the intepreter to which the preuninstall scriptlet will be passed. The intepreter pointed to by this index record shall be /bin/sh.

RPMTAG\_POSTUNPROG

  This program specifies the name of the intepreter to which the postuninstall scriptlet will be passed. The intepreter pointed to by this index record shall be /bin/sh.

#### 25.2.4.3 File Information

The following tag values are used to provide information about the files in the payload. This information is provided in the header to allow more efficient access of the information.

**Table 25-10 File Info Tag Values**

| **Name** | **Tag Value** | **Type** | **Count** | **Status** |
| --- | --- | --- | --- | --- |
| RPMTAG\_OLDFILENAMES | 1027 | STRING\_ARRAY |  | Optional |
| RPMTAG\_FILESIZES | 1028 | INT32 |  | Required |
| RPMTAG\_FILEMODES | 1030 | INT16 |  | Required |
| RPMTAG\_FILERDEVS | 1033 | INT16 |  | Required |
| RPMTAG\_FILEMTIMES | 1034 | INT32 |  | Required |
| RPMTAG\_FILEMD5S | 1035 | STRING\_ARRAY |  | Required |
| RPMTAG\_FILELINKTOS | 1036 | STRING\_ARRAY |  | Required |
| RPMTAG\_FILEFLAGS | 1037 | INT32 |  | Required |
| RPMTAG\_FILEUSERNAME | 1039 | STRING\_ARRAY |  | Required |
| RPMTAG\_FILEGROUPNAME | 1040 | STRING\_ARRAY |  | Required |
| RPMTAG\_FILEDEVICES | 1095 | INT32 |  | Required |
| RPMTAG\_FILEINODES | 1096 | INT32 |  | Required |
| RPMTAG\_FILELANGS | 1097 | STRING\_ARRAY |  | Required |
| RPMTAG\_DIRINDEXES | 1116 | INT32 |  | Optional |
| RPMTAG\_BASENAMES | 1117 | STRING\_ARRAY |  | Optional |
| RPMTAG\_DIRNAMES | 1118 | STRING\_ARRAY |  | Optional |

RPMTAG\_OLDFILENAMES

  This tag specifies the filenames when not in a compressed format as determined by the absence of rpmlib(CompressedFileNames) in the RPMTAG\_REQUIRENAME index.

RPMTAG\_FILESIZES

  This tag specifies the size of each file in the archive.

RPMTAG\_FILEMODES

  This tag specifies the mode of each file in the archive.

RPMTAG\_FILERDEVS

  This tag specifies the device number from which the file was copied.

RPMTAG\_FILEMTIMES

  This tag specifies the modification time in seconds since the epoch of each file in the archive.

RPMTAG\_FILEMD5S

  This tag specifies the ASCII representation of the MD5 sum of the corresponding file contents. This value is empty if the corresponding archive entry is not a regular file.

RPMTAG\_FILELINKTOS

  The target for a symlink, otherwise NULL.

RPMTAG\_FILEFLAGS

  This tag specifies the bit(s) to classify and control how files are to be installed. See below.

RPMTAG\_FILEUSERNAME

  This tag specifies the owner of the corresponding file.

RPMTAG\_FILEGROUPNAME

  This tag specifies the group of the corresponding file.

RPMTAG\_FILEDEVICES

  This tag specifies the 16 bit device number from which the file was copied.

RPMTAG\_FILEINODES

  This tag specifies the inode value from the original file system on the the system on which it was built.

RPMTAG\_FILELANGS

  This tag specifies a per-file locale marker used to install only locale specific subsets of files when the package is installed.

RPMTAG\_DIRINDEXES

  This tag specifies the index into the array provided by the RPMTAG\_DIRNAMES Index which contains the directory name for the corresponding filename.

RPMTAG\_BASENAMES

  This tag specifies the base portion of the corresponding filename.

RPMTAG\_DIRNAMES

One of RPMTAG\_OLDFILENAMES or the tuple RPMTAG\_DIRINDEXES,RPMTAG\_BASENAMES,RPMTAG\_DIRNAMES shall be present, but not both.

##### *25.2.4.3.1 File Flags*

The RPMTAG\_FILEFLAGS tag value shall identify various characteristics of the file in the payload that it describes. It shall be an INT32 value consisting of either the value RPMFILE\_NONE (0) or the bitwise inclusive or of one or more of the following values:

**Table 25-11 File Flags**

| **Name** | **Value** |
| --- | --- |
| RPMFILE\_CONFIG | (1 << 0) |
| RPMFILE\_DOC | (1 << 1) |
| RPMFILE\_DONOTUSE | (1 << 2) |
| RPMFILE\_MISSINGOK | (1 << 3) |
| RPMFILE\_NOREPLACE | (1 << 4) |
| RPMFILE\_SPECFILE | (1 << 5) |
| RPMFILE\_GHOST | (1 << 6) |
| RPMFILE\_LICENSE | (1 << 7) |
| RPMFILE\_README | (1 << 8) |
| RPMFILE\_EXCLUDE | (1 << 9) |

These bits have the following meaning:

RPMFILE\_CONFIG

  The file is a configuration file, and an existing file should be saved during a package upgrade operation and not removed during a pakage removal operation.

RPMFILE\_DOC

  The file contains documentation.

RPMFILE\_DONOTUSE

  This value is reserved for future use; conforming packages may not use this flag.

RPMFILE\_MISSINGOK

  The file need not exist on the installed system.

RPMFILE\_NOREPLACE

  Similar to the RPMFILE\_CONFIG, this flag indicates that during an upgrade operation the original file on the system should not be altered.

RPMFILE\_SPECFILE

  The file is a package specification.

RPMFILE\_GHOST

  The file is not actually included in the payload, but should still be considered as a part of the package. For example, a log file generated by the application at run time.

RPMFILE\_LICENSE

  The file contains the license conditions.

RPMFILE\_README

  The file contains high level notes about the package.

RPMFILE\_EXCLUDE

  The corresponding file is not a part of the package, and should not be installed.

#### 25.2.4.4 Dependency Information

The following tag values are used to provide information about interdependencies between packages.

**Table 25-12 Package Dependency Tag Values**

| **Name** | **Tag Value** | **Type** | **Count** | **Status** |
| --- | --- | --- | --- | --- |
| RPMTAG\_PROVIDENAME | 1047 | STRING\_ARRAY | 1 | Required |
| RPMTAG\_REQUIREFLAGS | 1048 | INT32 |  | Required |
| RPMTAG\_REQUIRENAME | 1049 | STRING\_ARRAY |  | Required |
| RPMTAG\_REQUIREVERSION | 1050 | STRING\_ARRAY |  | Required |
| RPMTAG\_CONFLICTFLAGS | 1053 | INT32 |  | Optional |
| RPMTAG\_CONFLICTNAME | 1054 | STRING\_ARRAY |  | Optional |
| RPMTAG\_CONFLICTVERSION | 1055 | STRING\_ARRAY |  | Optional |
| RPMTAG\_OBSOLETENAME | 1090 | STRING\_ARRAY |  | Optional |
| RPMTAG\_PROVIDEFLAGS | 1112 | INT32 |  | Required |
| RPMTAG\_PROVIDEVERSION | 1113 | STRING\_ARRAY |  | Required |
| RPMTAG\_OBSOLETEFLAGS | 1114 | INT32 | 1 | Optional |
| RPMTAG\_OBSOLETEVERSION | 1115 | STRING\_ARRAY |  | Optional |

RPMTAG\_PROVIDENAME

  This tag indicates the name of the dependency provided by this package.

RPMTAG\_REQUIREFLAGS

  Bits(s) to specify the dependency range and context.

RPMTAG\_REQUIRENAME

  This tag indicates the dependencies for this package.

RPMTAG\_REQUIREVERSION

  This tag indicates the versions associated with the values found in the RPMTAG\_REQUIRENAME Index.

RPMTAG\_CONFLICTFLAGS

  Bits(s) to specify the conflict range and context.

RPMTAG\_CONFLICTNAME

  This tag indicates the conflicting dependencies for this package.

RPMTAG\_CONFLICTVERSION

  This tag indicates the versions associated with the values found in the RPMTAG\_CONFLICTNAME Index.

RPMTAG\_OBSOLETENAME

  This tag indicates the obsoleted dependencies for this package.

RPMTAG\_PROVIDEFLAGS

  Bits(s) to specify the conflict range and context.

RPMTAG\_PROVIDEVERSION

  This tag indicates the versions associated with the values found in the RPMTAG\_PROVIDENAME Index.

RPMTAG\_OBSOLETEFLAGS

  Bits(s) to specify the conflict range and context.

RPMTAG\_OBSOLETEVERSION

  This tag indicates the versions associated with the values found in the RPMTAG\_OBSOLETENAME Index.

##### *25.2.4.4.1 Package Dependency Values*

The package dependencies are stored in the RPMTAG\_REQUIRENAME and RPMTAG\_REQUIREVERSION index records. The following values may be used.

**Table 25-13 Index Type values**

| **Name** | **Version** | **Meaning** | **Status** |
| --- | --- | --- | --- |
| rpmlib(VersionedDependencies) | 3.0.3-1 | Indicates that the package contains RPMTAG\_PROVIDENAME, RPMTAG\_OBSOLETENAME or RPMTAG\_PREREQ records that have a version associated with them. | Optional |
| rpmlib(PayloadFilesHavePrefix) | 4.0-1 | Indicates the filenames in the Archive have had "." prepended to them. | Optional |
| rpmlib(CompressedFileNames) | 3.0.4-1 | Indicates that the filenames in the Payload are represented in the RPMTAG\_DIRINDEXES, RPMTAG\_DIRNAME and RPMTAG\_BASENAMES indexes. | Optional |
| **/bin/sh** |  | Interpreter usually required for installation scripts. | Optional |

Additional dependencies are specified in the Package Dependencies section of this specification, and in the relevant architecture specific part of the LSB Core Specification.

##### *25.2.4.4.2 Package Dependencies Attributes*

The package dependency attributes are stored in the RPMTAG\_REQUIREFLAGS, RPMTAG\_PROVIDEFLAGS and RPMTAG\_OBSOLETEFLAGS index records. The following values may be used.

**Table 25-14 Package Dependency Attributes**

| **Name** | **Value** | **Meaning** |
| --- | --- | --- |
| RPMSENSE\_LESS | 0x02 |  |
| RPMSENSE\_GREATER | 0x04 |  |
| RPMSENSE\_EQUAL | 0x08 |  |
| RPMSENSE\_PREREQ | 0x40 |  |
| RPMSENSE\_INTERP | 0x100 |  |
| RPMSENSE\_SCRIPT\_PRE | 0x200 |  |
| RPMSENSE\_SCRIPT\_POST | 0x400 |  |
| RPMSENSE\_SCRIPT\_PREUN | 0x800 |  |
| RPMSENSE\_SCRIPT\_POSTUN | 0x1000 |  |
| RPMSENSE\_RPMLIB | 0x1000000 |  |

#### 25.2.4.5 Other Information

The following tag values are also found in the Header section.

**Table 25-15 Other Tag Values**

| **Name** | **Tag Value** | **Type** | **Count** | **Status** |
| --- | --- | --- | --- | --- |
| RPMTAG\_BUILDTIME | 1006 | INT32 | 1 | Informational |
| RPMTAG\_BUILDHOST | 1007 | STRING | 1 | Informational |
| RPMTAG\_FILEVERIFYFLAGS | 1045 | INT32 |  | Optional |
| RPMTAG\_CHANGELOGTIME | 1080 | INT32 |  | Optional |
| RPMTAG\_CHANGELOGNAME | 1081 | STRING\_ARRAY |  | Optional |
| RPMTAG\_CHANGELOGTEXT | 1082 | STRING\_ARRAY |  | Optional |
| RPMTAG\_OPTFLAGS | 1122 | STRING | 1 | Informational |
| RPMTAG\_RHNPLATFORM | 1131 | STRING | 1 | Deprecated |
| RPMTAG\_PLATFORM | 1132 | STRING | 1 | Informational |

RPMTAG\_BUILDTIME

  This tag specifies the time as seconds since the epoch at which the package was built.

RPMTAG\_BUILDHOST

  This tag specifies the hostname of the system on which which the package was built.

RPMTAG\_FILEVERIFYFLAGS

  This tag specifies the bit(s) to control how files are to be verified after install, specifying which checks should be performed.

RPMTAG\_CHANGELOGTIME

  This tag specifies the Unix time in seconds since the epoch associated with each entry in the Changelog file.

RPMTAG\_CHANGELOGNAME

  This tag specifies the name of who made a change to this package.

RPMTAG\_CHANGELOGTEXT

  This tag specifies the changes asssociated with a changelog entry.

RPMTAG\_OPTFLAGS

  This tag indicates additional flags which may have been passed to the compiler when building this package.

RPMTAG\_RHNPLATFORM

  This tag contains an opaque string whose contents are undefined.

RPMTAG\_PLATFORM

  This tag contains an opaque string whose contents are undefined.

### **25.2.5 Payload Section**

The Payload section contains a compressed cpio archive. The format of this section is defined by [RFC 1952: GZIP File Format Specification](#ID_STD_46_RFC1952).

When uncompressed, the cpio archive contains a sequence of records for each file. Each record contains a CPIO Header, Filename, Padding, and File Data.

**Table 25-16 CPIO File Format**

|  |  |
| --- | --- |
| CPIO Header | Header structure as defined below. |
| Filename | NUL terminated ASCII string containing the name of the file. |
| Padding | 0-3 bytes as needed to align the file stream to a 4 byte boundary. |
| File data | The contents of the file. |
| Padding | 0-3 bytes as needed to align the file stream to a 4 byte boundary. |

The CPIO Header uses the following header structure (sometimes referred to as "new ASCII" or "SVR4 cpio"). All numbers are stored as ASCII representations of their hexadecimal value with leading zeros as needed to fill the field. With the exception of *c\_namesize* and the corresponding name string, and *c\_checksum*, all information contained in the CPIO Header is also represented in the Header Section. The values in the CPIO Header shall match the values contained in the Header Section.

struct {

char c\_magic[6];

char c\_ino[8];

char c\_mode[8];

char c\_uid[8];

char c\_gid[8];

char c\_nlink[8];

char c\_mtime[8];

char c\_filesize[8];

char c\_devmajor[8];

char c\_devminor[8];

char c\_rdevmajor[8];

char c\_rdevminor[8];

char c\_namesize[8];

char c\_checksum[8];

};

*c\_magic*

  Value identifying this cpio format. This value shall be "070701".

*c\_ino*

  This field contains the inode number from the filesystem from which the file was read. This field is ignored when installing a package. This field shall match the corresponding value in the RPMTAG\_FILEINODES index in the Header section.

*c\_mode*

  Permission bits of the file. This is an ascii representation of the hexadecimal number representing the bit as defined for the *st\_mode* field of the stat structure defined for the stat function. This field shall match the corresponding value in the RPMTAG\_FILEMODES index in the Header section.

*c\_uid*

  Value identifying this owner of this file. This value matches the uid value of the corresponding user in the RPMTAG\_FILEUSERNAME as found on the system where this package was built. The username specified in RPMTAG\_FILEUSERNAME should take precedence when installing the package.

*c\_gid*

  Value identifying this group of this file. This value matches the gid value of the corresponding user in the RPMTAG\_FILEGROUPNAME as found on the system where this package was built. The groupname specified in RPMTAG\_FILEGROUPNAME should take precedence when installing the package.

*c\_nlink*

  Value identifying the number of links associated with this file. If the value is greater than 1, then this filename will be linked to 1 or more files in this archive that has a matching value for the c\_ino, c\_devmajor and c\_devminor fields.

*c\_mtime*

  Value identifying the modification time of the file when it was read. This field shall match the corresponding value in the RPMTAG\_FILEMTIMES index in the Header section.

*c\_filesize*

  Value identifying the size of the file. This field shall match the corresponding value in the RPMTAG\_FILESIZES index in the Header section.

*c\_devmajor*

  The major number of the device containing the file system from which the file was read. With the exception of processing files with c\_nlink >1, this field is ignored when installing a package. This field shall match the corresponding value in the RPMTAG\_FILEDEVICES index in the Header section.

*c\_devminor*

  The minor number of the device containing the file system from which the file was read. With the exception of processing files with c\_nlink >1, this field is ignored when installing a package. This field shall match the corresponding value in the RPMTAG\_FILEDEVICES index in the Header section.

*c\_rdevmajor*

  The major number of the raw device containing the file system from which the file was read. This field is ignored when installing a package. This field shall match the corresponding value in the RPMTAG\_RDEVS index in the Header section.

*c\_rdevminor*

  The minor number of the raw device containing the file system from which the file was read. This field is ignored when installing a package. This field shall match the corresponding value in the RPMTAG\_RDEVS index in the Header section.

*c\_namesize*

  Value identifying the length of the filename, which is located immediately following the CPIO Header structure.

*c\_checksum*

  Value containing the CRC checksum of the file data. This field is not used, and shall contain the value "00000000". This field is ignored when installing a package.

A record with the filename "TRAILER!!!" indicates the last record in the archive.

## **25.3 Package Script Restrictions**

Scripts used as part of the package install and uninstall shall only use commands and interfaces that are specified by the LSB. All other commands are not guaranteed to be present, or to behave in expected ways.

Packages shall not use RPM triggers.

Packages shall not depend on the order in which scripts are executed (pre-install, pre-uninstall, etc), when doing an upgrade.

## **25.4 Package Tools**

The LSB does not specify the interface to the tools used to manipulate LSB-conformant packages. Each conforming implementation shall provide documentation for installing LSB packages.

## **25.5 Package Naming Conventions**

Packages supplied by distributions and applications should adhere to the following conventions for the name field within the package. The rules are optional for the filename of the package file itself.

**Note:** There are discrepancies among implementations concerning whether the name might be frobnicator-1.7-21-ppc32.rpm or frobnicator-1.7-21-powerpc32.rpm. The architecture aside, recommended practice is for the filename of the package file to match the name within the package.

The following conventions apply to the name portion of the field alone, not including any release or version portion.

**Note:** If the package name with the release and version is frobnicator-1.7-21, the name part is frobnicator and falls under the conventions for a name with no hyphens.

• If the name begins with lsb- and contains no other hyphens, the name should be a package name registered with the Linux Assigned Names and Numbers Authority (http://www.lanana.org) (LANANA), which shall maintain a registry of LSB names. The name may be registered by either an implementation or an application.

• If the name begins with lsb- and contains more than one hyphen the portion of the name between the first and second hyphens should be either an LSB provider name registered with the LANANA (for example lsb-gnome-gnumeric if gnome is registered), or a domain name registered to the provider in the DNS system. (for example lsb-distro.example.com-database). The LSB provider name registered with the LANANA shall only consist of the ASCII characters [a-z0-9]. The domain name, in accordance with DNS rules, shall be lower case only. The provider name or domain name may be either that of a distribution or an application.

• Package names containing no hyphens are reserved for use by distributions. Applications shall not use such names.

• Package names which do not start with lsb- and which contain a hyphen are open to both distributions and applications. Distributions may name packages in any part of this namespace. They are encouraged to use names from one of the other namespaces available to them, but this is not mandatory due to the large amount of current practice to the contrary.

**Note:** Widespread existing practice includes such names as ssh-common, ssh-client, kernel-pcmcia, and the like. Possible alternative names include sshcommon, foolinux-ssh-common (where foolinux is registered to the distribution), or lsb-foolinux-ssh-common.

Applications may name their packages this way, but only if the portion of the name before the first hyphen is a provider name or registered domain name as described above.

**Note:** If an application vendor has domain name such as visicalc.example.com and has registered visicalc as a provider name, they could name packages either in the visicalc-base style or the visicalc.example.com-charting style.

Package names in this namespace are available to both the distribution and an application. Distributions and applications need to consider the potential for conflicts when deciding to use these names rather than the alternatives (such as names starting with lsb-).

## **25.6 Package Dependencies**

Packages shall have a dependency that indicates which LSB modules are required. LSB module descriptions are dash seperated tuples containing the name 'lsb', the module name, and the architecture name. The following dependencies may be used.

lsb-core

  This dependency is used to indicate that the application is dependent on features contained in the LSB Core specification.

lsb-core-*arch*

  This dependency is used to indicate that the application is dependent on features contained in the LSB Core specification and that the package contains architecture specific features. This architecture specific dependency is described in the relevant architecture specific part of the LSB Core specification.

lsb-core-noarch

  This dependency is used to indicate that the application is dependent on features contained in the LSB Core specification and that the package does not contain any architecture specific files.

These dependencies shall have a version of 5.0.

Packages shall not depend on other system-provided dependencies. They shall not depend on non-system-provided dependencies unless the package provider also makes available the LSB conforming packages needed to satisfy such dependencies.

Other modules in the LSB may supplement this list. The architecture specific dependencies are described in the relevant architecture specific LSB.

## **25.7 Package Architecture Considerations**

Packages which do not contain any architecture specific files should specify an architecture of noarch. An LSB runtime environment shall accept values noarch, or the value specified in the relevant architecture specific part of the LSB Core Specification.

Additional specifications or restrictions may be found in the architecture specific LSB specification.

# **Annex A Alphabetical Listing of Interfaces by Library**

## **A.1 libc**

The behavior of the interfaces in this library is specified by the following Standards.

|  |
| --- |
| [Large File Support](#ID_STD_46_LFS) [LFS] |
| [This Specification](#ID_STD_46_LSB) [LSB] |
| [RFC 5531/4506 RPC & XDR](#ID_STD_46_RPC_46_XDR) [RPC + XDR] |
| [SUSv2](#ID_STD_46_SUSV2) [SUSv2] |
| [POSIX 1003.1-2001 (ISO/IEC 9945-2003)](#ID_STD_46_SUSV3) [SUSv3] |
| [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) [SUSv4] |
| [SVID Issue 4](#ID_STD_46_SVID_46_4) [SVID.4] |

**Table A-1 libc Function Interfaces**

|  |  |  |
| --- | --- | --- |
| \_Exit[[SUSv4]](#ID_STD_46_SUSV4) | getdelim[[SUSv4]](#ID_STD_46_SUSV4) | scandir[[SUSv4]](#ID_STD_46_SUSV4) |
| \_IO\_feof[[LSB]](#ID_STD_46_LSB) | getdomainname[[LSB]](#ID_STD_46_LSB) | scandir64[[LSB]](#ID_STD_46_LSB) |
| \_IO\_getc[[LSB]](#ID_STD_46_LSB) | getdtablesize[[LSB]](#ID_STD_46_LSB) | scanf[[LSB]](#ID_STD_46_LSB) |
| \_IO\_putc[[LSB]](#ID_STD_46_LSB) | getegid[[SUSv4]](#ID_STD_46_SUSV4) | sched\_get\_priority\_max[[SUSv4]](#ID_STD_46_SUSV4) |
| \_IO\_puts[[LSB]](#ID_STD_46_LSB) | getenv[[SUSv4]](#ID_STD_46_SUSV4) | sched\_get\_priority\_min[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_assert\_fail[[LSB]](#ID_STD_46_LSB) | geteuid[[SUSv4]](#ID_STD_46_SUSV4) | sched\_getaffinity(GLIBC\_2.3.4)[[LSB]](#ID_STD_46_LSB) |
| \_\_chk\_fail(GLIBC\_2.3.4)[[LSB]](#ID_STD_46_LSB) | getgid[[SUSv4]](#ID_STD_46_SUSV4) | sched\_getparam[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_confstr\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | getgrent[[SUSv4]](#ID_STD_46_SUSV4) | sched\_getscheduler[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_ctype\_b\_loc(GLIBC\_2.3)[[LSB]](#ID_STD_46_LSB) | getgrent\_r[[LSB]](#ID_STD_46_LSB) | sched\_rr\_get\_interval[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_ctype\_get\_mb\_cur\_max[[LSB]](#ID_STD_46_LSB) | getgrgid[[SUSv4]](#ID_STD_46_SUSV4) | sched\_setaffinity(GLIBC\_2.3.4)[[LSB]](#ID_STD_46_LSB) |
| \_\_ctype\_tolower\_loc(GLIBC\_2.3)[[LSB]](#ID_STD_46_LSB) | getgrgid\_r[[SUSv4]](#ID_STD_46_SUSV4) | sched\_setparam[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_ctype\_toupper\_loc(GLIBC\_2.3)[[LSB]](#ID_STD_46_LSB) | getgrnam[[SUSv4]](#ID_STD_46_SUSV4) | sched\_setscheduler[[LSB]](#ID_STD_46_LSB) |
| \_\_cxa\_atexit[[LSB]](#ID_STD_46_LSB) | getgrnam\_r[[SUSv4]](#ID_STD_46_SUSV4) | sched\_yield[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_cxa\_finalize[[LSB]](#ID_STD_46_LSB) | getgrouplist[[LSB]](#ID_STD_46_LSB) | seed48[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_errno\_location[[LSB]](#ID_STD_46_LSB) | getgroups[[SUSv4]](#ID_STD_46_SUSV4) | seed48\_r[[LSB]](#ID_STD_46_LSB) |
| \_\_fgets\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | gethostbyaddr[[SUSv3]](#ID_STD_46_SUSV3) | seekdir[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_fgets\_unlocked\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | gethostbyaddr\_r[[LSB]](#ID_STD_46_LSB) | select[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_fgetws\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | gethostbyname[[SUSv3]](#ID_STD_46_SUSV3) | semctl[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_fgetws\_unlocked\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | gethostbyname2[[LSB]](#ID_STD_46_LSB) | semget[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_fpending[[LSB]](#ID_STD_46_LSB) | gethostbyname2\_r[[LSB]](#ID_STD_46_LSB) | semop[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_fprintf\_chk[[LSB]](#ID_STD_46_LSB) | gethostbyname\_r[[LSB]](#ID_STD_46_LSB) | send[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_fwprintf\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | gethostid[[SUSv4]](#ID_STD_46_SUSV4) | sendfile[[LSB]](#ID_STD_46_LSB) |
| \_\_fxstat[[LSB]](#ID_STD_46_LSB) | gethostname[[SUSv4]](#ID_STD_46_SUSV4) | sendfile64(GLIBC\_2.3)[[LSB]](#ID_STD_46_LSB) |
| \_\_fxstat64[[LSB]](#ID_STD_46_LSB) | getifaddrs(GLIBC\_2.3)[[LSB]](#ID_STD_46_LSB) | sendmsg[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_fxstatat(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | getitimer[[SUSv4]](#ID_STD_46_SUSV4) | sendto[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_fxstatat64(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | getline[[SUSv4]](#ID_STD_46_SUSV4) | setbuf[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_getcwd\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | getloadavg[[LSB]](#ID_STD_46_LSB) | setbuffer[[LSB]](#ID_STD_46_LSB) |
| \_\_getgroups\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | getlogin[[SUSv4]](#ID_STD_46_SUSV4) | setcontext[[SUSv3]](#ID_STD_46_SUSV3) |
| \_\_gethostname\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | getlogin\_r[[SUSv4]](#ID_STD_46_SUSV4) | setegid[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_getlogin\_r\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | getnameinfo[[SUSv4]](#ID_STD_46_SUSV4) | setenv[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_getpagesize[[LSB]](#ID_STD_46_LSB) | getopt[[LSB]](#ID_STD_46_LSB) | seteuid[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_getpgid[[LSB]](#ID_STD_46_LSB) | getopt\_long[[LSB]](#ID_STD_46_LSB) | setgid[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_h\_errno\_location[[LSB]](#ID_STD_46_LSB) | getopt\_long\_only[[LSB]](#ID_STD_46_LSB) | setgrent[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_isinf[[LSB]](#ID_STD_46_LSB) | getpagesize[[LSB]](#ID_STD_46_LSB) | setgroups[[LSB]](#ID_STD_46_LSB) |
| \_\_isinff[[LSB]](#ID_STD_46_LSB) | getpeername[[SUSv4]](#ID_STD_46_SUSV4) | sethostname[[LSB]](#ID_STD_46_LSB) |
| \_\_isinfl[[LSB]](#ID_STD_46_LSB) | getpgid[[SUSv4]](#ID_STD_46_SUSV4) | setitimer[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_isnan[[LSB]](#ID_STD_46_LSB) | getpgrp[[SUSv4]](#ID_STD_46_SUSV4) | setlocale[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_isnanf[[LSB]](#ID_STD_46_LSB) | getpid[[SUSv4]](#ID_STD_46_SUSV4) | setlogmask[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_isnanl[[LSB]](#ID_STD_46_LSB) | getppid[[SUSv4]](#ID_STD_46_SUSV4) | setpgid[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_libc\_current\_sigrtmax[[LSB]](#ID_STD_46_LSB) | getpriority[[SUSv4]](#ID_STD_46_SUSV4) | setpgrp[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_libc\_current\_sigrtmin[[LSB]](#ID_STD_46_LSB) | getprotobyname[[SUSv4]](#ID_STD_46_SUSV4) | setpriority[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_libc\_start\_main[[LSB]](#ID_STD_46_LSB) | getprotobyname\_r[[LSB]](#ID_STD_46_LSB) | setprotoent[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_lxstat[[LSB]](#ID_STD_46_LSB) | getprotobynumber[[SUSv4]](#ID_STD_46_SUSV4) | setpwent[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_lxstat64[[LSB]](#ID_STD_46_LSB) | getprotobynumber\_r[[LSB]](#ID_STD_46_LSB) | setregid[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_mbsnrtowcs\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | getprotoent[[SUSv4]](#ID_STD_46_SUSV4) | setreuid[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_mbsrtowcs\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | getprotoent\_r[[LSB]](#ID_STD_46_LSB) | setrlimit[[LSB]](#ID_STD_46_LSB) |
| \_\_mbstowcs\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | getpwent[[SUSv4]](#ID_STD_46_SUSV4) | setrlimit64[[LFS]](#ID_STD_46_LFS) |
| \_\_memcpy\_chk(GLIBC\_2.3.4)[[LSB]](#ID_STD_46_LSB) | getpwent\_r[[LSB]](#ID_STD_46_LSB) | setservent[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_memmove\_chk(GLIBC\_2.3.4)[[LSB]](#ID_STD_46_LSB) | getpwnam[[SUSv4]](#ID_STD_46_SUSV4) | setsid[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_mempcpy[[LSB]](#ID_STD_46_LSB) | getpwnam\_r[[SUSv4]](#ID_STD_46_SUSV4) | setsockopt[[LSB]](#ID_STD_46_LSB) |
| \_\_mempcpy\_chk(GLIBC\_2.3.4)[[LSB]](#ID_STD_46_LSB) | getpwuid[[SUSv4]](#ID_STD_46_SUSV4) | setstate[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_memset\_chk(GLIBC\_2.3.4)[[LSB]](#ID_STD_46_LSB) | getpwuid\_r[[SUSv4]](#ID_STD_46_SUSV4) | setstate\_r[[LSB]](#ID_STD_46_LSB) |
| \_\_pread64\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | getrlimit[[LSB]](#ID_STD_46_LSB) | setuid[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_pread\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | getrlimit64[[LFS]](#ID_STD_46_LFS) | setutent[[LSB]](#ID_STD_46_LSB) |
| \_\_printf\_chk[[LSB]](#ID_STD_46_LSB) | getrusage[[SUSv4]](#ID_STD_46_SUSV4) | setutxent[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_rawmemchr[[LSB]](#ID_STD_46_LSB) | getservbyname[[SUSv4]](#ID_STD_46_SUSV4) | setvbuf[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_read\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | getservbyname\_r[[LSB]](#ID_STD_46_LSB) | shmat[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_readlink\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | getservbyport[[SUSv4]](#ID_STD_46_SUSV4) | shmctl[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_realpath\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | getservbyport\_r[[LSB]](#ID_STD_46_LSB) | shmdt[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_recv\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | getservent[[SUSv4]](#ID_STD_46_SUSV4) | shmget[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_recvfrom\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | getservent\_r[[LSB]](#ID_STD_46_LSB) | shutdown[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_register\_atfork(GLIBC\_2.3.2)[[LSB]](#ID_STD_46_LSB) | getsid[[SUSv4]](#ID_STD_46_SUSV4) | sigaction[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_sigsetjmp[[LSB]](#ID_STD_46_LSB) | getsockname[[SUSv4]](#ID_STD_46_SUSV4) | sigaddset[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_snprintf\_chk[[LSB]](#ID_STD_46_LSB) | getsockopt[[LSB]](#ID_STD_46_LSB) | sigaltstack[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_sprintf\_chk[[LSB]](#ID_STD_46_LSB) | getsubopt[[SUSv4]](#ID_STD_46_SUSV4) | sigandset[[LSB]](#ID_STD_46_LSB) |
| \_\_stack\_chk\_fail(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | gettext[[LSB]](#ID_STD_46_LSB) | sigdelset[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_stpcpy[[LSB]](#ID_STD_46_LSB) | gettimeofday[[SUSv4]](#ID_STD_46_SUSV4) | sigemptyset[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_stpcpy\_chk(GLIBC\_2.3.4)[[LSB]](#ID_STD_46_LSB) | getuid[[SUSv4]](#ID_STD_46_SUSV4) | sigfillset[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_stpncpy\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | getutent[[LSB]](#ID_STD_46_LSB) | sighold[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_strcat\_chk(GLIBC\_2.3.4)[[LSB]](#ID_STD_46_LSB) | getutent\_r[[LSB]](#ID_STD_46_LSB) | sigignore[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_strcpy\_chk(GLIBC\_2.3.4)[[LSB]](#ID_STD_46_LSB) | getutxent[[SUSv4]](#ID_STD_46_SUSV4) | siginterrupt[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_strdup[[LSB]](#ID_STD_46_LSB) | getutxid[[SUSv4]](#ID_STD_46_SUSV4) | sigisemptyset[[LSB]](#ID_STD_46_LSB) |
| \_\_strncat\_chk(GLIBC\_2.3.4)[[LSB]](#ID_STD_46_LSB) | getutxline[[SUSv4]](#ID_STD_46_SUSV4) | sigismember[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_strncpy\_chk(GLIBC\_2.3.4)[[LSB]](#ID_STD_46_LSB) | getw[[SUSv2]](#ID_STD_46_SUSV2) | siglongjmp[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_strtod\_internal[[LSB]](#ID_STD_46_LSB) | getwc[[SUSv4]](#ID_STD_46_SUSV4) | signal[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_strtof\_internal[[LSB]](#ID_STD_46_LSB) | getwc\_unlocked[[LSB]](#ID_STD_46_LSB) | sigorset[[LSB]](#ID_STD_46_LSB) |
| \_\_strtok\_r[[LSB]](#ID_STD_46_LSB) | getwchar[[SUSv4]](#ID_STD_46_SUSV4) | sigpause[[LSB]](#ID_STD_46_LSB) |
| \_\_strtol\_internal[[LSB]](#ID_STD_46_LSB) | getwchar\_unlocked[[LSB]](#ID_STD_46_LSB) | sigpending[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_strtold\_internal[[LSB]](#ID_STD_46_LSB) | getwd[[SUSv3]](#ID_STD_46_SUSV3) | sigprocmask[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_strtoll\_internal[[LSB]](#ID_STD_46_LSB) | glob[[SUSv4]](#ID_STD_46_SUSV4) | sigqueue[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_strtoul\_internal[[LSB]](#ID_STD_46_LSB) | glob64[[LSB]](#ID_STD_46_LSB) | sigrelse[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_strtoull\_internal[[LSB]](#ID_STD_46_LSB) | globfree[[SUSv4]](#ID_STD_46_SUSV4) | sigreturn[[LSB]](#ID_STD_46_LSB) |
| \_\_swprintf\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | globfree64[[LSB]](#ID_STD_46_LSB) | sigset[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_sysconf[[LSB]](#ID_STD_46_LSB) | gmtime[[SUSv4]](#ID_STD_46_SUSV4) | sigsuspend[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_syslog\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | gmtime\_r[[SUSv4]](#ID_STD_46_SUSV4) | sigtimedwait[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_sysv\_signal[[LSB]](#ID_STD_46_LSB) | gnu\_get\_libc\_release[[LSB]](#ID_STD_46_LSB) | sigwait[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_ttyname\_r\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | gnu\_get\_libc\_version[[LSB]](#ID_STD_46_LSB) | sigwaitinfo[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_vfprintf\_chk[[LSB]](#ID_STD_46_LSB) | grantpt[[SUSv4]](#ID_STD_46_SUSV4) | sleep[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_vfwprintf\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | hcreate[[SUSv4]](#ID_STD_46_SUSV4) | snprintf[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_vprintf\_chk[[LSB]](#ID_STD_46_LSB) | hcreate\_r[[LSB]](#ID_STD_46_LSB) | sockatmark[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_vsnprintf\_chk[[LSB]](#ID_STD_46_LSB) | hdestroy[[SUSv4]](#ID_STD_46_SUSV4) | socket[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_vsprintf\_chk[[LSB]](#ID_STD_46_LSB) | hdestroy\_r[[LSB]](#ID_STD_46_LSB) | socketpair[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_vswprintf\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | hsearch[[SUSv4]](#ID_STD_46_SUSV4) | sprintf[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_vsyslog\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | hsearch\_r[[LSB]](#ID_STD_46_LSB) | srand[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_vwprintf\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | htonl[[SUSv4]](#ID_STD_46_SUSV4) | srand48[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_wcpcpy\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | htons[[SUSv4]](#ID_STD_46_SUSV4) | srand48\_r[[LSB]](#ID_STD_46_LSB) |
| \_\_wcpncpy\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | iconv[[SUSv4]](#ID_STD_46_SUSV4) | srandom[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_wcrtomb\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | iconv\_close[[SUSv4]](#ID_STD_46_SUSV4) | srandom\_r[[LSB]](#ID_STD_46_LSB) |
| \_\_wcscat\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | iconv\_open[[SUSv4]](#ID_STD_46_SUSV4) | sscanf[[LSB]](#ID_STD_46_LSB) |
| \_\_wcscpy\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | if\_freenameindex[[SUSv4]](#ID_STD_46_SUSV4) | statfs[[LSB]](#ID_STD_46_LSB) |
| \_\_wcsncat\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | if\_indextoname[[SUSv4]](#ID_STD_46_SUSV4) | statfs64[[LSB]](#ID_STD_46_LSB) |
| \_\_wcsncpy\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | if\_nameindex[[SUSv4]](#ID_STD_46_SUSV4) | statvfs[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_wcsnrtombs\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | if\_nametoindex[[SUSv4]](#ID_STD_46_SUSV4) | statvfs64[[LFS]](#ID_STD_46_LFS) |
| \_\_wcsrtombs\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | imaxabs[[SUSv4]](#ID_STD_46_SUSV4) | stime[[LSB]](#ID_STD_46_LSB) |
| \_\_wcstod\_internal[[LSB]](#ID_STD_46_LSB) | imaxdiv[[SUSv4]](#ID_STD_46_SUSV4) | stpcpy[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_wcstof\_internal[[LSB]](#ID_STD_46_LSB) | index[[SUSv3]](#ID_STD_46_SUSV3) | stpncpy[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_wcstol\_internal[[LSB]](#ID_STD_46_LSB) | inet\_addr[[SUSv4]](#ID_STD_46_SUSV4) | strcasecmp[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_wcstold\_internal[[LSB]](#ID_STD_46_LSB) | inet\_aton[[LSB]](#ID_STD_46_LSB) | strcasecmp\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_wcstombs\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | inet\_ntoa[[SUSv4]](#ID_STD_46_SUSV4) | strcasestr[[LSB]](#ID_STD_46_LSB) |
| \_\_wcstoul\_internal[[LSB]](#ID_STD_46_LSB) | inet\_ntop[[SUSv4]](#ID_STD_46_SUSV4) | strcat[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_wctomb\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | inet\_pton[[SUSv4]](#ID_STD_46_SUSV4) | strchr[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_wmemcpy\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | initgroups[[LSB]](#ID_STD_46_LSB) | strcmp[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_wmemmove\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | initstate[[SUSv4]](#ID_STD_46_SUSV4) | strcoll[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_wmempcpy\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | initstate\_r[[LSB]](#ID_STD_46_LSB) | strcoll\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_wmemset\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | inotify\_add\_watch(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | strcpy[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_wprintf\_chk(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | inotify\_init(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | strcspn[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_xmknod[[LSB]](#ID_STD_46_LSB) | inotify\_rm\_watch(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | strdup[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_xmknodat(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | insque[[SUSv4]](#ID_STD_46_SUSV4) | strerror[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_xpg\_basename[[LSB]](#ID_STD_46_LSB) | ioctl[[LSB]](#ID_STD_46_LSB) | strerror\_l(GLIBC\_2.6)[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_xpg\_sigpause[[LSB]](#ID_STD_46_LSB) | isalnum[[SUSv4]](#ID_STD_46_SUSV4) | strerror\_r[[LSB]](#ID_STD_46_LSB) |
| \_\_xpg\_strerror\_r(GLIBC\_2.3.4)[[LSB]](#ID_STD_46_LSB) | isalnum\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) | strfmon[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_xstat[[LSB]](#ID_STD_46_LSB) | isalpha[[SUSv4]](#ID_STD_46_SUSV4) | strfmon\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_xstat64[[LSB]](#ID_STD_46_LSB) | isalpha\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) | strftime[[SUSv4]](#ID_STD_46_SUSV4) |
| \_exit[[SUSv4]](#ID_STD_46_SUSV4) | isascii[[SUSv4]](#ID_STD_46_SUSV4) | strftime\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) |
| \_longjmp[[SUSv4]](#ID_STD_46_SUSV4) | isatty[[SUSv4]](#ID_STD_46_SUSV4) | strlen[[SUSv4]](#ID_STD_46_SUSV4) |
| \_setjmp[[SUSv4]](#ID_STD_46_SUSV4) | isblank[[SUSv4]](#ID_STD_46_SUSV4) | strncasecmp[[SUSv4]](#ID_STD_46_SUSV4) |
| \_tolower[[SUSv4]](#ID_STD_46_SUSV4) | isblank\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) | strncasecmp\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) |
| \_toupper[[SUSv4]](#ID_STD_46_SUSV4) | iscntrl[[SUSv4]](#ID_STD_46_SUSV4) | strncat[[SUSv4]](#ID_STD_46_SUSV4) |
| a64l[[SUSv4]](#ID_STD_46_SUSV4) | iscntrl\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) | strncmp[[SUSv4]](#ID_STD_46_SUSV4) |
| abort[[SUSv4]](#ID_STD_46_SUSV4) | isdigit[[SUSv4]](#ID_STD_46_SUSV4) | strncpy[[SUSv4]](#ID_STD_46_SUSV4) |
| abs[[SUSv4]](#ID_STD_46_SUSV4) | isdigit\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) | strndup[[SUSv4]](#ID_STD_46_SUSV4) |
| accept[[SUSv4]](#ID_STD_46_SUSV4) | isgraph[[SUSv4]](#ID_STD_46_SUSV4) | strnlen[[SUSv4]](#ID_STD_46_SUSV4) |
| access[[SUSv4]](#ID_STD_46_SUSV4) | isgraph\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) | strpbrk[[SUSv4]](#ID_STD_46_SUSV4) |
| acct[[LSB]](#ID_STD_46_LSB) | islower[[SUSv4]](#ID_STD_46_SUSV4) | strptime[[LSB]](#ID_STD_46_LSB) |
| adjtime[[LSB]](#ID_STD_46_LSB) | islower\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) | strrchr[[SUSv4]](#ID_STD_46_SUSV4) |
| alarm[[SUSv4]](#ID_STD_46_SUSV4) | isprint[[SUSv4]](#ID_STD_46_SUSV4) | strsep[[LSB]](#ID_STD_46_LSB) |
| alphasort[[SUSv4]](#ID_STD_46_SUSV4) | isprint\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) | strsignal[[SUSv4]](#ID_STD_46_SUSV4) |
| alphasort64[[LSB]](#ID_STD_46_LSB) | ispunct[[SUSv4]](#ID_STD_46_SUSV4) | strspn[[SUSv4]](#ID_STD_46_SUSV4) |
| argz\_add[[LSB]](#ID_STD_46_LSB) | ispunct\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) | strstr[[SUSv4]](#ID_STD_46_SUSV4) |
| argz\_add\_sep[[LSB]](#ID_STD_46_LSB) | isspace[[SUSv4]](#ID_STD_46_SUSV4) | strtod[[SUSv4]](#ID_STD_46_SUSV4) |
| argz\_append[[LSB]](#ID_STD_46_LSB) | isspace\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) | strtof[[SUSv4]](#ID_STD_46_SUSV4) |
| argz\_count[[LSB]](#ID_STD_46_LSB) | isupper[[SUSv4]](#ID_STD_46_SUSV4) | strtoimax[[SUSv4]](#ID_STD_46_SUSV4) |
| argz\_create[[LSB]](#ID_STD_46_LSB) | isupper\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) | strtok[[SUSv4]](#ID_STD_46_SUSV4) |
| argz\_create\_sep[[LSB]](#ID_STD_46_LSB) | iswalnum[[SUSv4]](#ID_STD_46_SUSV4) | strtok\_r[[SUSv4]](#ID_STD_46_SUSV4) |
| argz\_delete[[LSB]](#ID_STD_46_LSB) | iswalnum\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) | strtol[[SUSv4]](#ID_STD_46_SUSV4) |
| argz\_extract[[LSB]](#ID_STD_46_LSB) | iswalpha[[SUSv4]](#ID_STD_46_SUSV4) | strtold[[SUSv4]](#ID_STD_46_SUSV4) |
| argz\_insert[[LSB]](#ID_STD_46_LSB) | iswalpha\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) | strtoll[[SUSv4]](#ID_STD_46_SUSV4) |
| argz\_next[[LSB]](#ID_STD_46_LSB) | iswblank[[SUSv4]](#ID_STD_46_SUSV4) | strtoq[[LSB]](#ID_STD_46_LSB) |
| argz\_replace[[LSB]](#ID_STD_46_LSB) | iswblank\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) | strtoul[[SUSv4]](#ID_STD_46_SUSV4) |
| argz\_stringify[[LSB]](#ID_STD_46_LSB) | iswcntrl[[SUSv4]](#ID_STD_46_SUSV4) | strtoull[[SUSv4]](#ID_STD_46_SUSV4) |
| asctime[[SUSv4]](#ID_STD_46_SUSV4) | iswcntrl\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) | strtoumax[[SUSv4]](#ID_STD_46_SUSV4) |
| asctime\_r[[SUSv4]](#ID_STD_46_SUSV4) | iswctype[[SUSv4]](#ID_STD_46_SUSV4) | strtouq[[LSB]](#ID_STD_46_LSB) |
| asprintf[[LSB]](#ID_STD_46_LSB) | iswctype\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) | strxfrm[[SUSv4]](#ID_STD_46_SUSV4) |
| atof[[SUSv4]](#ID_STD_46_SUSV4) | iswdigit[[SUSv4]](#ID_STD_46_SUSV4) | strxfrm\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) |
| atoi[[SUSv4]](#ID_STD_46_SUSV4) | iswdigit\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) | svc\_getreqset[[SVID.4]](#ID_STD_46_SVID_46_4) |
| atol[[SUSv4]](#ID_STD_46_SUSV4) | iswgraph[[SUSv4]](#ID_STD_46_SUSV4) | svc\_register[[LSB]](#ID_STD_46_LSB) |
| atoll[[SUSv4]](#ID_STD_46_SUSV4) | iswgraph\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) | svc\_run[[LSB]](#ID_STD_46_LSB) |
| authnone\_create[[SVID.4]](#ID_STD_46_SVID_46_4) | iswlower[[SUSv4]](#ID_STD_46_SUSV4) | svc\_sendreply[[LSB]](#ID_STD_46_LSB) |
| backtrace[[LSB]](#ID_STD_46_LSB) | iswlower\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) | svcerr\_auth[[SVID.4]](#ID_STD_46_SVID_46_4) |
| backtrace\_symbols[[LSB]](#ID_STD_46_LSB) | iswprint[[SUSv4]](#ID_STD_46_SUSV4) | svcerr\_decode[[SVID.4]](#ID_STD_46_SVID_46_4) |
| backtrace\_symbols\_fd[[LSB]](#ID_STD_46_LSB) | iswprint\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) | svcerr\_noproc[[SVID.4]](#ID_STD_46_SVID_46_4) |
| basename[[LSB]](#ID_STD_46_LSB) | iswpunct[[SUSv4]](#ID_STD_46_SUSV4) | svcerr\_noprog[[SVID.4]](#ID_STD_46_SVID_46_4) |
| bcmp[[SUSv3]](#ID_STD_46_SUSV3) | iswpunct\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) | svcerr\_progvers[[SVID.4]](#ID_STD_46_SVID_46_4) |
| bcopy[[SUSv3]](#ID_STD_46_SUSV3) | iswspace[[SUSv4]](#ID_STD_46_SUSV4) | svcerr\_systemerr[[SVID.4]](#ID_STD_46_SVID_46_4) |
| bind[[SUSv4]](#ID_STD_46_SUSV4) | iswspace\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) | svcerr\_weakauth[[SVID.4]](#ID_STD_46_SVID_46_4) |
| bind\_textdomain\_codeset[[LSB]](#ID_STD_46_LSB) | iswupper[[SUSv4]](#ID_STD_46_SUSV4) | svcfd\_create[[RPC + XDR]](#ID_STD_46_RPC_46_XDR) |
| bindresvport[[LSB]](#ID_STD_46_LSB) | iswupper\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) | svcraw\_create[[RPC + XDR]](#ID_STD_46_RPC_46_XDR) |
| bindtextdomain[[LSB]](#ID_STD_46_LSB) | iswxdigit[[SUSv4]](#ID_STD_46_SUSV4) | svctcp\_create[[LSB]](#ID_STD_46_LSB) |
| brk[[SUSv2]](#ID_STD_46_SUSV2) | iswxdigit\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) | svcudp\_create[[LSB]](#ID_STD_46_LSB) |
| bsd\_signal[[SUSv3]](#ID_STD_46_SUSV3) | isxdigit[[SUSv4]](#ID_STD_46_SUSV4) | swab[[SUSv4]](#ID_STD_46_SUSV4) |
| bsearch[[SUSv4]](#ID_STD_46_SUSV4) | isxdigit\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) | swapcontext[[SUSv3]](#ID_STD_46_SUSV3) |
| btowc[[SUSv4]](#ID_STD_46_SUSV4) | jrand48[[SUSv4]](#ID_STD_46_SUSV4) | swprintf[[SUSv4]](#ID_STD_46_SUSV4) |
| bzero[[SUSv3]](#ID_STD_46_SUSV3) | jrand48\_r[[LSB]](#ID_STD_46_LSB) | swscanf[[LSB]](#ID_STD_46_LSB) |
| calloc[[SUSv4]](#ID_STD_46_SUSV4) | key\_decryptsession[[SVID.4]](#ID_STD_46_SVID_46_4) | symlink[[SUSv4]](#ID_STD_46_SUSV4) |
| callrpc[[RPC + XDR]](#ID_STD_46_RPC_46_XDR) | kill[[LSB]](#ID_STD_46_LSB) | symlinkat(GLIBC\_2.4)[[SUSv4]](#ID_STD_46_SUSV4) |
| catclose[[SUSv4]](#ID_STD_46_SUSV4) | killpg[[SUSv4]](#ID_STD_46_SUSV4) | sync[[SUSv4]](#ID_STD_46_SUSV4) |
| catgets[[SUSv4]](#ID_STD_46_SUSV4) | l64a[[SUSv4]](#ID_STD_46_SUSV4) | sysconf[[LSB]](#ID_STD_46_LSB) |
| catopen[[SUSv4]](#ID_STD_46_SUSV4) | labs[[SUSv4]](#ID_STD_46_SUSV4) | sysinfo[[LSB]](#ID_STD_46_LSB) |
| cfgetispeed[[SUSv4]](#ID_STD_46_SUSV4) | lchown[[SUSv4]](#ID_STD_46_SUSV4) | syslog[[SUSv4]](#ID_STD_46_SUSV4) |
| cfgetospeed[[SUSv4]](#ID_STD_46_SUSV4) | lcong48[[SUSv4]](#ID_STD_46_SUSV4) | system[[LSB]](#ID_STD_46_LSB) |
| cfmakeraw[[LSB]](#ID_STD_46_LSB) | lcong48\_r[[LSB]](#ID_STD_46_LSB) | tcdrain[[SUSv4]](#ID_STD_46_SUSV4) |
| cfsetispeed[[SUSv4]](#ID_STD_46_SUSV4) | ldiv[[SUSv4]](#ID_STD_46_SUSV4) | tcflow[[SUSv4]](#ID_STD_46_SUSV4) |
| cfsetospeed[[SUSv4]](#ID_STD_46_SUSV4) | lfind[[SUSv4]](#ID_STD_46_SUSV4) | tcflush[[SUSv4]](#ID_STD_46_SUSV4) |
| cfsetspeed[[LSB]](#ID_STD_46_LSB) | link[[LSB]](#ID_STD_46_LSB) | tcgetattr[[SUSv4]](#ID_STD_46_SUSV4) |
| chdir[[SUSv4]](#ID_STD_46_SUSV4) | linkat(GLIBC\_2.4)[[SUSv4]](#ID_STD_46_SUSV4) | tcgetpgrp[[SUSv4]](#ID_STD_46_SUSV4) |
| chmod[[SUSv4]](#ID_STD_46_SUSV4) | listen[[SUSv4]](#ID_STD_46_SUSV4) | tcgetsid[[SUSv4]](#ID_STD_46_SUSV4) |
| chown[[SUSv4]](#ID_STD_46_SUSV4) | llabs[[SUSv4]](#ID_STD_46_SUSV4) | tcsendbreak[[SUSv4]](#ID_STD_46_SUSV4) |
| chroot[[SUSv2]](#ID_STD_46_SUSV2) | lldiv[[SUSv4]](#ID_STD_46_SUSV4) | tcsetattr[[SUSv4]](#ID_STD_46_SUSV4) |
| clearerr[[SUSv4]](#ID_STD_46_SUSV4) | localeconv[[SUSv4]](#ID_STD_46_SUSV4) | tcsetpgrp[[SUSv4]](#ID_STD_46_SUSV4) |
| clearerr\_unlocked[[LSB]](#ID_STD_46_LSB) | localtime[[SUSv4]](#ID_STD_46_SUSV4) | tdelete[[SUSv4]](#ID_STD_46_SUSV4) |
| clnt\_create[[SVID.4]](#ID_STD_46_SVID_46_4) | localtime\_r[[SUSv4]](#ID_STD_46_SUSV4) | telldir[[SUSv4]](#ID_STD_46_SUSV4) |
| clnt\_pcreateerror[[SVID.4]](#ID_STD_46_SVID_46_4) | lockf[[SUSv4]](#ID_STD_46_SUSV4) | tempnam[[SUSv4]](#ID_STD_46_SUSV4) |
| clnt\_perrno[[SVID.4]](#ID_STD_46_SVID_46_4) | lockf64[[LFS]](#ID_STD_46_LFS) | textdomain[[LSB]](#ID_STD_46_LSB) |
| clnt\_perror[[SVID.4]](#ID_STD_46_SVID_46_4) | longjmp[[SUSv4]](#ID_STD_46_SUSV4) | tfind[[SUSv4]](#ID_STD_46_SUSV4) |
| clnt\_spcreateerror[[SVID.4]](#ID_STD_46_SVID_46_4) | lrand48[[SUSv4]](#ID_STD_46_SUSV4) | time[[SUSv4]](#ID_STD_46_SUSV4) |
| clnt\_sperrno[[SVID.4]](#ID_STD_46_SVID_46_4) | lrand48\_r[[LSB]](#ID_STD_46_LSB) | times[[SUSv4]](#ID_STD_46_SUSV4) |
| clnt\_sperror[[SVID.4]](#ID_STD_46_SVID_46_4) | lsearch[[SUSv4]](#ID_STD_46_SUSV4) | tmpfile[[SUSv4]](#ID_STD_46_SUSV4) |
| clntraw\_create[[RPC + XDR]](#ID_STD_46_RPC_46_XDR) | lseek[[SUSv4]](#ID_STD_46_SUSV4) | tmpfile64[[LFS]](#ID_STD_46_LFS) |
| clnttcp\_create[[RPC + XDR]](#ID_STD_46_RPC_46_XDR) | lseek64[[LFS]](#ID_STD_46_LFS) | tmpnam[[SUSv4]](#ID_STD_46_SUSV4) |
| clntudp\_bufcreate[[RPC + XDR]](#ID_STD_46_RPC_46_XDR) | lutimes(GLIBC\_2.3)[[LSB]](#ID_STD_46_LSB) | toascii[[SUSv4]](#ID_STD_46_SUSV4) |
| clntudp\_create[[RPC + XDR]](#ID_STD_46_RPC_46_XDR) | makecontext[[SUSv3]](#ID_STD_46_SUSV3) | tolower[[SUSv4]](#ID_STD_46_SUSV4) |
| clock[[SUSv4]](#ID_STD_46_SUSV4) | malloc[[SUSv4]](#ID_STD_46_SUSV4) | tolower\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) |
| close[[SUSv4]](#ID_STD_46_SUSV4) | mblen[[SUSv4]](#ID_STD_46_SUSV4) | toupper[[SUSv4]](#ID_STD_46_SUSV4) |
| closedir[[SUSv4]](#ID_STD_46_SUSV4) | mbrlen[[SUSv4]](#ID_STD_46_SUSV4) | toupper\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) |
| closelog[[SUSv4]](#ID_STD_46_SUSV4) | mbrtowc[[SUSv4]](#ID_STD_46_SUSV4) | towctrans[[SUSv4]](#ID_STD_46_SUSV4) |
| confstr[[SUSv4]](#ID_STD_46_SUSV4) | mbsinit[[SUSv4]](#ID_STD_46_SUSV4) | towctrans\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) |
| connect[[SUSv4]](#ID_STD_46_SUSV4) | mbsnrtowcs[[SUSv4]](#ID_STD_46_SUSV4) | towlower[[SUSv4]](#ID_STD_46_SUSV4) |
| creat[[SUSv4]](#ID_STD_46_SUSV4) | mbsrtowcs[[SUSv4]](#ID_STD_46_SUSV4) | towlower\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) |
| creat64[[LFS]](#ID_STD_46_LFS) | mbstowcs[[SUSv4]](#ID_STD_46_SUSV4) | towupper[[SUSv4]](#ID_STD_46_SUSV4) |
| ctermid[[SUSv4]](#ID_STD_46_SUSV4) | mbtowc[[SUSv4]](#ID_STD_46_SUSV4) | towupper\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) |
| ctime[[SUSv4]](#ID_STD_46_SUSV4) | memccpy[[SUSv4]](#ID_STD_46_SUSV4) | truncate[[SUSv4]](#ID_STD_46_SUSV4) |
| ctime\_r[[SUSv4]](#ID_STD_46_SUSV4) | memchr[[SUSv4]](#ID_STD_46_SUSV4) | truncate64[[LFS]](#ID_STD_46_LFS) |
| cuserid[[SUSv2]](#ID_STD_46_SUSV2) | memcmp[[SUSv4]](#ID_STD_46_SUSV4) | tsearch[[SUSv4]](#ID_STD_46_SUSV4) |
| daemon[[LSB]](#ID_STD_46_LSB) | memcpy[[SUSv4]](#ID_STD_46_SUSV4) | ttyname[[SUSv4]](#ID_STD_46_SUSV4) |
| dcgettext[[LSB]](#ID_STD_46_LSB) | memmem[[LSB]](#ID_STD_46_LSB) | ttyname\_r[[SUSv4]](#ID_STD_46_SUSV4) |
| dcngettext[[LSB]](#ID_STD_46_LSB) | memmove[[SUSv4]](#ID_STD_46_SUSV4) | twalk[[SUSv4]](#ID_STD_46_SUSV4) |
| dgettext[[LSB]](#ID_STD_46_LSB) | memrchr[[LSB]](#ID_STD_46_LSB) | tzset[[SUSv4]](#ID_STD_46_SUSV4) |
| difftime[[SUSv4]](#ID_STD_46_SUSV4) | memset[[SUSv4]](#ID_STD_46_SUSV4) | ualarm[[SUSv3]](#ID_STD_46_SUSV3) |
| dirfd[[SUSv4]](#ID_STD_46_SUSV4) | mkdir[[SUSv4]](#ID_STD_46_SUSV4) | ulimit[[SUSv4]](#ID_STD_46_SUSV4) |
| dirname[[SUSv4]](#ID_STD_46_SUSV4) | mkdirat(GLIBC\_2.4)[[SUSv4]](#ID_STD_46_SUSV4) | umask[[SUSv4]](#ID_STD_46_SUSV4) |
| div[[SUSv4]](#ID_STD_46_SUSV4) | mkdtemp[[SUSv4]](#ID_STD_46_SUSV4) | uname[[SUSv4]](#ID_STD_46_SUSV4) |
| dl\_iterate\_phdr[[LSB]](#ID_STD_46_LSB) | mkfifo[[SUSv4]](#ID_STD_46_SUSV4) | ungetc[[SUSv4]](#ID_STD_46_SUSV4) |
| dngettext[[LSB]](#ID_STD_46_LSB) | mkfifoat(GLIBC\_2.4)[[SUSv4]](#ID_STD_46_SUSV4) | ungetwc[[SUSv4]](#ID_STD_46_SUSV4) |
| dprintf[[SUSv4]](#ID_STD_46_SUSV4) | mkstemp[[SUSv4]](#ID_STD_46_SUSV4) | unlink[[LSB]](#ID_STD_46_LSB) |
| drand48[[SUSv4]](#ID_STD_46_SUSV4) | mkstemp64[[LSB]](#ID_STD_46_LSB) | unlinkat(GLIBC\_2.4)[[SUSv4]](#ID_STD_46_SUSV4) |
| drand48\_r[[LSB]](#ID_STD_46_LSB) | mktemp[[SUSv3]](#ID_STD_46_SUSV3) | unlockpt[[SUSv4]](#ID_STD_46_SUSV4) |
| dup[[SUSv4]](#ID_STD_46_SUSV4) | mktime[[SUSv4]](#ID_STD_46_SUSV4) | unsetenv[[SUSv4]](#ID_STD_46_SUSV4) |
| dup2[[SUSv4]](#ID_STD_46_SUSV4) | mlock[[SUSv4]](#ID_STD_46_SUSV4) | uselocale(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) |
| duplocale(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) | mlockall[[SUSv4]](#ID_STD_46_SUSV4) | usleep[[SUSv3]](#ID_STD_46_SUSV3) |
| ecvt[[SUSv3]](#ID_STD_46_SUSV3) | mmap[[SUSv4]](#ID_STD_46_SUSV4) | utime[[SUSv4]](#ID_STD_46_SUSV4) |
| endgrent[[SUSv4]](#ID_STD_46_SUSV4) | mmap64[[LFS]](#ID_STD_46_LFS) | utimensat(GLIBC\_2.6)[[SUSv4]](#ID_STD_46_SUSV4) |
| endprotoent[[SUSv4]](#ID_STD_46_SUSV4) | mprotect[[SUSv4]](#ID_STD_46_SUSV4) | utimes[[SUSv4]](#ID_STD_46_SUSV4) |
| endpwent[[SUSv4]](#ID_STD_46_SUSV4) | mrand48[[SUSv4]](#ID_STD_46_SUSV4) | utmpname[[LSB]](#ID_STD_46_LSB) |
| endservent[[SUSv4]](#ID_STD_46_SUSV4) | mrand48\_r[[LSB]](#ID_STD_46_LSB) | vasprintf[[LSB]](#ID_STD_46_LSB) |
| endutent[[LSB]](#ID_STD_46_LSB) | mremap[[LSB]](#ID_STD_46_LSB) | vdprintf[[SUSv4]](#ID_STD_46_SUSV4) |
| endutxent[[SUSv4]](#ID_STD_46_SUSV4) | msgctl[[SUSv4]](#ID_STD_46_SUSV4) | verrx[[LSB]](#ID_STD_46_LSB) |
| envz\_add[[LSB]](#ID_STD_46_LSB) | msgget[[SUSv4]](#ID_STD_46_SUSV4) | vfork[[SUSv3]](#ID_STD_46_SUSV3) |
| envz\_entry[[LSB]](#ID_STD_46_LSB) | msgrcv[[SUSv4]](#ID_STD_46_SUSV4) | vfprintf[[SUSv4]](#ID_STD_46_SUSV4) |
| envz\_get[[LSB]](#ID_STD_46_LSB) | msgsnd[[SUSv4]](#ID_STD_46_SUSV4) | vfscanf[[LSB]](#ID_STD_46_LSB) |
| envz\_merge[[LSB]](#ID_STD_46_LSB) | msync[[SUSv4]](#ID_STD_46_SUSV4) | vfwprintf[[SUSv4]](#ID_STD_46_SUSV4) |
| envz\_remove[[LSB]](#ID_STD_46_LSB) | munlock[[SUSv4]](#ID_STD_46_SUSV4) | vfwscanf[[LSB]](#ID_STD_46_LSB) |
| envz\_strip[[LSB]](#ID_STD_46_LSB) | munlockall[[SUSv4]](#ID_STD_46_SUSV4) | vprintf[[SUSv4]](#ID_STD_46_SUSV4) |
| epoll\_create(GLIBC\_2.3.2)[[LSB]](#ID_STD_46_LSB) | munmap[[SUSv4]](#ID_STD_46_SUSV4) | vscanf[[LSB]](#ID_STD_46_LSB) |
| epoll\_ctl(GLIBC\_2.3.2)[[LSB]](#ID_STD_46_LSB) | nanosleep[[SUSv4]](#ID_STD_46_SUSV4) | vsnprintf[[SUSv4]](#ID_STD_46_SUSV4) |
| epoll\_wait(GLIBC\_2.3.2)[[LSB]](#ID_STD_46_LSB) | newlocale(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) | vsprintf[[SUSv4]](#ID_STD_46_SUSV4) |
| erand48[[SUSv4]](#ID_STD_46_SUSV4) | nftw[[SUSv4]](#ID_STD_46_SUSV4) | vsscanf[[LSB]](#ID_STD_46_LSB) |
| erand48\_r[[LSB]](#ID_STD_46_LSB) | nftw64[[LFS]](#ID_STD_46_LFS) | vswprintf[[SUSv4]](#ID_STD_46_SUSV4) |
| err[[LSB]](#ID_STD_46_LSB) | ngettext[[LSB]](#ID_STD_46_LSB) | vswscanf[[LSB]](#ID_STD_46_LSB) |
| error[[LSB]](#ID_STD_46_LSB) | nice[[SUSv4]](#ID_STD_46_SUSV4) | vsyslog[[LSB]](#ID_STD_46_LSB) |
| errx[[LSB]](#ID_STD_46_LSB) | nl\_langinfo[[SUSv4]](#ID_STD_46_SUSV4) | vwprintf[[SUSv4]](#ID_STD_46_SUSV4) |
| execl[[SUSv4]](#ID_STD_46_SUSV4) | nrand48[[SUSv4]](#ID_STD_46_SUSV4) | vwscanf[[LSB]](#ID_STD_46_LSB) |
| execle[[SUSv4]](#ID_STD_46_SUSV4) | nrand48\_r[[LSB]](#ID_STD_46_LSB) | wait[[SUSv4]](#ID_STD_46_SUSV4) |
| execlp[[SUSv4]](#ID_STD_46_SUSV4) | ntohl[[SUSv4]](#ID_STD_46_SUSV4) | wait4[[LSB]](#ID_STD_46_LSB) |
| execv[[SUSv4]](#ID_STD_46_SUSV4) | ntohs[[SUSv4]](#ID_STD_46_SUSV4) | waitid[[SUSv4]](#ID_STD_46_SUSV4) |
| execve[[SUSv4]](#ID_STD_46_SUSV4) | open[[SUSv4]](#ID_STD_46_SUSV4) | waitpid[[SUSv4]](#ID_STD_46_SUSV4) |
| execvp[[SUSv4]](#ID_STD_46_SUSV4) | open64[[LFS]](#ID_STD_46_LFS) | warn[[LSB]](#ID_STD_46_LSB) |
| exit[[SUSv4]](#ID_STD_46_SUSV4) | open\_memstream[[SUSv4]](#ID_STD_46_SUSV4) | warnx[[LSB]](#ID_STD_46_LSB) |
| faccessat(GLIBC\_2.4)[[SUSv4]](#ID_STD_46_SUSV4) | open\_wmemstream(GLIBC\_2.4)[[SUSv4]](#ID_STD_46_SUSV4) | wcpcpy[[SUSv4]](#ID_STD_46_SUSV4) |
| fchdir[[SUSv4]](#ID_STD_46_SUSV4) | openat(GLIBC\_2.4)[[SUSv4]](#ID_STD_46_SUSV4) | wcpncpy[[SUSv4]](#ID_STD_46_SUSV4) |
| fchmod[[SUSv4]](#ID_STD_46_SUSV4) | openat64(GLIBC\_2.4)[[LSB]](#ID_STD_46_LSB) | wcrtomb[[SUSv4]](#ID_STD_46_SUSV4) |
| fchmodat(GLIBC\_2.4)[[SUSv4]](#ID_STD_46_SUSV4) | opendir[[SUSv4]](#ID_STD_46_SUSV4) | wcscasecmp[[SUSv4]](#ID_STD_46_SUSV4) |
| fchown[[SUSv4]](#ID_STD_46_SUSV4) | openlog[[SUSv4]](#ID_STD_46_SUSV4) | wcscasecmp\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) |
| fchownat(GLIBC\_2.4)[[SUSv4]](#ID_STD_46_SUSV4) | pathconf[[SUSv4]](#ID_STD_46_SUSV4) | wcscat[[SUSv4]](#ID_STD_46_SUSV4) |
| fclose[[SUSv4]](#ID_STD_46_SUSV4) | pause[[SUSv4]](#ID_STD_46_SUSV4) | wcschr[[SUSv4]](#ID_STD_46_SUSV4) |
| fcntl[[LSB]](#ID_STD_46_LSB) | pclose[[SUSv4]](#ID_STD_46_SUSV4) | wcscmp[[SUSv4]](#ID_STD_46_SUSV4) |
| fcvt[[SUSv3]](#ID_STD_46_SUSV3) | perror[[SUSv4]](#ID_STD_46_SUSV4) | wcscoll[[SUSv4]](#ID_STD_46_SUSV4) |
| fdatasync[[SUSv4]](#ID_STD_46_SUSV4) | pipe[[SUSv4]](#ID_STD_46_SUSV4) | wcscoll\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) |
| fdopen[[SUSv4]](#ID_STD_46_SUSV4) | pmap\_getport[[LSB]](#ID_STD_46_LSB) | wcscpy[[SUSv4]](#ID_STD_46_SUSV4) |
| fdopendir(GLIBC\_2.4)[[SUSv4]](#ID_STD_46_SUSV4) | pmap\_set[[LSB]](#ID_STD_46_LSB) | wcscspn[[SUSv4]](#ID_STD_46_SUSV4) |
| feof[[SUSv4]](#ID_STD_46_SUSV4) | pmap\_unset[[LSB]](#ID_STD_46_LSB) | wcsdup[[SUSv4]](#ID_STD_46_SUSV4) |
| feof\_unlocked[[LSB]](#ID_STD_46_LSB) | poll[[SUSv4]](#ID_STD_46_SUSV4) | wcsftime[[SUSv4]](#ID_STD_46_SUSV4) |
| ferror[[SUSv4]](#ID_STD_46_SUSV4) | popen[[SUSv4]](#ID_STD_46_SUSV4) | wcslen[[SUSv4]](#ID_STD_46_SUSV4) |
| ferror\_unlocked[[LSB]](#ID_STD_46_LSB) | posix\_fadvise[[SUSv4]](#ID_STD_46_SUSV4) | wcsncasecmp[[SUSv4]](#ID_STD_46_SUSV4) |
| fexecve[[SUSv4]](#ID_STD_46_SUSV4) | posix\_fadvise64[[LSB]](#ID_STD_46_LSB) | wcsncasecmp\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) |
| fflush[[SUSv4]](#ID_STD_46_SUSV4) | posix\_fallocate[[SUSv4]](#ID_STD_46_SUSV4) | wcsncat[[SUSv4]](#ID_STD_46_SUSV4) |
| fflush\_unlocked[[LSB]](#ID_STD_46_LSB) | posix\_fallocate64[[LSB]](#ID_STD_46_LSB) | wcsncmp[[SUSv4]](#ID_STD_46_SUSV4) |
| ffs[[SUSv4]](#ID_STD_46_SUSV4) | posix\_madvise[[SUSv4]](#ID_STD_46_SUSV4) | wcsncpy[[SUSv4]](#ID_STD_46_SUSV4) |
| fgetc[[SUSv4]](#ID_STD_46_SUSV4) | posix\_memalign[[SUSv4]](#ID_STD_46_SUSV4) | wcsnlen[[SUSv4]](#ID_STD_46_SUSV4) |
| fgetc\_unlocked[[LSB]](#ID_STD_46_LSB) | posix\_openpt[[SUSv4]](#ID_STD_46_SUSV4) | wcsnrtombs[[SUSv4]](#ID_STD_46_SUSV4) |
| fgetpos[[SUSv4]](#ID_STD_46_SUSV4) | posix\_spawn[[SUSv4]](#ID_STD_46_SUSV4) | wcspbrk[[SUSv4]](#ID_STD_46_SUSV4) |
| fgetpos64[[LFS]](#ID_STD_46_LFS) | posix\_spawn\_file\_actions\_addclose[[SUSv4]](#ID_STD_46_SUSV4) | wcsrchr[[SUSv4]](#ID_STD_46_SUSV4) |
| fgets[[SUSv4]](#ID_STD_46_SUSV4) | posix\_spawn\_file\_actions\_adddup2[[SUSv4]](#ID_STD_46_SUSV4) | wcsrtombs[[SUSv4]](#ID_STD_46_SUSV4) |
| fgets\_unlocked[[LSB]](#ID_STD_46_LSB) | posix\_spawn\_file\_actions\_addopen[[SUSv4]](#ID_STD_46_SUSV4) | wcsspn[[SUSv4]](#ID_STD_46_SUSV4) |
| fgetwc[[SUSv4]](#ID_STD_46_SUSV4) | posix\_spawn\_file\_actions\_destroy[[SUSv4]](#ID_STD_46_SUSV4) | wcsstr[[SUSv4]](#ID_STD_46_SUSV4) |
| fgetwc\_unlocked[[LSB]](#ID_STD_46_LSB) | posix\_spawn\_file\_actions\_init[[SUSv4]](#ID_STD_46_SUSV4) | wcstod[[SUSv4]](#ID_STD_46_SUSV4) |
| fgetws[[SUSv4]](#ID_STD_46_SUSV4) | posix\_spawnattr\_destroy[[SUSv4]](#ID_STD_46_SUSV4) | wcstof[[SUSv4]](#ID_STD_46_SUSV4) |
| fgetws\_unlocked[[LSB]](#ID_STD_46_LSB) | posix\_spawnattr\_getflags[[SUSv4]](#ID_STD_46_SUSV4) | wcstoimax[[SUSv4]](#ID_STD_46_SUSV4) |
| fileno[[SUSv4]](#ID_STD_46_SUSV4) | posix\_spawnattr\_getpgroup[[SUSv4]](#ID_STD_46_SUSV4) | wcstok[[SUSv4]](#ID_STD_46_SUSV4) |
| fileno\_unlocked[[LSB]](#ID_STD_46_LSB) | posix\_spawnattr\_getschedparam[[SUSv4]](#ID_STD_46_SUSV4) | wcstol[[SUSv4]](#ID_STD_46_SUSV4) |
| flock[[LSB]](#ID_STD_46_LSB) | posix\_spawnattr\_getschedpolicy[[SUSv4]](#ID_STD_46_SUSV4) | wcstold[[SUSv4]](#ID_STD_46_SUSV4) |
| flockfile[[SUSv4]](#ID_STD_46_SUSV4) | posix\_spawnattr\_getsigdefault[[SUSv4]](#ID_STD_46_SUSV4) | wcstoll[[SUSv4]](#ID_STD_46_SUSV4) |
| fmemopen[[SUSv4]](#ID_STD_46_SUSV4) | posix\_spawnattr\_getsigmask[[SUSv4]](#ID_STD_46_SUSV4) | wcstombs[[SUSv4]](#ID_STD_46_SUSV4) |
| fmtmsg[[SUSv4]](#ID_STD_46_SUSV4) | posix\_spawnattr\_init[[SUSv4]](#ID_STD_46_SUSV4) | wcstoq[[LSB]](#ID_STD_46_LSB) |
| fnmatch[[LSB]](#ID_STD_46_LSB) | posix\_spawnattr\_setflags[[SUSv4]](#ID_STD_46_SUSV4) | wcstoul[[SUSv4]](#ID_STD_46_SUSV4) |
| fopen[[SUSv4]](#ID_STD_46_SUSV4) | posix\_spawnattr\_setpgroup[[SUSv4]](#ID_STD_46_SUSV4) | wcstoull[[SUSv4]](#ID_STD_46_SUSV4) |
| fopen64[[LFS]](#ID_STD_46_LFS) | posix\_spawnattr\_setschedparam[[SUSv4]](#ID_STD_46_SUSV4) | wcstoumax[[SUSv4]](#ID_STD_46_SUSV4) |
| fork[[SUSv4]](#ID_STD_46_SUSV4) | posix\_spawnattr\_setschedpolicy[[SUSv4]](#ID_STD_46_SUSV4) | wcstouq[[LSB]](#ID_STD_46_LSB) |
| fpathconf[[SUSv4]](#ID_STD_46_SUSV4) | posix\_spawnattr\_setsigdefault[[SUSv4]](#ID_STD_46_SUSV4) | wcswcs[[SUSv3]](#ID_STD_46_SUSV3) |
| fprintf[[SUSv4]](#ID_STD_46_SUSV4) | posix\_spawnattr\_setsigmask[[SUSv4]](#ID_STD_46_SUSV4) | wcswidth[[SUSv4]](#ID_STD_46_SUSV4) |
| fputc[[SUSv4]](#ID_STD_46_SUSV4) | posix\_spawnp[[SUSv4]](#ID_STD_46_SUSV4) | wcsxfrm[[SUSv4]](#ID_STD_46_SUSV4) |
| fputc\_unlocked[[LSB]](#ID_STD_46_LSB) | pread[[SUSv4]](#ID_STD_46_SUSV4) | wcsxfrm\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) |
| fputs[[SUSv4]](#ID_STD_46_SUSV4) | pread64[[LSB]](#ID_STD_46_LSB) | wctob[[SUSv4]](#ID_STD_46_SUSV4) |
| fputs\_unlocked[[LSB]](#ID_STD_46_LSB) | printf[[SUSv4]](#ID_STD_46_SUSV4) | wctomb[[SUSv4]](#ID_STD_46_SUSV4) |
| fputwc[[SUSv4]](#ID_STD_46_SUSV4) | pselect[[SUSv4]](#ID_STD_46_SUSV4) | wctrans[[SUSv4]](#ID_STD_46_SUSV4) |
| fputwc\_unlocked[[LSB]](#ID_STD_46_LSB) | psiginfo(GLIBC\_2.10)[[SUSv4]](#ID_STD_46_SUSV4) | wctrans\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) |
| fputws[[SUSv4]](#ID_STD_46_SUSV4) | psignal[[SUSv4]](#ID_STD_46_SUSV4) | wctype[[SUSv4]](#ID_STD_46_SUSV4) |
| fputws\_unlocked[[LSB]](#ID_STD_46_LSB) | ptrace[[LSB]](#ID_STD_46_LSB) | wctype\_l(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) |
| fread[[SUSv4]](#ID_STD_46_SUSV4) | ptsname[[SUSv4]](#ID_STD_46_SUSV4) | wcwidth[[SUSv4]](#ID_STD_46_SUSV4) |
| fread\_unlocked[[LSB]](#ID_STD_46_LSB) | putc[[SUSv4]](#ID_STD_46_SUSV4) | wmemchr[[SUSv4]](#ID_STD_46_SUSV4) |
| free[[SUSv4]](#ID_STD_46_SUSV4) | putc\_unlocked[[SUSv4]](#ID_STD_46_SUSV4) | wmemcmp[[SUSv4]](#ID_STD_46_SUSV4) |
| freeaddrinfo[[SUSv4]](#ID_STD_46_SUSV4) | putchar[[SUSv4]](#ID_STD_46_SUSV4) | wmemcpy[[SUSv4]](#ID_STD_46_SUSV4) |
| freeifaddrs(GLIBC\_2.3)[[LSB]](#ID_STD_46_LSB) | putchar\_unlocked[[SUSv4]](#ID_STD_46_SUSV4) | wmemmove[[SUSv4]](#ID_STD_46_SUSV4) |
| freelocale(GLIBC\_2.3)[[SUSv4]](#ID_STD_46_SUSV4) | putenv[[SUSv4]](#ID_STD_46_SUSV4) | wmemset[[SUSv4]](#ID_STD_46_SUSV4) |
| freopen[[SUSv4]](#ID_STD_46_SUSV4) | puts[[SUSv4]](#ID_STD_46_SUSV4) | wordexp[[SUSv4]](#ID_STD_46_SUSV4) |
| freopen64[[LFS]](#ID_STD_46_LFS) | pututxline[[SUSv4]](#ID_STD_46_SUSV4) | wordfree[[SUSv4]](#ID_STD_46_SUSV4) |
| fscanf[[LSB]](#ID_STD_46_LSB) | putw[[SUSv2]](#ID_STD_46_SUSV2) | wprintf[[SUSv4]](#ID_STD_46_SUSV4) |
| fseek[[SUSv4]](#ID_STD_46_SUSV4) | putwc[[SUSv4]](#ID_STD_46_SUSV4) | write[[SUSv4]](#ID_STD_46_SUSV4) |
| fseeko[[SUSv4]](#ID_STD_46_SUSV4) | putwc\_unlocked[[LSB]](#ID_STD_46_LSB) | writev[[SUSv4]](#ID_STD_46_SUSV4) |
| fseeko64[[LFS]](#ID_STD_46_LFS) | putwchar[[SUSv4]](#ID_STD_46_SUSV4) | wscanf[[LSB]](#ID_STD_46_LSB) |
| fsetpos[[SUSv4]](#ID_STD_46_SUSV4) | putwchar\_unlocked[[LSB]](#ID_STD_46_LSB) | xdr\_accepted\_reply[[SVID.4]](#ID_STD_46_SVID_46_4) |
| fsetpos64[[LFS]](#ID_STD_46_LFS) | pwrite[[SUSv4]](#ID_STD_46_SUSV4) | xdr\_array[[SVID.4]](#ID_STD_46_SVID_46_4) |
| fstatfs[[LSB]](#ID_STD_46_LSB) | pwrite64[[LSB]](#ID_STD_46_LSB) | xdr\_bool[[SVID.4]](#ID_STD_46_SVID_46_4) |
| fstatfs64[[LSB]](#ID_STD_46_LSB) | qsort[[SUSv4]](#ID_STD_46_SUSV4) | xdr\_bytes[[SVID.4]](#ID_STD_46_SVID_46_4) |
| fstatvfs[[SUSv4]](#ID_STD_46_SUSV4) | raise[[SUSv4]](#ID_STD_46_SUSV4) | xdr\_callhdr[[SVID.4]](#ID_STD_46_SVID_46_4) |
| fstatvfs64[[LFS]](#ID_STD_46_LFS) | rand[[SUSv4]](#ID_STD_46_SUSV4) | xdr\_callmsg[[SVID.4]](#ID_STD_46_SVID_46_4) |
| fsync[[SUSv4]](#ID_STD_46_SUSV4) | rand\_r[[SUSv4]](#ID_STD_46_SUSV4) | xdr\_char[[SVID.4]](#ID_STD_46_SVID_46_4) |
| ftell[[SUSv4]](#ID_STD_46_SUSV4) | random[[SUSv4]](#ID_STD_46_SUSV4) | xdr\_double[[SVID.4]](#ID_STD_46_SVID_46_4) |
| ftello[[SUSv4]](#ID_STD_46_SUSV4) | random\_r[[LSB]](#ID_STD_46_LSB) | xdr\_enum[[SVID.4]](#ID_STD_46_SVID_46_4) |
| ftello64[[LFS]](#ID_STD_46_LFS) | read[[SUSv4]](#ID_STD_46_SUSV4) | xdr\_float[[SVID.4]](#ID_STD_46_SVID_46_4) |
| ftime[[SUSv3]](#ID_STD_46_SUSV3) | readdir[[SUSv4]](#ID_STD_46_SUSV4) | xdr\_free[[SVID.4]](#ID_STD_46_SVID_46_4) |
| ftok[[SUSv4]](#ID_STD_46_SUSV4) | readdir64[[LFS]](#ID_STD_46_LFS) | xdr\_int[[SVID.4]](#ID_STD_46_SVID_46_4) |
| ftruncate[[SUSv4]](#ID_STD_46_SUSV4) | readdir64\_r[[LSB]](#ID_STD_46_LSB) | xdr\_long[[SVID.4]](#ID_STD_46_SVID_46_4) |
| ftruncate64[[LFS]](#ID_STD_46_LFS) | readdir\_r[[SUSv4]](#ID_STD_46_SUSV4) | xdr\_opaque[[SVID.4]](#ID_STD_46_SVID_46_4) |
| ftrylockfile[[SUSv4]](#ID_STD_46_SUSV4) | readlink[[SUSv4]](#ID_STD_46_SUSV4) | xdr\_opaque\_auth[[SVID.4]](#ID_STD_46_SVID_46_4) |
| ftw[[SUSv4]](#ID_STD_46_SUSV4) | readlinkat(GLIBC\_2.4)[[SUSv4]](#ID_STD_46_SUSV4) | xdr\_pointer[[SVID.4]](#ID_STD_46_SVID_46_4) |
| ftw64[[LFS]](#ID_STD_46_LFS) | readv[[SUSv4]](#ID_STD_46_SUSV4) | xdr\_reference[[SVID.4]](#ID_STD_46_SVID_46_4) |
| funlockfile[[SUSv4]](#ID_STD_46_SUSV4) | realloc[[SUSv4]](#ID_STD_46_SUSV4) | xdr\_rejected\_reply[[SVID.4]](#ID_STD_46_SVID_46_4) |
| futimens(GLIBC\_2.6)[[SUSv4]](#ID_STD_46_SUSV4) | realpath[[SUSv4]](#ID_STD_46_SUSV4) | xdr\_replymsg[[SVID.4]](#ID_STD_46_SVID_46_4) |
| futimes(GLIBC\_2.3)[[LSB]](#ID_STD_46_LSB) | recv[[SUSv4]](#ID_STD_46_SUSV4) | xdr\_short[[SVID.4]](#ID_STD_46_SVID_46_4) |
| fwide[[SUSv4]](#ID_STD_46_SUSV4) | recvfrom[[SUSv4]](#ID_STD_46_SUSV4) | xdr\_string[[SVID.4]](#ID_STD_46_SVID_46_4) |
| fwprintf[[SUSv4]](#ID_STD_46_SUSV4) | recvmsg[[SUSv4]](#ID_STD_46_SUSV4) | xdr\_u\_char[[SVID.4]](#ID_STD_46_SVID_46_4) |
| fwrite[[SUSv4]](#ID_STD_46_SUSV4) | regcomp[[SUSv4]](#ID_STD_46_SUSV4) | xdr\_u\_int[[LSB]](#ID_STD_46_LSB) |
| fwrite\_unlocked[[LSB]](#ID_STD_46_LSB) | regerror[[SUSv4]](#ID_STD_46_SUSV4) | xdr\_u\_long[[SVID.4]](#ID_STD_46_SVID_46_4) |
| fwscanf[[LSB]](#ID_STD_46_LSB) | regexec[[LSB]](#ID_STD_46_LSB) | xdr\_u\_short[[SVID.4]](#ID_STD_46_SVID_46_4) |
| gai\_strerror[[SUSv4]](#ID_STD_46_SUSV4) | regfree[[SUSv4]](#ID_STD_46_SUSV4) | xdr\_union[[SVID.4]](#ID_STD_46_SVID_46_4) |
| gcvt[[SUSv3]](#ID_STD_46_SUSV3) | remove[[SUSv4]](#ID_STD_46_SUSV4) | xdr\_vector[[SVID.4]](#ID_STD_46_SVID_46_4) |
| getaddrinfo[[SUSv4]](#ID_STD_46_SUSV4) | remque[[SUSv4]](#ID_STD_46_SUSV4) | xdr\_void[[SVID.4]](#ID_STD_46_SVID_46_4) |
| getc[[SUSv4]](#ID_STD_46_SUSV4) | rename[[SUSv4]](#ID_STD_46_SUSV4) | xdr\_wrapstring[[SVID.4]](#ID_STD_46_SVID_46_4) |
| getc\_unlocked[[SUSv4]](#ID_STD_46_SUSV4) | renameat(GLIBC\_2.4)[[SUSv4]](#ID_STD_46_SUSV4) | xdrmem\_create[[SVID.4]](#ID_STD_46_SVID_46_4) |
| getchar[[SUSv4]](#ID_STD_46_SUSV4) | rewind[[SUSv4]](#ID_STD_46_SUSV4) | xdrrec\_create[[SVID.4]](#ID_STD_46_SVID_46_4) |
| getchar\_unlocked[[SUSv4]](#ID_STD_46_SUSV4) | rewinddir[[SUSv4]](#ID_STD_46_SUSV4) | xdrrec\_endofrecord[[RPC + XDR]](#ID_STD_46_RPC_46_XDR) |
| getcontext[[SUSv3]](#ID_STD_46_SUSV3) | rindex[[SUSv3]](#ID_STD_46_SUSV3) | xdrrec\_eof[[SVID.4]](#ID_STD_46_SVID_46_4) |
| getcwd[[LSB]](#ID_STD_46_LSB) | rmdir[[SUSv4]](#ID_STD_46_SUSV4) | xdrrec\_skiprecord[[RPC + XDR]](#ID_STD_46_RPC_46_XDR) |
| getdate[[SUSv4]](#ID_STD_46_SUSV4) | sbrk[[SUSv2]](#ID_STD_46_SUSV2) | xdrstdio\_create[[LSB]](#ID_STD_46_LSB) |

**Table A-2 libc Data Interfaces**

|  |  |  |
| --- | --- | --- |
| \_\_daylight[[LSB]](#ID_STD_46_LSB) | \_\_tzname[[LSB]](#ID_STD_46_LSB) | in6addr\_loopback[[SUSv3]](#ID_STD_46_SUSV3) |
| \_\_environ[[LSB]](#ID_STD_46_LSB) | \_sys\_errlist[[LSB]](#ID_STD_46_LSB) |  |
| \_\_timezone[[LSB]](#ID_STD_46_LSB) | in6addr\_any[[SUSv3]](#ID_STD_46_SUSV3) |  |

## **A.2 libcrypt**

The behavior of the interfaces in this library is specified by the following Standards.

|  |
| --- |
| [This Specification](#ID_STD_46_LSB) [LSB] |
| [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) [SUSv4] |

**Table A-3 libcrypt Function Interfaces**

|  |  |  |
| --- | --- | --- |
| crypt[[SUSv4]](#ID_STD_46_SUSV4) | encrypt[[SUSv4]](#ID_STD_46_SUSV4) | setkey[[SUSv4]](#ID_STD_46_SUSV4) |
| crypt\_r[[LSB]](#ID_STD_46_LSB) | encrypt\_r[[LSB]](#ID_STD_46_LSB) | setkey\_r[[LSB]](#ID_STD_46_LSB) |

## **A.3 libdl**

The behavior of the interfaces in this library is specified by the following Standards.

|  |
| --- |
| [This Specification](#ID_STD_46_LSB) [LSB] |
| [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) [SUSv4] |

**Table A-4 libdl Function Interfaces**

|  |  |  |
| --- | --- | --- |
| dladdr[[LSB]](#ID_STD_46_LSB) | dlerror[[SUSv4]](#ID_STD_46_SUSV4) | dlsym[[LSB]](#ID_STD_46_LSB) |
| dlclose[[SUSv4]](#ID_STD_46_SUSV4) | dlopen[[LSB]](#ID_STD_46_LSB) | dlvsym[[LSB]](#ID_STD_46_LSB) |

## **A.4 libgcc\_s**

The behavior of the interfaces in this library is specified by the following Standards.

|  |
| --- |
| [This Specification](#ID_STD_46_LSB) [LSB] |

**Table A-5 libgcc\_s Function Interfaces**

|  |  |  |
| --- | --- | --- |
| \_Unwind\_Backtrace[[LSB]](#ID_STD_46_LSB) | \_Unwind\_GetGR[[LSB]](#ID_STD_46_LSB) | \_Unwind\_RaiseException[[LSB]](#ID_STD_46_LSB) |
| \_Unwind\_DeleteException[[LSB]](#ID_STD_46_LSB) | \_Unwind\_GetIP[[LSB]](#ID_STD_46_LSB) | \_Unwind\_Resume[[LSB]](#ID_STD_46_LSB) |
| \_Unwind\_FindEnclosingFunction[[LSB]](#ID_STD_46_LSB) | \_Unwind\_GetIPInfo(GCC\_4.2.0)[[LSB]](#ID_STD_46_LSB) | \_Unwind\_Resume\_or\_Rethrow[[LSB]](#ID_STD_46_LSB) |
| \_Unwind\_ForcedUnwind[[LSB]](#ID_STD_46_LSB) | \_Unwind\_GetLanguageSpecificData[[LSB]](#ID_STD_46_LSB) | \_Unwind\_SetGR[[LSB]](#ID_STD_46_LSB) |
| \_Unwind\_GetCFA[[LSB]](#ID_STD_46_LSB) | \_Unwind\_GetRegionStart[[LSB]](#ID_STD_46_LSB) | \_Unwind\_SetIP[[LSB]](#ID_STD_46_LSB) |

## **A.5 libm**

The behavior of the interfaces in this library is specified by the following Standards.

|  |
| --- |
| [This Specification](#ID_STD_46_LSB) [LSB] |
| [POSIX 1003.1-2001 (ISO/IEC 9945-2003)](#ID_STD_46_SUSV3) [SUSv3] |
| [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) [SUSv4] |

**Table A-6 libm Function Interfaces**

|  |  |  |
| --- | --- | --- |
| \_\_finite[[LSB]](#ID_STD_46_LSB) | csinl[[SUSv4]](#ID_STD_46_SUSV4) | llroundf[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_finitef[[LSB]](#ID_STD_46_LSB) | csqrt[[SUSv4]](#ID_STD_46_SUSV4) | llroundl[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_finitel[[LSB]](#ID_STD_46_LSB) | csqrtf[[SUSv4]](#ID_STD_46_SUSV4) | log[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_fpclassify[[LSB]](#ID_STD_46_LSB) | csqrtl[[SUSv4]](#ID_STD_46_SUSV4) | log10[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_fpclassifyf[[LSB]](#ID_STD_46_LSB) | ctan[[SUSv4]](#ID_STD_46_SUSV4) | log10f[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_signbit[[LSB]](#ID_STD_46_LSB) | ctanf[[SUSv4]](#ID_STD_46_SUSV4) | log10l[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_signbitf[[LSB]](#ID_STD_46_LSB) | ctanh[[SUSv4]](#ID_STD_46_SUSV4) | log1p[[SUSv4]](#ID_STD_46_SUSV4) |
| acos[[SUSv4]](#ID_STD_46_SUSV4) | ctanhf[[SUSv4]](#ID_STD_46_SUSV4) | log1pf[[SUSv4]](#ID_STD_46_SUSV4) |
| acosf[[SUSv4]](#ID_STD_46_SUSV4) | ctanhl[[SUSv4]](#ID_STD_46_SUSV4) | log1pl[[SUSv4]](#ID_STD_46_SUSV4) |
| acosh[[SUSv4]](#ID_STD_46_SUSV4) | ctanl[[SUSv4]](#ID_STD_46_SUSV4) | log2[[SUSv4]](#ID_STD_46_SUSV4) |
| acoshf[[SUSv4]](#ID_STD_46_SUSV4) | drem[[LSB]](#ID_STD_46_LSB) | log2f[[SUSv4]](#ID_STD_46_SUSV4) |
| acoshl[[SUSv4]](#ID_STD_46_SUSV4) | dremf[[LSB]](#ID_STD_46_LSB) | log2l[[SUSv4]](#ID_STD_46_SUSV4) |
| acosl[[SUSv4]](#ID_STD_46_SUSV4) | dreml[[LSB]](#ID_STD_46_LSB) | logb[[SUSv4]](#ID_STD_46_SUSV4) |
| asin[[SUSv4]](#ID_STD_46_SUSV4) | erf[[SUSv4]](#ID_STD_46_SUSV4) | logbf[[SUSv4]](#ID_STD_46_SUSV4) |
| asinf[[SUSv4]](#ID_STD_46_SUSV4) | erfc[[SUSv4]](#ID_STD_46_SUSV4) | logbl[[SUSv4]](#ID_STD_46_SUSV4) |
| asinh[[SUSv4]](#ID_STD_46_SUSV4) | erfcf[[SUSv4]](#ID_STD_46_SUSV4) | logf[[SUSv4]](#ID_STD_46_SUSV4) |
| asinhf[[SUSv4]](#ID_STD_46_SUSV4) | erfcl[[SUSv4]](#ID_STD_46_SUSV4) | logl[[SUSv4]](#ID_STD_46_SUSV4) |
| asinhl[[SUSv4]](#ID_STD_46_SUSV4) | erff[[SUSv4]](#ID_STD_46_SUSV4) | lrint[[SUSv4]](#ID_STD_46_SUSV4) |
| asinl[[SUSv4]](#ID_STD_46_SUSV4) | erfl[[SUSv4]](#ID_STD_46_SUSV4) | lrintf[[SUSv4]](#ID_STD_46_SUSV4) |
| atan[[SUSv4]](#ID_STD_46_SUSV4) | exp[[SUSv4]](#ID_STD_46_SUSV4) | lrintl[[SUSv4]](#ID_STD_46_SUSV4) |
| atan2[[SUSv4]](#ID_STD_46_SUSV4) | exp10[[LSB]](#ID_STD_46_LSB) | lround[[SUSv4]](#ID_STD_46_SUSV4) |
| atan2f[[SUSv4]](#ID_STD_46_SUSV4) | exp10f[[LSB]](#ID_STD_46_LSB) | lroundf[[SUSv4]](#ID_STD_46_SUSV4) |
| atan2l[[SUSv4]](#ID_STD_46_SUSV4) | exp10l[[LSB]](#ID_STD_46_LSB) | lroundl[[SUSv4]](#ID_STD_46_SUSV4) |
| atanf[[SUSv4]](#ID_STD_46_SUSV4) | exp2[[SUSv4]](#ID_STD_46_SUSV4) | matherr[[LSB]](#ID_STD_46_LSB) |
| atanh[[SUSv4]](#ID_STD_46_SUSV4) | exp2f[[SUSv4]](#ID_STD_46_SUSV4) | modf[[SUSv4]](#ID_STD_46_SUSV4) |
| atanhf[[SUSv4]](#ID_STD_46_SUSV4) | expf[[SUSv4]](#ID_STD_46_SUSV4) | modff[[SUSv4]](#ID_STD_46_SUSV4) |
| atanhl[[SUSv4]](#ID_STD_46_SUSV4) | expl[[SUSv4]](#ID_STD_46_SUSV4) | modfl[[SUSv4]](#ID_STD_46_SUSV4) |
| atanl[[SUSv4]](#ID_STD_46_SUSV4) | expm1[[SUSv4]](#ID_STD_46_SUSV4) | nan[[SUSv4]](#ID_STD_46_SUSV4) |
| cabs[[SUSv4]](#ID_STD_46_SUSV4) | expm1f[[SUSv4]](#ID_STD_46_SUSV4) | nanf[[SUSv4]](#ID_STD_46_SUSV4) |
| cabsf[[SUSv4]](#ID_STD_46_SUSV4) | expm1l[[SUSv4]](#ID_STD_46_SUSV4) | nanl[[SUSv4]](#ID_STD_46_SUSV4) |
| cabsl[[SUSv4]](#ID_STD_46_SUSV4) | fabs[[SUSv4]](#ID_STD_46_SUSV4) | nearbyint[[SUSv4]](#ID_STD_46_SUSV4) |
| cacos[[SUSv4]](#ID_STD_46_SUSV4) | fabsf[[SUSv4]](#ID_STD_46_SUSV4) | nearbyintf[[SUSv4]](#ID_STD_46_SUSV4) |
| cacosf[[SUSv4]](#ID_STD_46_SUSV4) | fabsl[[SUSv4]](#ID_STD_46_SUSV4) | nearbyintl[[SUSv4]](#ID_STD_46_SUSV4) |
| cacosh[[SUSv4]](#ID_STD_46_SUSV4) | fdim[[SUSv4]](#ID_STD_46_SUSV4) | nextafter[[SUSv4]](#ID_STD_46_SUSV4) |
| cacoshf[[SUSv4]](#ID_STD_46_SUSV4) | fdimf[[SUSv4]](#ID_STD_46_SUSV4) | nextafterf[[SUSv4]](#ID_STD_46_SUSV4) |
| cacoshl[[SUSv4]](#ID_STD_46_SUSV4) | fdiml[[SUSv4]](#ID_STD_46_SUSV4) | nextafterl[[SUSv4]](#ID_STD_46_SUSV4) |
| cacosl[[SUSv4]](#ID_STD_46_SUSV4) | feclearexcept[[SUSv4]](#ID_STD_46_SUSV4) | nexttoward[[SUSv4]](#ID_STD_46_SUSV4) |
| carg[[SUSv4]](#ID_STD_46_SUSV4) | fedisableexcept[[LSB]](#ID_STD_46_LSB) | nexttowardf[[SUSv4]](#ID_STD_46_SUSV4) |
| cargf[[SUSv4]](#ID_STD_46_SUSV4) | feenableexcept[[LSB]](#ID_STD_46_LSB) | nexttowardl[[SUSv4]](#ID_STD_46_SUSV4) |
| cargl[[SUSv4]](#ID_STD_46_SUSV4) | fegetenv[[SUSv4]](#ID_STD_46_SUSV4) | pow[[SUSv4]](#ID_STD_46_SUSV4) |
| casin[[SUSv4]](#ID_STD_46_SUSV4) | fegetexcept[[LSB]](#ID_STD_46_LSB) | pow10[[LSB]](#ID_STD_46_LSB) |
| casinf[[SUSv4]](#ID_STD_46_SUSV4) | fegetexceptflag[[SUSv4]](#ID_STD_46_SUSV4) | pow10f[[LSB]](#ID_STD_46_LSB) |
| casinh[[SUSv4]](#ID_STD_46_SUSV4) | fegetround[[SUSv4]](#ID_STD_46_SUSV4) | pow10l[[LSB]](#ID_STD_46_LSB) |
| casinhf[[SUSv4]](#ID_STD_46_SUSV4) | feholdexcept[[SUSv4]](#ID_STD_46_SUSV4) | powf[[SUSv4]](#ID_STD_46_SUSV4) |
| casinhl[[SUSv4]](#ID_STD_46_SUSV4) | feraiseexcept[[SUSv4]](#ID_STD_46_SUSV4) | powl[[SUSv4]](#ID_STD_46_SUSV4) |
| casinl[[SUSv4]](#ID_STD_46_SUSV4) | fesetenv[[SUSv4]](#ID_STD_46_SUSV4) | remainder[[SUSv4]](#ID_STD_46_SUSV4) |
| catan[[SUSv4]](#ID_STD_46_SUSV4) | fesetexceptflag[[SUSv4]](#ID_STD_46_SUSV4) | remainderf[[SUSv4]](#ID_STD_46_SUSV4) |
| catanf[[SUSv4]](#ID_STD_46_SUSV4) | fesetround[[SUSv4]](#ID_STD_46_SUSV4) | remainderl[[SUSv4]](#ID_STD_46_SUSV4) |
| catanh[[SUSv4]](#ID_STD_46_SUSV4) | fetestexcept[[SUSv4]](#ID_STD_46_SUSV4) | remquo[[SUSv4]](#ID_STD_46_SUSV4) |
| catanhf[[SUSv4]](#ID_STD_46_SUSV4) | feupdateenv[[SUSv4]](#ID_STD_46_SUSV4) | remquof[[SUSv4]](#ID_STD_46_SUSV4) |
| catanhl[[SUSv4]](#ID_STD_46_SUSV4) | finite[[LSB]](#ID_STD_46_LSB) | remquol[[SUSv4]](#ID_STD_46_SUSV4) |
| catanl[[SUSv4]](#ID_STD_46_SUSV4) | finitef[[LSB]](#ID_STD_46_LSB) | rint[[SUSv4]](#ID_STD_46_SUSV4) |
| cbrt[[SUSv4]](#ID_STD_46_SUSV4) | finitel[[LSB]](#ID_STD_46_LSB) | rintf[[SUSv4]](#ID_STD_46_SUSV4) |
| cbrtf[[SUSv4]](#ID_STD_46_SUSV4) | floor[[SUSv4]](#ID_STD_46_SUSV4) | rintl[[SUSv4]](#ID_STD_46_SUSV4) |
| cbrtl[[SUSv4]](#ID_STD_46_SUSV4) | floorf[[SUSv4]](#ID_STD_46_SUSV4) | round[[SUSv4]](#ID_STD_46_SUSV4) |
| ccos[[SUSv4]](#ID_STD_46_SUSV4) | floorl[[SUSv4]](#ID_STD_46_SUSV4) | roundf[[SUSv4]](#ID_STD_46_SUSV4) |
| ccosf[[SUSv4]](#ID_STD_46_SUSV4) | fma[[SUSv4]](#ID_STD_46_SUSV4) | roundl[[SUSv4]](#ID_STD_46_SUSV4) |
| ccosh[[SUSv4]](#ID_STD_46_SUSV4) | fmaf[[SUSv4]](#ID_STD_46_SUSV4) | scalb[[SUSv3]](#ID_STD_46_SUSV3) |
| ccoshf[[SUSv4]](#ID_STD_46_SUSV4) | fmal[[SUSv4]](#ID_STD_46_SUSV4) | scalbf[[LSB]](#ID_STD_46_LSB) |
| ccoshl[[SUSv4]](#ID_STD_46_SUSV4) | fmax[[SUSv4]](#ID_STD_46_SUSV4) | scalbl[[LSB]](#ID_STD_46_LSB) |
| ccosl[[SUSv4]](#ID_STD_46_SUSV4) | fmaxf[[SUSv4]](#ID_STD_46_SUSV4) | scalbln[[SUSv4]](#ID_STD_46_SUSV4) |
| ceil[[SUSv4]](#ID_STD_46_SUSV4) | fmaxl[[SUSv4]](#ID_STD_46_SUSV4) | scalblnf[[SUSv4]](#ID_STD_46_SUSV4) |
| ceilf[[SUSv4]](#ID_STD_46_SUSV4) | fmin[[SUSv4]](#ID_STD_46_SUSV4) | scalblnl[[SUSv4]](#ID_STD_46_SUSV4) |
| ceill[[SUSv4]](#ID_STD_46_SUSV4) | fminf[[SUSv4]](#ID_STD_46_SUSV4) | scalbn[[SUSv4]](#ID_STD_46_SUSV4) |
| cexp[[SUSv4]](#ID_STD_46_SUSV4) | fminl[[SUSv4]](#ID_STD_46_SUSV4) | scalbnf[[SUSv4]](#ID_STD_46_SUSV4) |
| cexpf[[SUSv4]](#ID_STD_46_SUSV4) | fmod[[SUSv4]](#ID_STD_46_SUSV4) | scalbnl[[SUSv4]](#ID_STD_46_SUSV4) |
| cexpl[[SUSv4]](#ID_STD_46_SUSV4) | fmodf[[SUSv4]](#ID_STD_46_SUSV4) | significand[[LSB]](#ID_STD_46_LSB) |
| cimag[[SUSv4]](#ID_STD_46_SUSV4) | fmodl[[SUSv4]](#ID_STD_46_SUSV4) | significandf[[LSB]](#ID_STD_46_LSB) |
| cimagf[[SUSv4]](#ID_STD_46_SUSV4) | frexp[[SUSv4]](#ID_STD_46_SUSV4) | significandl[[LSB]](#ID_STD_46_LSB) |
| cimagl[[SUSv4]](#ID_STD_46_SUSV4) | frexpf[[SUSv4]](#ID_STD_46_SUSV4) | sin[[SUSv4]](#ID_STD_46_SUSV4) |
| clog[[SUSv4]](#ID_STD_46_SUSV4) | frexpl[[SUSv4]](#ID_STD_46_SUSV4) | sincos[[LSB]](#ID_STD_46_LSB) |
| clog10[[LSB]](#ID_STD_46_LSB) | gamma[[LSB]](#ID_STD_46_LSB) | sincosf[[LSB]](#ID_STD_46_LSB) |
| clog10f[[LSB]](#ID_STD_46_LSB) | gammaf[[LSB]](#ID_STD_46_LSB) | sincosl[[LSB]](#ID_STD_46_LSB) |
| clog10l[[LSB]](#ID_STD_46_LSB) | gammal[[LSB]](#ID_STD_46_LSB) | sinf[[SUSv4]](#ID_STD_46_SUSV4) |
| clogf[[SUSv4]](#ID_STD_46_SUSV4) | hypot[[SUSv4]](#ID_STD_46_SUSV4) | sinh[[SUSv4]](#ID_STD_46_SUSV4) |
| clogl[[SUSv4]](#ID_STD_46_SUSV4) | hypotf[[SUSv4]](#ID_STD_46_SUSV4) | sinhf[[SUSv4]](#ID_STD_46_SUSV4) |
| conj[[SUSv4]](#ID_STD_46_SUSV4) | hypotl[[SUSv4]](#ID_STD_46_SUSV4) | sinhl[[SUSv4]](#ID_STD_46_SUSV4) |
| conjf[[SUSv4]](#ID_STD_46_SUSV4) | ilogb[[SUSv4]](#ID_STD_46_SUSV4) | sinl[[SUSv4]](#ID_STD_46_SUSV4) |
| conjl[[SUSv4]](#ID_STD_46_SUSV4) | ilogbf[[SUSv4]](#ID_STD_46_SUSV4) | sqrt[[SUSv4]](#ID_STD_46_SUSV4) |
| copysign[[SUSv4]](#ID_STD_46_SUSV4) | ilogbl[[SUSv4]](#ID_STD_46_SUSV4) | sqrtf[[SUSv4]](#ID_STD_46_SUSV4) |
| copysignf[[SUSv4]](#ID_STD_46_SUSV4) | j0[[SUSv4]](#ID_STD_46_SUSV4) | sqrtl[[SUSv4]](#ID_STD_46_SUSV4) |
| copysignl[[SUSv4]](#ID_STD_46_SUSV4) | j0f[[LSB]](#ID_STD_46_LSB) | tan[[SUSv4]](#ID_STD_46_SUSV4) |
| cos[[SUSv4]](#ID_STD_46_SUSV4) | j0l[[LSB]](#ID_STD_46_LSB) | tanf[[SUSv4]](#ID_STD_46_SUSV4) |
| cosf[[SUSv4]](#ID_STD_46_SUSV4) | j1[[SUSv4]](#ID_STD_46_SUSV4) | tanh[[SUSv4]](#ID_STD_46_SUSV4) |
| cosh[[SUSv4]](#ID_STD_46_SUSV4) | j1f[[LSB]](#ID_STD_46_LSB) | tanhf[[SUSv4]](#ID_STD_46_SUSV4) |
| coshf[[SUSv4]](#ID_STD_46_SUSV4) | j1l[[LSB]](#ID_STD_46_LSB) | tanhl[[SUSv4]](#ID_STD_46_SUSV4) |
| coshl[[SUSv4]](#ID_STD_46_SUSV4) | jn[[SUSv4]](#ID_STD_46_SUSV4) | tanl[[SUSv4]](#ID_STD_46_SUSV4) |
| cosl[[SUSv4]](#ID_STD_46_SUSV4) | jnf[[LSB]](#ID_STD_46_LSB) | tgamma[[SUSv4]](#ID_STD_46_SUSV4) |
| cpow[[SUSv4]](#ID_STD_46_SUSV4) | jnl[[LSB]](#ID_STD_46_LSB) | tgammaf[[SUSv4]](#ID_STD_46_SUSV4) |
| cpowf[[SUSv4]](#ID_STD_46_SUSV4) | ldexp[[SUSv4]](#ID_STD_46_SUSV4) | tgammal[[SUSv4]](#ID_STD_46_SUSV4) |
| cpowl[[SUSv4]](#ID_STD_46_SUSV4) | ldexpf[[SUSv4]](#ID_STD_46_SUSV4) | trunc[[SUSv4]](#ID_STD_46_SUSV4) |
| cproj[[SUSv4]](#ID_STD_46_SUSV4) | ldexpl[[SUSv4]](#ID_STD_46_SUSV4) | truncf[[SUSv4]](#ID_STD_46_SUSV4) |
| cprojf[[SUSv4]](#ID_STD_46_SUSV4) | lgamma[[SUSv4]](#ID_STD_46_SUSV4) | truncl[[SUSv4]](#ID_STD_46_SUSV4) |
| cprojl[[SUSv4]](#ID_STD_46_SUSV4) | lgamma\_r[[LSB]](#ID_STD_46_LSB) | y0[[SUSv4]](#ID_STD_46_SUSV4) |
| creal[[SUSv4]](#ID_STD_46_SUSV4) | lgammaf[[SUSv4]](#ID_STD_46_SUSV4) | y0f[[LSB]](#ID_STD_46_LSB) |
| crealf[[SUSv4]](#ID_STD_46_SUSV4) | lgammaf\_r[[LSB]](#ID_STD_46_LSB) | y0l[[LSB]](#ID_STD_46_LSB) |
| creall[[SUSv4]](#ID_STD_46_SUSV4) | lgammal[[SUSv4]](#ID_STD_46_SUSV4) | y1[[SUSv4]](#ID_STD_46_SUSV4) |
| csin[[SUSv4]](#ID_STD_46_SUSV4) | lgammal\_r[[LSB]](#ID_STD_46_LSB) | y1f[[LSB]](#ID_STD_46_LSB) |
| csinf[[SUSv4]](#ID_STD_46_SUSV4) | llrint[[SUSv4]](#ID_STD_46_SUSV4) | y1l[[LSB]](#ID_STD_46_LSB) |
| csinh[[SUSv4]](#ID_STD_46_SUSV4) | llrintf[[SUSv4]](#ID_STD_46_SUSV4) | yn[[SUSv4]](#ID_STD_46_SUSV4) |
| csinhf[[SUSv4]](#ID_STD_46_SUSV4) | llrintl[[SUSv4]](#ID_STD_46_SUSV4) | ynf[[LSB]](#ID_STD_46_LSB) |
| csinhl[[SUSv4]](#ID_STD_46_SUSV4) | llround[[SUSv4]](#ID_STD_46_SUSV4) | ynl[[LSB]](#ID_STD_46_LSB) |

**Table A-7 libm Data Interfaces**

|  |  |  |
| --- | --- | --- |
| signgam[[SUSv4]](#ID_STD_46_SUSV4) |  |  |

## **A.6 libncurses**

The behavior of the interfaces in this library is specified by the following Standards.

|  |
| --- |
| [This Specification](#ID_STD_46_LSB) [LSB] |
| [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES) [X-CURSES] |

**Table A-8 libncurses Function Interfaces**

|  |  |  |
| --- | --- | --- |
| addch[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvdelch[[X-CURSES]](#ID_STD_46_X_46_CURSES) | slk\_refresh[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| addchnstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvderwin[[X-CURSES]](#ID_STD_46_X_46_CURSES) | slk\_restore[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| addchstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvgetch[[X-CURSES]](#ID_STD_46_X_46_CURSES) | slk\_set[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| addnstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvgetnstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) | slk\_touch[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| addstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvgetstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) | standend[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| attr\_get[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvhline[[X-CURSES]](#ID_STD_46_X_46_CURSES) | standout[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| attr\_off[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvinch[[X-CURSES]](#ID_STD_46_X_46_CURSES) | start\_color[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| attr\_on[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvinchnstr[[LSB]](#ID_STD_46_LSB) | subpad[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| attr\_set[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvinchstr[[LSB]](#ID_STD_46_LSB) | subwin[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| attroff[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvinnstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) | syncok[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| attron[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvinsch[[X-CURSES]](#ID_STD_46_X_46_CURSES) | termattrs[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| attrset[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvinsnstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) | termname[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| baudrate[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvinsstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) | tgetent[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| beep[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvinstr[[LSB]](#ID_STD_46_LSB) | tgetflag[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| bkgd[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvprintw[[X-CURSES]](#ID_STD_46_X_46_CURSES) | tgetnum[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| bkgdset[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvscanw[[LSB]](#ID_STD_46_LSB) | tgetstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| border[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvvline[[X-CURSES]](#ID_STD_46_X_46_CURSES) | tgoto[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| box[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvwaddch[[X-CURSES]](#ID_STD_46_X_46_CURSES) | tigetflag[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| can\_change\_color[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvwaddchnstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) | tigetnum[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| cbreak[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvwaddchstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) | tigetstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| chgat[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvwaddnstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) | timeout[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| clear[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvwaddstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) | touchline[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| clearok[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvwchgat[[X-CURSES]](#ID_STD_46_X_46_CURSES) | touchwin[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| clrtobot[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvwdelch[[X-CURSES]](#ID_STD_46_X_46_CURSES) | tparm[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| clrtoeol[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvwgetch[[X-CURSES]](#ID_STD_46_X_46_CURSES) | tputs[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| color\_content[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvwgetnstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) | typeahead[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| color\_set[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvwgetstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) | unctrl[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| copywin[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvwhline[[X-CURSES]](#ID_STD_46_X_46_CURSES) | ungetch[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| curs\_set[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvwin[[X-CURSES]](#ID_STD_46_X_46_CURSES) | untouchwin[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| def\_prog\_mode[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvwinch[[X-CURSES]](#ID_STD_46_X_46_CURSES) | use\_env[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| def\_shell\_mode[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvwinchnstr[[LSB]](#ID_STD_46_LSB) | vidattr[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| del\_curterm[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvwinchstr[[LSB]](#ID_STD_46_LSB) | vidputs[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| delay\_output[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvwinnstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) | vline[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| delch[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvwinsch[[X-CURSES]](#ID_STD_46_X_46_CURSES) | vw\_printw[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| deleteln[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvwinsnstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) | vw\_scanw[[LSB]](#ID_STD_46_LSB) |
| delscreen[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvwinsstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) | vwprintw[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| delwin[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvwinstr[[LSB]](#ID_STD_46_LSB) | vwscanw[[LSB]](#ID_STD_46_LSB) |
| derwin[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvwprintw[[X-CURSES]](#ID_STD_46_X_46_CURSES) | waddch[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| doupdate[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvwscanw[[LSB]](#ID_STD_46_LSB) | waddchnstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| dupwin[[X-CURSES]](#ID_STD_46_X_46_CURSES) | mvwvline[[X-CURSES]](#ID_STD_46_X_46_CURSES) | waddchstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| echo[[X-CURSES]](#ID_STD_46_X_46_CURSES) | napms[[X-CURSES]](#ID_STD_46_X_46_CURSES) | waddnstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| echochar[[X-CURSES]](#ID_STD_46_X_46_CURSES) | newpad[[X-CURSES]](#ID_STD_46_X_46_CURSES) | waddstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| endwin[[X-CURSES]](#ID_STD_46_X_46_CURSES) | newterm[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wattr\_get[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| erase[[X-CURSES]](#ID_STD_46_X_46_CURSES) | newwin[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wattr\_off[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| erasechar[[X-CURSES]](#ID_STD_46_X_46_CURSES) | nl[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wattr\_on[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| filter[[X-CURSES]](#ID_STD_46_X_46_CURSES) | nocbreak[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wattr\_set[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| flash[[X-CURSES]](#ID_STD_46_X_46_CURSES) | nodelay[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wattroff[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| flushinp[[X-CURSES]](#ID_STD_46_X_46_CURSES) | noecho[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wattron[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| getbkgd[[X-CURSES]](#ID_STD_46_X_46_CURSES) | nonl[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wattrset[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| getch[[X-CURSES]](#ID_STD_46_X_46_CURSES) | noqiflush[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wbkgd[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| getnstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) | noraw[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wbkgdset[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| getstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) | notimeout[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wborder[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| getwin[[X-CURSES]](#ID_STD_46_X_46_CURSES) | overlay[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wchgat[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| halfdelay[[X-CURSES]](#ID_STD_46_X_46_CURSES) | overwrite[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wclear[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| has\_colors[[X-CURSES]](#ID_STD_46_X_46_CURSES) | pair\_content[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wclrtobot[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| has\_ic[[X-CURSES]](#ID_STD_46_X_46_CURSES) | pechochar[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wclrtoeol[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| has\_il[[X-CURSES]](#ID_STD_46_X_46_CURSES) | pnoutrefresh[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wcolor\_set[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| hline[[X-CURSES]](#ID_STD_46_X_46_CURSES) | prefresh[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wcursyncup[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| idcok[[X-CURSES]](#ID_STD_46_X_46_CURSES) | printw[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wdelch[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| idlok[[X-CURSES]](#ID_STD_46_X_46_CURSES) | putp[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wdeleteln[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| immedok[[X-CURSES]](#ID_STD_46_X_46_CURSES) | putwin[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wechochar[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| inch[[X-CURSES]](#ID_STD_46_X_46_CURSES) | qiflush[[X-CURSES]](#ID_STD_46_X_46_CURSES) | werase[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| inchnstr[[LSB]](#ID_STD_46_LSB) | raw[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wgetch[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| inchstr[[LSB]](#ID_STD_46_LSB) | redrawwin[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wgetnstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| init\_color[[X-CURSES]](#ID_STD_46_X_46_CURSES) | refresh[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wgetstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| init\_pair[[X-CURSES]](#ID_STD_46_X_46_CURSES) | reset\_prog\_mode[[X-CURSES]](#ID_STD_46_X_46_CURSES) | whline[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| initscr[[X-CURSES]](#ID_STD_46_X_46_CURSES) | reset\_shell\_mode[[X-CURSES]](#ID_STD_46_X_46_CURSES) | winch[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| innstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) | resetty[[X-CURSES]](#ID_STD_46_X_46_CURSES) | winchnstr[[LSB]](#ID_STD_46_LSB) |
| insch[[X-CURSES]](#ID_STD_46_X_46_CURSES) | restartterm[[X-CURSES]](#ID_STD_46_X_46_CURSES) | winchstr[[LSB]](#ID_STD_46_LSB) |
| insdelln[[X-CURSES]](#ID_STD_46_X_46_CURSES) | ripoffline[[LSB]](#ID_STD_46_LSB) | winnstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| insertln[[X-CURSES]](#ID_STD_46_X_46_CURSES) | savetty[[X-CURSES]](#ID_STD_46_X_46_CURSES) | winsch[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| insnstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) | scanw[[LSB]](#ID_STD_46_LSB) | winsdelln[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| insstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) | scr\_dump[[X-CURSES]](#ID_STD_46_X_46_CURSES) | winsertln[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| instr[[LSB]](#ID_STD_46_LSB) | scr\_init[[X-CURSES]](#ID_STD_46_X_46_CURSES) | winsnstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| intrflush[[X-CURSES]](#ID_STD_46_X_46_CURSES) | scr\_restore[[X-CURSES]](#ID_STD_46_X_46_CURSES) | winsstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| is\_linetouched[[X-CURSES]](#ID_STD_46_X_46_CURSES) | scr\_set[[X-CURSES]](#ID_STD_46_X_46_CURSES) | winstr[[LSB]](#ID_STD_46_LSB) |
| is\_wintouched[[X-CURSES]](#ID_STD_46_X_46_CURSES) | scrl[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wmove[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| isendwin[[X-CURSES]](#ID_STD_46_X_46_CURSES) | scroll[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wnoutrefresh[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| keyname[[X-CURSES]](#ID_STD_46_X_46_CURSES) | scrollok[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wprintw[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| keypad[[X-CURSES]](#ID_STD_46_X_46_CURSES) | set\_curterm[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wredrawln[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| killchar[[X-CURSES]](#ID_STD_46_X_46_CURSES) | set\_term[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wrefresh[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| leaveok[[X-CURSES]](#ID_STD_46_X_46_CURSES) | setscrreg[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wscanw[[LSB]](#ID_STD_46_LSB) |
| longname[[X-CURSES]](#ID_STD_46_X_46_CURSES) | setupterm[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wscrl[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| meta[[X-CURSES]](#ID_STD_46_X_46_CURSES) | slk\_attr\_set[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wsetscrreg[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| move[[X-CURSES]](#ID_STD_46_X_46_CURSES) | slk\_attroff[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wstandend[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| mvaddch[[X-CURSES]](#ID_STD_46_X_46_CURSES) | slk\_attron[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wstandout[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| mvaddchnstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) | slk\_attrset[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wsyncdown[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| mvaddchstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) | slk\_clear[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wsyncup[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| mvaddnstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) | slk\_color[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wtimeout[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| mvaddstr[[X-CURSES]](#ID_STD_46_X_46_CURSES) | slk\_init[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wtouchln[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| mvchgat[[X-CURSES]](#ID_STD_46_X_46_CURSES) | slk\_label[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wvline[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| mvcur[[LSB]](#ID_STD_46_LSB) | slk\_noutrefresh[[X-CURSES]](#ID_STD_46_X_46_CURSES) |  |

**Table A-9 libncurses Data Interfaces**

|  |  |  |
| --- | --- | --- |
| COLORS[[X-CURSES]](#ID_STD_46_X_46_CURSES) | acs\_map[[X-CURSES]](#ID_STD_46_X_46_CURSES) | stdscr[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| COLOR\_PAIRS[[X-CURSES]](#ID_STD_46_X_46_CURSES) | cur\_term[[X-CURSES]](#ID_STD_46_X_46_CURSES) | ttytype[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| COLS[[X-CURSES]](#ID_STD_46_X_46_CURSES) | curscr[[X-CURSES]](#ID_STD_46_X_46_CURSES) |  |
| LINES[[X-CURSES]](#ID_STD_46_X_46_CURSES) | newscr[[LSB]](#ID_STD_46_LSB) |  |

## **A.7 libncursesw**

The behavior of the interfaces in this library is specified by the following Standards.

|  |
| --- |
| [Libncursesw API](#ID_STD_46_LIBNCURSESW) [Libncursesw] |
| [This Specification](#ID_STD_46_LSB) [LSB] |
| [Libncursesw Placeholder](#ID_STD_46_NCURSESW) [ncursesw] |
| [X/Open Curses, Issue 7](#ID_STD_46_X_46_CURSES) [X-CURSES] |

**Table A-10 libncursesw Function Interfaces**

|  |  |  |
| --- | --- | --- |
| add\_wch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvaddnwstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | slk\_init[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| add\_wchnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvaddstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | slk\_label[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| add\_wchstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvaddwstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | slk\_noutrefresh[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| addch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvchgat[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | slk\_refresh[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| addchnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvcur[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | slk\_restore[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| addchstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvdelch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | slk\_set[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| addnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvderwin[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | slk\_touch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| addnwstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvget\_wch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | slk\_wset[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| addstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvget\_wstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | standend[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| addwstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvgetch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | standout[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| assume\_default\_colors[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvgetn\_wstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | start\_color[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| attr\_get[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvgetnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | subpad[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| attr\_off[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvgetstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | subwin[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| attr\_on[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvhline[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | syncok[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| attr\_set[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvhline\_set[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | term\_attrs[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| attroff[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvin\_wch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | termattrs[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| attron[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvin\_wchnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | termname[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| attrset[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvin\_wchstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | tgetent[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| baudrate[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvinch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | tgetflag[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| beep[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvinchnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | tgetnum[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| bkgd[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvinchstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | tgetstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| bkgdset[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvinnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | tgoto[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| bkgrnd[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvinnwstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | tigetflag[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| bkgrndset[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvins\_nwstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | tigetnum[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| border[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvins\_wch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | tigetstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| border\_set[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvins\_wstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | timeout[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| box[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvinsch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | touchline[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| box\_set[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvinsnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | touchwin[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| can\_change\_color[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvinsstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | tparm[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| cbreak[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvinstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | tputs[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| chgat[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvinwstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | typeahead[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| clear[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvprintw[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | unctrl[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| clearok[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvscanw[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | unget\_wch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| clrtobot[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvvline[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | ungetch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| clrtoeol[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvvline\_set[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | ungetmouse[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| color\_content[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwadd\_wch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | untouchwin[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| color\_set[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwadd\_wchnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | use\_default\_colors[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| copywin[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwadd\_wchstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | use\_env[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| curs\_set[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwaddch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | use\_extended\_names[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| curses\_version[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwaddchnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | vid\_attr[[X-CURSES]](#ID_STD_46_X_46_CURSES) |
| def\_prog\_mode[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwaddchstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | vid\_puts[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| def\_shell\_mode[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwaddnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | vidattr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| define\_key[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwaddnwstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | vidputs[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| del\_curterm[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwaddstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | vline[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| delay\_output[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwaddwstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | vline\_set[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| delch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwchgat[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | vw\_printw[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| deleteln[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwdelch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | vw\_scanw[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| delscreen[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwget\_wch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | vwprintw[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| delwin[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwget\_wstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | vwscanw[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| derwin[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwgetch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wadd\_wch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| doupdate[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwgetn\_wstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wadd\_wchnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| dupwin[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwgetnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wadd\_wchstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| echo[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwgetstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | waddch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| echo\_wchar[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwhline[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | waddchnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| echochar[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwhline\_set[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | waddchstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| endwin[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwin[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | waddnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| erase[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwin\_wch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | waddnwstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| erasechar[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwin\_wchnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | waddstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| erasewchar[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwin\_wchstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | waddwstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| filter[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwinch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wattr\_get[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| flash[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwinchnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wattr\_off[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| flushinp[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwinchstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wattr\_on[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| get\_wch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwinnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wattr\_set[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| get\_wstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwinnwstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wattroff[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| getbkgd[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwins\_nwstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wattron[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| getbkgrnd[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwins\_wch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wattrset[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| getcchar[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwins\_wstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wbkgd[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| getch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwinsch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wbkgdset[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| getmouse[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwinsnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wbkgrnd[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| getn\_wstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwinsstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wbkgrndset[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| getnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwinstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wborder[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| getstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwinwstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wborder\_set[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| getwin[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwprintw[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wchgat[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| halfdelay[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwscanw[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wclear[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| has\_colors[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwvline[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wclrtobot[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| has\_ic[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | mvwvline\_set[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wclrtoeol[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| has\_il[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | napms[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wcolor\_set[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| has\_key[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | newpad[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wcursyncup[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| has\_mouse[[LSB]](#ID_STD_46_LSB) | newterm[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wdelch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| hline[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | newwin[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wdeleteln[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| hline\_set[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | nl[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wecho\_wchar[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| idcok[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | nocbreak[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wechochar[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| idlok[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | nodelay[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | werase[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| immedok[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | noecho[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wget\_wch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| in\_wch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | nonl[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wget\_wstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| in\_wchnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | noqiflush[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wgetbkgrnd[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| in\_wchstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | noraw[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wgetch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| inch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | notimeout[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wgetn\_wstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| inchnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | overlay[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wgetnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| inchstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | overwrite[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wgetstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| init\_color[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | pair\_content[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | whline[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| init\_pair[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | pecho\_wchar[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | whline\_set[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| initscr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | pechochar[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | win\_wch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| innstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | pnoutrefresh[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | win\_wchnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| innwstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | prefresh[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | win\_wchstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| ins\_nwstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | printw[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | winch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| ins\_wch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | putp[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | winchnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| ins\_wstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | putwin[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | winchstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| insch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | qiflush[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | winnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| insdelln[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | raw[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | winnwstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| insertln[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | redrawwin[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wins\_nwstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| insnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | refresh[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wins\_wch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| insstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | reset\_prog\_mode[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wins\_wstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| instr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | reset\_shell\_mode[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | winsch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| intrflush[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | resetty[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | winsdelln[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| inwstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | resizeterm[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | winsertln[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| is\_linetouched[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | restartterm[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | winsnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| is\_wintouched[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | ripoffline[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | winsstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| isendwin[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | savetty[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | winstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| key\_name[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | scanw[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | winwstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| keybound[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | scr\_dump[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wmouse\_trafo[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| keyname[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | scr\_init[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wmove[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| keyok[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | scr\_restore[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wnoutrefresh[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| keypad[[LSB]](#ID_STD_46_LSB) | scr\_set[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wprintw[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| killchar[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | scrl[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wredrawln[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| killwchar[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | scroll[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wrefresh[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| leaveok[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | scrollok[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wresize[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| longname[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | set\_curterm[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wscanw[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| mcprint[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | set\_term[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wscrl[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| meta[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | setcchar[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wsetscrreg[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| mouse\_trafo[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | setscrreg[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wstandend[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| mouseinterval[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | setupterm[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wstandout[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| mousemask[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | slk\_attr[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wsyncdown[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| move[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | slk\_attr\_off[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wsyncup[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| mvadd\_wch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | slk\_attr\_on[[X-CURSES]](#ID_STD_46_X_46_CURSES) | wtimeout[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| mvadd\_wchnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | slk\_attr\_set[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wtouchln[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| mvadd\_wchstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | slk\_attroff[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wunctrl[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| mvaddch[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | slk\_attron[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wvline[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| mvaddchnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | slk\_attrset[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | wvline\_set[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |
| mvaddchstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | slk\_clear[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |  |
| mvaddnstr[[Libncursesw]](#ID_STD_46_LIBNCURSESW) | slk\_color[[Libncursesw]](#ID_STD_46_LIBNCURSESW) |  |

**Table A-11 libncursesw Data Interfaces**

|  |  |  |
| --- | --- | --- |
| COLORS[[ncursesw]](#ID_STD_46_NCURSESW) | acs\_map[[LSB]](#ID_STD_46_LSB) | stdscr[[ncursesw]](#ID_STD_46_NCURSESW) |
| COLOR\_PAIRS[[ncursesw]](#ID_STD_46_NCURSESW) | cur\_term[[LSB]](#ID_STD_46_LSB) | ttytype[[ncursesw]](#ID_STD_46_NCURSESW) |
| COLS[[ncursesw]](#ID_STD_46_NCURSESW) | curscr[[ncursesw]](#ID_STD_46_NCURSESW) |  |
| LINES[[ncursesw]](#ID_STD_46_NCURSESW) | newscr[[ncursesw]](#ID_STD_46_NCURSESW) |  |

## **A.8 libpam**

The behavior of the interfaces in this library is specified by the following Standards.

|  |
| --- |
| [This Specification](#ID_STD_46_LSB) [LSB] |
| [PAM](#ID_STD_46_PAM) [PAM] |

**Table A-12 libpam Function Interfaces**

|  |  |  |
| --- | --- | --- |
| pam\_acct\_mgmt(LIBPAM\_1.0)[[LSB]](#ID_STD_46_LSB) | pam\_get\_data(LIBPAM\_1.0)[[PAM]](#ID_STD_46_PAM) | pam\_putenv(LIBPAM\_1.0)[[LSB]](#ID_STD_46_LSB) |
| pam\_authenticate(LIBPAM\_1.0)[[LSB]](#ID_STD_46_LSB) | pam\_get\_item(LIBPAM\_1.0)[[LSB]](#ID_STD_46_LSB) | pam\_set\_data(LIBPAM\_1.0)[[PAM]](#ID_STD_46_PAM) |
| pam\_chauthtok(LIBPAM\_1.0)[[LSB]](#ID_STD_46_LSB) | pam\_get\_user(LIBPAM\_1.0)[[PAM]](#ID_STD_46_PAM) | pam\_set\_item(LIBPAM\_1.0)[[LSB]](#ID_STD_46_LSB) |
| pam\_close\_session(LIBPAM\_1.0)[[LSB]](#ID_STD_46_LSB) | pam\_getenv(LIBPAM\_1.0)[[LSB]](#ID_STD_46_LSB) | pam\_setcred(LIBPAM\_1.0)[[LSB]](#ID_STD_46_LSB) |
| pam\_end(LIBPAM\_1.0)[[LSB]](#ID_STD_46_LSB) | pam\_getenvlist(LIBPAM\_1.0)[[LSB]](#ID_STD_46_LSB) | pam\_start(LIBPAM\_1.0)[[LSB]](#ID_STD_46_LSB) |
| pam\_fail\_delay(LIBPAM\_1.0)[[LSB]](#ID_STD_46_LSB) | pam\_open\_session(LIBPAM\_1.0)[[LSB]](#ID_STD_46_LSB) | pam\_strerror(LIBPAM\_1.0)[[LSB]](#ID_STD_46_LSB) |

## **A.9 libpthread**

The behavior of the interfaces in this library is specified by the following Standards.

|  |
| --- |
| [Large File Support](#ID_STD_46_LFS) [LFS] |
| [This Specification](#ID_STD_46_LSB) [LSB] |
| [POSIX 1003.1-2001 (ISO/IEC 9945-2003)](#ID_STD_46_SUSV3) [SUSv3] |
| [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) [SUSv4] |

**Table A-13 libpthread Function Interfaces**

|  |  |  |
| --- | --- | --- |
| \_\_errno\_location[[LSB]](#ID_STD_46_LSB) | pthread\_cond\_destroy[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_rwlock\_tryrdlock[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_h\_errno\_location[[LSB]](#ID_STD_46_LSB) | pthread\_cond\_init[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_rwlock\_trywrlock[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_libc\_current\_sigrtmax[[LSB]](#ID_STD_46_LSB) | pthread\_cond\_signal[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_rwlock\_unlock[[SUSv4]](#ID_STD_46_SUSV4) |
| \_\_libc\_current\_sigrtmin[[LSB]](#ID_STD_46_LSB) | pthread\_cond\_timedwait[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_rwlock\_wrlock[[SUSv4]](#ID_STD_46_SUSV4) |
| \_pthread\_cleanup\_pop[[LSB]](#ID_STD_46_LSB) | pthread\_cond\_wait[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_rwlockattr\_destroy[[SUSv4]](#ID_STD_46_SUSV4) |
| \_pthread\_cleanup\_push[[LSB]](#ID_STD_46_LSB) | pthread\_condattr\_destroy[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_rwlockattr\_getkind\_np[[LSB]](#ID_STD_46_LSB) |
| accept[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_condattr\_getclock(GLIBC\_2.3.3)[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_rwlockattr\_getpshared[[SUSv4]](#ID_STD_46_SUSV4) |
| close[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_condattr\_getpshared[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_rwlockattr\_init[[SUSv4]](#ID_STD_46_SUSV4) |
| connect[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_condattr\_init[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_rwlockattr\_setkind\_np[[LSB]](#ID_STD_46_LSB) |
| fcntl[[LSB]](#ID_STD_46_LSB) | pthread\_condattr\_setclock(GLIBC\_2.3.3)[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_rwlockattr\_setpshared[[SUSv4]](#ID_STD_46_SUSV4) |
| flockfile[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_condattr\_setpshared[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_self[[SUSv4]](#ID_STD_46_SUSV4) |
| fork[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_create[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_setcancelstate[[SUSv4]](#ID_STD_46_SUSV4) |
| fsync[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_detach[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_setcanceltype[[SUSv4]](#ID_STD_46_SUSV4) |
| ftrylockfile[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_equal[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_setconcurrency[[SUSv4]](#ID_STD_46_SUSV4) |
| funlockfile[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_exit[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_setschedparam[[SUSv4]](#ID_STD_46_SUSV4) |
| longjmp[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_getattr\_np[[LSB]](#ID_STD_46_LSB) | pthread\_setschedprio(GLIBC\_2.3.4)[[SUSv4]](#ID_STD_46_SUSV4) |
| lseek[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_getconcurrency[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_setspecific[[SUSv4]](#ID_STD_46_SUSV4) |
| lseek64[[LFS]](#ID_STD_46_LFS) | pthread\_getcpuclockid[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_sigmask[[SUSv4]](#ID_STD_46_SUSV4) |
| msync[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_getschedparam[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_spin\_destroy[[SUSv4]](#ID_STD_46_SUSV4) |
| nanosleep[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_getspecific[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_spin\_init[[SUSv4]](#ID_STD_46_SUSV4) |
| open[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_join[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_spin\_lock[[SUSv4]](#ID_STD_46_SUSV4) |
| open64[[LFS]](#ID_STD_46_LFS) | pthread\_key\_create[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_spin\_trylock[[SUSv4]](#ID_STD_46_SUSV4) |
| pause[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_key\_delete[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_spin\_unlock[[SUSv4]](#ID_STD_46_SUSV4) |
| pread[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_kill[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_testcancel[[SUSv4]](#ID_STD_46_SUSV4) |
| pread64[[LSB]](#ID_STD_46_LSB) | pthread\_mutex\_consistent(GLIBC\_2.12)[[SUSv4]](#ID_STD_46_SUSV4) | pwrite[[SUSv4]](#ID_STD_46_SUSV4) |
| pthread\_attr\_destroy[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_mutex\_consistent\_np[[LSB]](#ID_STD_46_LSB) | pwrite64[[LSB]](#ID_STD_46_LSB) |
| pthread\_attr\_getdetachstate[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_mutex\_destroy[[SUSv4]](#ID_STD_46_SUSV4) | raise[[SUSv4]](#ID_STD_46_SUSV4) |
| pthread\_attr\_getguardsize[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_mutex\_getprioceiling(GLIBC\_2.4)[[SUSv4]](#ID_STD_46_SUSV4) | read[[SUSv4]](#ID_STD_46_SUSV4) |
| pthread\_attr\_getinheritsched[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_mutex\_init[[SUSv4]](#ID_STD_46_SUSV4) | recv[[SUSv4]](#ID_STD_46_SUSV4) |
| pthread\_attr\_getschedparam[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_mutex\_lock[[SUSv4]](#ID_STD_46_SUSV4) | recvfrom[[SUSv4]](#ID_STD_46_SUSV4) |
| pthread\_attr\_getschedpolicy[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_mutex\_setprioceiling(GLIBC\_2.4)[[SUSv4]](#ID_STD_46_SUSV4) | recvmsg[[SUSv4]](#ID_STD_46_SUSV4) |
| pthread\_attr\_getscope[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_mutex\_timedlock[[SUSv4]](#ID_STD_46_SUSV4) | sem\_close[[SUSv4]](#ID_STD_46_SUSV4) |
| pthread\_attr\_getstack[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_mutex\_trylock[[SUSv4]](#ID_STD_46_SUSV4) | sem\_destroy[[SUSv4]](#ID_STD_46_SUSV4) |
| pthread\_attr\_getstackaddr[[SUSv3]](#ID_STD_46_SUSV3) | pthread\_mutex\_unlock[[SUSv4]](#ID_STD_46_SUSV4) | sem\_getvalue[[SUSv4]](#ID_STD_46_SUSV4) |
| pthread\_attr\_getstacksize[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_mutexattr\_destroy[[SUSv4]](#ID_STD_46_SUSV4) | sem\_init[[SUSv4]](#ID_STD_46_SUSV4) |
| pthread\_attr\_init[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_mutexattr\_getprioceiling(GLIBC\_2.4)[[SUSv4]](#ID_STD_46_SUSV4) | sem\_open[[SUSv4]](#ID_STD_46_SUSV4) |
| pthread\_attr\_setdetachstate[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_mutexattr\_getprotocol(GLIBC\_2.4)[[SUSv4]](#ID_STD_46_SUSV4) | sem\_post[[SUSv4]](#ID_STD_46_SUSV4) |
| pthread\_attr\_setguardsize[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_mutexattr\_getpshared[[SUSv4]](#ID_STD_46_SUSV4) | sem\_timedwait[[SUSv4]](#ID_STD_46_SUSV4) |
| pthread\_attr\_setinheritsched[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_mutexattr\_getrobust(GLIBC\_2.12)[[SUSv4]](#ID_STD_46_SUSV4) | sem\_trywait[[SUSv4]](#ID_STD_46_SUSV4) |
| pthread\_attr\_setschedparam[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_mutexattr\_getrobust\_np[[LSB]](#ID_STD_46_LSB) | sem\_unlink[[SUSv4]](#ID_STD_46_SUSV4) |
| pthread\_attr\_setschedpolicy[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_mutexattr\_gettype[[SUSv4]](#ID_STD_46_SUSV4) | sem\_wait[[SUSv4]](#ID_STD_46_SUSV4) |
| pthread\_attr\_setscope[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_mutexattr\_init[[SUSv4]](#ID_STD_46_SUSV4) | send[[SUSv4]](#ID_STD_46_SUSV4) |
| pthread\_attr\_setstack[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_mutexattr\_setprioceiling(GLIBC\_2.4)[[SUSv4]](#ID_STD_46_SUSV4) | sendmsg[[SUSv4]](#ID_STD_46_SUSV4) |
| pthread\_attr\_setstackaddr[[SUSv3]](#ID_STD_46_SUSV3) | pthread\_mutexattr\_setprotocol(GLIBC\_2.4)[[SUSv4]](#ID_STD_46_SUSV4) | sendto[[SUSv4]](#ID_STD_46_SUSV4) |
| pthread\_attr\_setstacksize[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_mutexattr\_setpshared[[SUSv4]](#ID_STD_46_SUSV4) | sigaction[[SUSv4]](#ID_STD_46_SUSV4) |
| pthread\_barrier\_destroy[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_mutexattr\_setrobust(GLIBC\_2.12)[[SUSv4]](#ID_STD_46_SUSV4) | siglongjmp[[SUSv4]](#ID_STD_46_SUSV4) |
| pthread\_barrier\_init[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_mutexattr\_setrobust\_np[[LSB]](#ID_STD_46_LSB) | sigwait[[SUSv4]](#ID_STD_46_SUSV4) |
| pthread\_barrier\_wait[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_mutexattr\_settype[[SUSv4]](#ID_STD_46_SUSV4) | system[[LSB]](#ID_STD_46_LSB) |
| pthread\_barrierattr\_destroy[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_once[[SUSv4]](#ID_STD_46_SUSV4) | tcdrain[[SUSv4]](#ID_STD_46_SUSV4) |
| pthread\_barrierattr\_getpshared(GLIBC\_2.3.3)[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_rwlock\_destroy[[SUSv4]](#ID_STD_46_SUSV4) | vfork[[SUSv3]](#ID_STD_46_SUSV3) |
| pthread\_barrierattr\_init[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_rwlock\_init[[SUSv4]](#ID_STD_46_SUSV4) | wait[[SUSv4]](#ID_STD_46_SUSV4) |
| pthread\_barrierattr\_setpshared[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_rwlock\_rdlock[[SUSv4]](#ID_STD_46_SUSV4) | waitpid[[LSB]](#ID_STD_46_LSB) |
| pthread\_cancel[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_rwlock\_timedrdlock[[SUSv4]](#ID_STD_46_SUSV4) | write[[SUSv4]](#ID_STD_46_SUSV4) |
| pthread\_cond\_broadcast[[SUSv4]](#ID_STD_46_SUSV4) | pthread\_rwlock\_timedwrlock[[SUSv4]](#ID_STD_46_SUSV4) |  |

## **A.10 librt**

The behavior of the interfaces in this library is specified by the following Standards.

|  |
| --- |
| [Large File Support](#ID_STD_46_LFS) [LFS] |
| [POSIX 1003.1-2008 (ISO/IEC 9945-2009)](#ID_STD_46_SUSV4) [SUSv4] |

**Table A-14 librt Function Interfaces**

|  |  |  |
| --- | --- | --- |
| aio\_cancel[[SUSv4]](#ID_STD_46_SUSV4) | aio\_write64[[LFS]](#ID_STD_46_LFS) | mq\_send(GLIBC\_2.3.4)[[SUSv4]](#ID_STD_46_SUSV4) |
| aio\_cancel64[[LFS]](#ID_STD_46_LFS) | clock\_getcpuclockid[[SUSv4]](#ID_STD_46_SUSV4) | mq\_setattr(GLIBC\_2.3.4)[[SUSv4]](#ID_STD_46_SUSV4) |
| aio\_error[[SUSv4]](#ID_STD_46_SUSV4) | clock\_getres[[SUSv4]](#ID_STD_46_SUSV4) | mq\_timedreceive(GLIBC\_2.3.4)[[SUSv4]](#ID_STD_46_SUSV4) |
| aio\_error64[[LFS]](#ID_STD_46_LFS) | clock\_gettime[[SUSv4]](#ID_STD_46_SUSV4) | mq\_timedsend(GLIBC\_2.3.4)[[SUSv4]](#ID_STD_46_SUSV4) |
| aio\_fsync[[SUSv4]](#ID_STD_46_SUSV4) | clock\_nanosleep[[SUSv4]](#ID_STD_46_SUSV4) | mq\_unlink(GLIBC\_2.3.4)[[SUSv4]](#ID_STD_46_SUSV4) |
| aio\_fsync64[[LFS]](#ID_STD_46_LFS) | clock\_settime[[SUSv4]](#ID_STD_46_SUSV4) | shm\_open[[SUSv4]](#ID_STD_46_SUSV4) |
| aio\_read[[SUSv4]](#ID_STD_46_SUSV4) | lio\_listio(GLIBC\_2.4)[[SUSv4]](#ID_STD_46_SUSV4) | shm\_unlink[[SUSv4]](#ID_STD_46_SUSV4) |
| aio\_read64[[LFS]](#ID_STD_46_LFS) | lio\_listio64(GLIBC\_2.4)[[LFS]](#ID_STD_46_LFS) | timer\_create[[SUSv4]](#ID_STD_46_SUSV4) |
| aio\_return[[SUSv4]](#ID_STD_46_SUSV4) | mq\_close(GLIBC\_2.3.4)[[SUSv4]](#ID_STD_46_SUSV4) | timer\_delete[[SUSv4]](#ID_STD_46_SUSV4) |
| aio\_return64[[LFS]](#ID_STD_46_LFS) | mq\_getattr(GLIBC\_2.3.4)[[SUSv4]](#ID_STD_46_SUSV4) | timer\_getoverrun[[SUSv4]](#ID_STD_46_SUSV4) |
| aio\_suspend[[SUSv4]](#ID_STD_46_SUSV4) | mq\_notify(GLIBC\_2.3.4)[[SUSv4]](#ID_STD_46_SUSV4) | timer\_gettime[[SUSv4]](#ID_STD_46_SUSV4) |
| aio\_suspend64[[LFS]](#ID_STD_46_LFS) | mq\_open(GLIBC\_2.3.4)[[SUSv4]](#ID_STD_46_SUSV4) | timer\_settime[[SUSv4]](#ID_STD_46_SUSV4) |
| aio\_write[[SUSv4]](#ID_STD_46_SUSV4) | mq\_receive(GLIBC\_2.3.4)[[SUSv4]](#ID_STD_46_SUSV4) |  |

## **A.11 libutil**

The behavior of the interfaces in this library is specified by the following Standards.

|  |
| --- |
| [This Specification](#ID_STD_46_LSB) [LSB] |

**Table A-15 libutil Function Interfaces**

|  |  |  |
| --- | --- | --- |
| forkpty[[LSB]](#ID_STD_46_LSB) | login\_tty[[LSB]](#ID_STD_46_LSB) | logwtmp[[LSB]](#ID_STD_46_LSB) |
| login[[LSB]](#ID_STD_46_LSB) | logout[[LSB]](#ID_STD_46_LSB) | openpty[[LSB]](#ID_STD_46_LSB) |

## **A.12 libz**

The behavior of the interfaces in this library is specified by the following Standards.

|  |
| --- |
| [This Specification](#ID_STD_46_LSB) [LSB] |

**Table A-16 libz Function Interfaces**

|  |  |  |
| --- | --- | --- |
| adler32[[LSB]](#ID_STD_46_LSB) | gzclose[[LSB]](#ID_STD_46_LSB) | inflate[[LSB]](#ID_STD_46_LSB) |
| compress[[LSB]](#ID_STD_46_LSB) | gzdopen[[LSB]](#ID_STD_46_LSB) | inflateBack(ZLIB\_1.2.0)[[LSB]](#ID_STD_46_LSB) |
| compress2[[LSB]](#ID_STD_46_LSB) | gzeof[[LSB]](#ID_STD_46_LSB) | inflateBackEnd(ZLIB\_1.2.0)[[LSB]](#ID_STD_46_LSB) |
| compressBound(ZLIB\_1.2.0)[[LSB]](#ID_STD_46_LSB) | gzerror[[LSB]](#ID_STD_46_LSB) | inflateBackInit\_(ZLIB\_1.2.0)[[LSB]](#ID_STD_46_LSB) |
| crc32[[LSB]](#ID_STD_46_LSB) | gzflush[[LSB]](#ID_STD_46_LSB) | inflateCopy(ZLIB\_1.2.0)[[LSB]](#ID_STD_46_LSB) |
| deflate[[LSB]](#ID_STD_46_LSB) | gzgetc[[LSB]](#ID_STD_46_LSB) | inflateEnd[[LSB]](#ID_STD_46_LSB) |
| deflateBound(ZLIB\_1.2.0)[[LSB]](#ID_STD_46_LSB) | gzgets[[LSB]](#ID_STD_46_LSB) | inflateInit2\_[[LSB]](#ID_STD_46_LSB) |
| deflateCopy[[LSB]](#ID_STD_46_LSB) | gzopen[[LSB]](#ID_STD_46_LSB) | inflateInit\_[[LSB]](#ID_STD_46_LSB) |
| deflateEnd[[LSB]](#ID_STD_46_LSB) | gzprintf[[LSB]](#ID_STD_46_LSB) | inflateReset[[LSB]](#ID_STD_46_LSB) |
| deflateInit2\_[[LSB]](#ID_STD_46_LSB) | gzputc[[LSB]](#ID_STD_46_LSB) | inflateSetDictionary[[LSB]](#ID_STD_46_LSB) |
| deflateInit\_[[LSB]](#ID_STD_46_LSB) | gzputs[[LSB]](#ID_STD_46_LSB) | inflateSync[[LSB]](#ID_STD_46_LSB) |
| deflateParams[[LSB]](#ID_STD_46_LSB) | gzread[[LSB]](#ID_STD_46_LSB) | inflateSyncPoint[[LSB]](#ID_STD_46_LSB) |
| deflatePrime(ZLIB\_1.2.0.8)[[LSB]](#ID_STD_46_LSB) | gzrewind[[LSB]](#ID_STD_46_LSB) | uncompress[[LSB]](#ID_STD_46_LSB) |
| deflateReset[[LSB]](#ID_STD_46_LSB) | gzseek[[LSB]](#ID_STD_46_LSB) | zError[[LSB]](#ID_STD_46_LSB) |
| deflateSetDictionary[[LSB]](#ID_STD_46_LSB) | gzsetparams[[LSB]](#ID_STD_46_LSB) | zlibVersion[[LSB]](#ID_STD_46_LSB) |
| get\_crc\_table[[LSB]](#ID_STD_46_LSB) | gztell[[LSB]](#ID_STD_46_LSB) |  |
| gzclearerr(ZLIB\_1.2.0.2)[[LSB]](#ID_STD_46_LSB) | gzwrite[[LSB]](#ID_STD_46_LSB) |  |

## **A.13 libnspr4**

The behavior of the interfaces in this library is specified by the following Standards.

|  |
| --- |
| [NSPR Reference](#ID_STD_46_NSPR) [NSPR] |

**Table A-17 libnspr4 Function Interfaces**

|  |  |  |
| --- | --- | --- |
| PR\_Abort[[NSPR]](#ID_STD_46_NSPR) | PR\_GetErrorTextLength[[NSPR]](#ID_STD_46_NSPR) | PR\_NotifyAllCondVar[[NSPR]](#ID_STD_46_NSPR) |
| PR\_Accept[[NSPR]](#ID_STD_46_NSPR) | PR\_GetLayersIdentity[[NSPR]](#ID_STD_46_NSPR) | PR\_NotifyCondVar[[NSPR]](#ID_STD_46_NSPR) |
| PR\_AtomicAdd[[NSPR]](#ID_STD_46_NSPR) | PR\_GetOSError[[NSPR]](#ID_STD_46_NSPR) | PR\_Now[[NSPR]](#ID_STD_46_NSPR) |
| PR\_AtomicDecrement[[NSPR]](#ID_STD_46_NSPR) | PR\_GetSocketOption[[NSPR]](#ID_STD_46_NSPR) | PR\_Open[[NSPR]](#ID_STD_46_NSPR) |
| PR\_AtomicIncrement[[NSPR]](#ID_STD_46_NSPR) | PR\_GetThreadPrivate[[NSPR]](#ID_STD_46_NSPR) | PR\_OpenTCPSocket[[NSPR]](#ID_STD_46_NSPR) |
| PR\_AtomicSet[[NSPR]](#ID_STD_46_NSPR) | PR\_GetThreadScope[[NSPR]](#ID_STD_46_NSPR) | PR\_OpenUDPSocket[[NSPR]](#ID_STD_46_NSPR) |
| PR\_Bind[[NSPR]](#ID_STD_46_NSPR) | PR\_GetThreadState[[NSPR]](#ID_STD_46_NSPR) | PR\_ParseTimeString[[NSPR]](#ID_STD_46_NSPR) |
| PR\_Calloc[[NSPR]](#ID_STD_46_NSPR) | PR\_GetUniqueIdentity[[NSPR]](#ID_STD_46_NSPR) | PR\_ParseTimeStringToExplodedTime[[NSPR]](#ID_STD_46_NSPR) |
| PR\_Cleanup[[NSPR]](#ID_STD_46_NSPR) | PR\_ImplodeTime[[NSPR]](#ID_STD_46_NSPR) | PR\_Poll[[NSPR]](#ID_STD_46_NSPR) |
| PR\_ClearInterrupt[[NSPR]](#ID_STD_46_NSPR) | PR\_ImportTCPSocket[[NSPR]](#ID_STD_46_NSPR) | PR\_PopIOLayer[[NSPR]](#ID_STD_46_NSPR) |
| PR\_Close[[NSPR]](#ID_STD_46_NSPR) | PR\_Init[[NSPR]](#ID_STD_46_NSPR) | PR\_ProcessExit[[NSPR]](#ID_STD_46_NSPR) |
| PR\_Connect[[NSPR]](#ID_STD_46_NSPR) | PR\_Initialize[[NSPR]](#ID_STD_46_NSPR) | PR\_PushIOLayer[[NSPR]](#ID_STD_46_NSPR) |
| PR\_ConnectContinue[[NSPR]](#ID_STD_46_NSPR) | PR\_InitializeNetAddr[[NSPR]](#ID_STD_46_NSPR) | PR\_Read[[NSPR]](#ID_STD_46_NSPR) |
| PR\_ConvertIPv4AddrToIPv6[[NSPR]](#ID_STD_46_NSPR) | PR\_Initialized[[NSPR]](#ID_STD_46_NSPR) | PR\_Realloc[[NSPR]](#ID_STD_46_NSPR) |
| PR\_CreateIOLayerStub[[NSPR]](#ID_STD_46_NSPR) | PR\_Interrupt[[NSPR]](#ID_STD_46_NSPR) | PR\_Recv[[NSPR]](#ID_STD_46_NSPR) |
| PR\_CreatePipe[[NSPR]](#ID_STD_46_NSPR) | PR\_IntervalNow[[NSPR]](#ID_STD_46_NSPR) | PR\_RecvFrom[[NSPR]](#ID_STD_46_NSPR) |
| PR\_DestroyCondVar[[NSPR]](#ID_STD_46_NSPR) | PR\_IntervalToMicroseconds[[NSPR]](#ID_STD_46_NSPR) | PR\_SecondsToInterval[[NSPR]](#ID_STD_46_NSPR) |
| PR\_DestroyLock[[NSPR]](#ID_STD_46_NSPR) | PR\_IntervalToMilliseconds[[NSPR]](#ID_STD_46_NSPR) | PR\_Send[[NSPR]](#ID_STD_46_NSPR) |
| PR\_EnumerateAddrInfo[[NSPR]](#ID_STD_46_NSPR) | PR\_IntervalToSeconds[[NSPR]](#ID_STD_46_NSPR) | PR\_SendTo[[NSPR]](#ID_STD_46_NSPR) |
| PR\_ExplodeTime[[NSPR]](#ID_STD_46_NSPR) | PR\_Listen[[NSPR]](#ID_STD_46_NSPR) | PR\_SetError[[NSPR]](#ID_STD_46_NSPR) |
| PR\_FormatTime[[NSPR]](#ID_STD_46_NSPR) | PR\_LocalTimeParameters[[NSPR]](#ID_STD_46_NSPR) | PR\_SetErrorText[[NSPR]](#ID_STD_46_NSPR) |
| PR\_Free[[NSPR]](#ID_STD_46_NSPR) | PR\_Lock[[NSPR]](#ID_STD_46_NSPR) | PR\_SetSocketOption[[NSPR]](#ID_STD_46_NSPR) |
| PR\_FreeAddrInfo[[NSPR]](#ID_STD_46_NSPR) | PR\_Malloc[[NSPR]](#ID_STD_46_NSPR) | PR\_SetThreadPrivate[[NSPR]](#ID_STD_46_NSPR) |
| PR\_GMTParameters[[NSPR]](#ID_STD_46_NSPR) | PR\_MicrosecondsToInterval[[NSPR]](#ID_STD_46_NSPR) | PR\_Shutdown[[NSPR]](#ID_STD_46_NSPR) |
| PR\_GetAddrInfoByName[[NSPR]](#ID_STD_46_NSPR) | PR\_MillisecondsToInterval[[NSPR]](#ID_STD_46_NSPR) | PR\_Sleep[[NSPR]](#ID_STD_46_NSPR) |
| PR\_GetCanonNameFromAddrInfo[[NSPR]](#ID_STD_46_NSPR) | PR\_NetAddrToString[[NSPR]](#ID_STD_46_NSPR) | PR\_StringToNetAddr[[NSPR]](#ID_STD_46_NSPR) |
| PR\_GetDefaultIOMethods[[NSPR]](#ID_STD_46_NSPR) | PR\_NewCondVar[[NSPR]](#ID_STD_46_NSPR) | PR\_TicksPerSecond[[NSPR]](#ID_STD_46_NSPR) |
| PR\_GetDescType[[NSPR]](#ID_STD_46_NSPR) | PR\_NewLock[[NSPR]](#ID_STD_46_NSPR) | PR\_Unlock[[NSPR]](#ID_STD_46_NSPR) |
| PR\_GetError[[NSPR]](#ID_STD_46_NSPR) | PR\_NewThreadPrivateIndex[[NSPR]](#ID_STD_46_NSPR) | PR\_WaitCondVar[[NSPR]](#ID_STD_46_NSPR) |
| PR\_GetErrorText[[NSPR]](#ID_STD_46_NSPR) | PR\_NormalizeTime[[NSPR]](#ID_STD_46_NSPR) | PR\_Write[[NSPR]](#ID_STD_46_NSPR) |

## **A.14 libnss3**

The behavior of the interfaces in this library is specified by the following Standards.

|  |
| --- |
| [Mozilla's NSS SSL Reference](#ID_STD_46_NSS_46_SSL) [NSS SSL] |

**Table A-18 libnss3 Function Interfaces**

|  |  |  |
| --- | --- | --- |
| CERT\_CheckCertValidTimes(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | CERT\_VerifyCertNow(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | PK11\_GetSlotName(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) |
| CERT\_DestroyCertificate(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | NSS\_Init(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | PK11\_GetTokenName(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) |
| CERT\_DupCertificate(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | NSS\_InitReadWrite(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | PK11\_IsHW(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) |
| CERT\_FreeNicknames(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | NSS\_NoDB\_Init(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | PK11\_IsPresent(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) |
| CERT\_GetCertNicknames(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | NSS\_Shutdown(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | PK11\_IsReadOnly(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) |
| CERT\_GetDefaultCertDB(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | PK11\_FindCertFromNickname(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | PK11\_SetPasswordFunc(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) |
| CERT\_VerifyCertName(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | PK11\_FindKeyByAnyCert(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | SECKEY\_DestroyPrivateKey(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) |

## **A.15 libssl3**

The behavior of the interfaces in this library is specified by the following Standards.

|  |
| --- |
| [Mozilla's NSS SSL Reference](#ID_STD_46_NSS_46_SSL) [NSS SSL] |

**Table A-19 libssl3 Function Interfaces**

|  |  |  |
| --- | --- | --- |
| NSS\_CmpCertChainWCANames(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | SSL\_ConfigMPServerSIDCache(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | SSL\_OptionSet(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) |
| NSS\_FindCertKEAType(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | SSL\_ConfigSecureServer(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | SSL\_OptionSetDefault(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) |
| NSS\_GetClientAuthData(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | SSL\_ConfigServerSessionIDCache(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | SSL\_PeerCertificate(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) |
| SSL\_AuthCertificate(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | SSL\_DataPending(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | SSL\_ReHandshake(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) |
| SSL\_AuthCertificateHook(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | SSL\_ForceHandshake(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | SSL\_ResetHandshake(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) |
| SSL\_BadCertHook(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | SSL\_GetClientAuthDataHook(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | SSL\_RevealPinArg(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) |
| SSL\_CipherPolicyGet(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | SSL\_GetSessionID(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | SSL\_RevealURL(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) |
| SSL\_CipherPolicySet(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | SSL\_HandshakeCallback(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | SSL\_SecurityStatus(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) |
| SSL\_CipherPrefGet(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | SSL\_ImportFD(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | SSL\_SetPKCS11PinArg(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) |
| SSL\_CipherPrefGetDefault(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | SSL\_InheritMPServerSIDCache(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | SSL\_SetSockPeerID(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) |
| SSL\_CipherPrefSet(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | SSL\_InvalidateSession(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | SSL\_SetURL(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) |
| SSL\_CipherPrefSetDefault(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | SSL\_OptionGet(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) |  |
| SSL\_ClearSessionCache(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) | SSL\_OptionGetDefault(NSS\_3.2)[[NSS SSL]](#ID_STD_46_NSS_46_SSL) |  |

# **Annex B GNU Free Documentation License (Informative)**

This specification is published under the terms of the GNU Free Documentation License, Version 1.1, March 2000

Copyright (C) 2000 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## **B.1 PREAMBLE**

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

## **B.2 APPLICABILITY AND DEFINITIONS**

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

## **B.3 VERBATIM COPYING**

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## **B.4 COPYING IN QUANTITY**

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

## **B.5 MODIFICATIONS**

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## **B.6 COMBINING DOCUMENTS**

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

## **B.7 COLLECTIONS OF DOCUMENTS**

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## **B.8 AGGREGATION WITH INDEPENDENT WORKS**

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

## **B.9 TRANSLATION**

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

## **B.10 TERMINATION**

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## **B.11 FUTURE REVISIONS OF THIS LICENSE**

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

## **B.12 How to use this License for your documents**

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.