

INTERNATIONAL
STANDARD

ISO/IEC
23360-1-2

First edition
2021-10

Linux Standard Base (LSB) —
Part 1-2:
Core specification generic part



Reference number
ISO/IEC 23360-1-2:2021(E)

This is a preview. [Click here to purchase the full publication.](#)

© ISO/IEC 2021



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2021

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

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. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by the Linux Foundation as Linux Standard Base (LSB): Core specification generic part and drafted in accordance with its editorial rules. It was assigned to Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages, their environments and system software interfaces*, and adopted by National Bodies.

This first edition of ISO/IEC 23360-1-2 cancels and replaces ISO/IEC 23360-1:2006, which has been technically revised.

This document is based on “The GNU Free Documentation License, version 1.1”. The license is available at <https://www.gnu.org/licenses/old-licenses/fdl-1.1.html>.

A list of all parts in the ISO/IEC 23660 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Contents

Foreword	iii
Introduction	vii
I Introductory Elements	1
1 Scope	2
2 References	3
2.1 Normative References	3
2.2 Informative References/Bibliography	5
3 Requirements	8
3.1 Relevant Libraries	8
3.2 LSB Implementation Conformance	8
3.3 LSB Application Conformance	10
4 Terms and Definitions	11
5 Documentation Conventions	13
6 Relationship To ISO/IEC 9945 POSIX	14
7 Relationship To Other Linux Foundation Specifications	15
II Executable And Linking Format (ELF)	16
8 Introduction	17
9 Low Level System Information	18
9.1 Operating System Interface	18
9.2 Machine Interface	18
10 Object Format	19
10.1 Object Files	19
10.2 Sections	19
10.3 Special Sections	23
10.4 Symbol Mapping	29
10.5 DWARF Extensions	29
10.6 Exception Frames	31
10.7 Symbol Versioning	36
10.8 ABI note tag	40
11 Dynamic Linking	41
11.1 Program Loading and Dynamic Linking	41
11.2 Program Header	41
11.3 Dynamic Entries	41
12 C++ Class Representations	46
12.1 C++ Data Representation	46
13 Symbol Mapping	50
13.1 Symbol Mapping	50
III Base Libraries	51
14 Base Libraries	52
14.1 Introduction	52
14.2 Program Interpreter	52
14.3 Interfaces for libc	52
14.4 Data Definitions for libc	73
14.5 Interface Definitions for libc	180
14.6 Interfaces for libm	393
14.7 Data Definitions for libm	397
14.8 Interface Definitions for libm	403
14.9 Interfaces for libpthread	428
14.10 Data Definitions for libpthread	434
14.11 Interface Definitions for libpthread	439

14.12 Interfaces for libgcc_s.....	444
14.13 Data Definitions for libgcc_s.....	445
14.14 Interface Definitions for libgcc_s.....	446
14.15 Interfaces for libdl.....	452
14.16 Data Definitions for libdl.....	453
14.17 Interface Definitions for libdl.....	454
14.18 Interfaces for librt.....	457
14.19 Data Definitions for librt.....	459
14.20 Interfaces for libcrypt.....	461
14.21 Data Definitions for libcrypt.....	462
14.22 Interface Definitions for libcrypt.....	462
14.23 Interfaces for libpam.....	464
14.24 Data Definitions for libpam.....	465
14.25 Interface Definitions for libpam.....	467
IV Utility Libraries.....	481
15 Utility Libraries.....	482
15.1 Introduction.....	482
15.2 Interfaces for libz.....	482
15.3 Data Definitions for libz.....	483
15.4 Interface Definitions for libz.....	486
15.5 Interfaces for libncurses.....	533
15.6 Data Definitions for libncurses.....	538
15.7 Interface Definitions for libncurses.....	546
15.8 Interfaces for libncursesw.....	554
15.9 Data Definitions for libncursesw.....	560
15.10 Interface Definitions for libncursesw.....	587
15.11 Interfaces for libutil.....	587
15.12 Data Definitions for libutil.....	588
15.13 Interface Definitions for libutil.....	588
V C++ Libraries.....	594
16 Libraries.....	595
16.1 Interfaces for libstdcxx.....	595
16.2 Interface Definitions for libstdcxx.....	850
VI Commands and Utilities.....	851
17 Commands and Utilities.....	852
17.1 Commands and Utilities.....	852
17.2 Command Behavior.....	853
VII Execution Environment.....	917
18 File System Hierarchy.....	918
18.1 /dev: Device Files.....	918
18.2 /etc: Host-specific system configuration.....	918
18.3 User Accounting Databases.....	920
18.4 Path For System Administration Utilities.....	920
19 Additional Recommendations.....	921
19.1 Recommendations for applications on ownership and permissions.....	921
20 Additional Behaviors.....	923
20.1 Mandatory Optional Behaviors.....	923
20.2 Optional Mandatory Behaviors.....	924
20.3 Executable Scripts.....	924
21 Localization.....	926
21.1 Introduction.....	926

21.2 Regular Expressions	926
21.3 Pattern Matching Notation	926
VIII System Initialization.....	928
22 System Initialization.....	929
22.1 Cron Jobs.....	929
22.2 Init Script Actions	930
22.3 Comment Conventions for Init Scripts	932
22.4 Installation and Removal of Init Scripts.....	934
22.5 Run Levels	935
22.6 Facility Names	935
22.7 Script Names	936
22.8 Init Script Functions	936
IX Users & Groups	939
23 Users & Groups	940
23.1 User and Group Database.....	940
23.2 User & Group Names	940
23.3 User ID Ranges	941
23.4 Rationale.....	941
X Network Security Services	942
24 Libraries.....	943
24.1 Interfaces for libnspr4	943
24.2 Data Definitions for libnspr4	945
24.3 Interfaces for libnss3.....	955
24.4 Data Definitions for libnss3.....	956
24.5 Interfaces for libssl3	979
24.6 Data Definitions for libssl3.....	980
XI Package Format and Installation	990
25 Software Installation	991
25.1 Introduction	991
25.2 Package File Format.....	991
25.3 Package Script Restrictions	1011
25.4 Package Tools	1011
25.5 Package Naming Conventions	1011
25.6 Package Dependencies	1012
25.7 Package Architecture Considerations.....	1013
Annex A Alphabetical Listing of Interfaces by Library	1014
A.1 libc	1014
A.2 libcrypt	1029
A.3 libdl	1029
A.4 libgcc_s.....	1029
A.5 libm	1029
A.6 libncurses	1033
A.7 libncursesw	1036
A.8 libpam	1043
A.9 libpthread.....	1043
A.10 librt.....	1047
A.11 libutil.....	1047
A.12 libz	1048
A.13 libnspr4	1049
A.14 libnss3	1050
A.15 libssl3	1051

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

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.

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.

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.

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.