INTERNATIONAL STANDARD

ISO/IEC 23360-1-5

First edition
2021-10

Linux Standard Base (LSB) —

Part 1-5: Imaging specification

Reference number
ISO/IEC 23360-1-5:2021(E)

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): Imaging specification 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 23360-1-5 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 ... **v**
I Introductory Elements ... **1**
1 Scope ... 2
2 Normative References ... 3
3 Requirements ... 4
3.1 Relevant Libraries ... 4
4 Terms and Definitions ... 5
5 Documentation Conventions ... 7
6 PPD Format Extensions ... 8
II Printing Libraries ... **12**
7 Libraries ... 13
7.1 Interfaces for libcups ... 13
7.2 Data Definitions for libcups ... 15
7.3 Interface Definitions for libcups ... 29
7.4 Interfaces for libcupsimage ... 49
7.5 Data Definitions for libcupsimage ... 50
7.6 Interface Definitions for libcupsimage ... 53
III Printing Commands ... **56**
8 Printing Commands ... 57
8.1 Commands and Utilities ... 57
8.2 Command Behavior ... 57
IV Execution Environment ... **63**
9 File System Hierarchy ... 64
V Scanning Libraries ... **65**
10 Libraries ... 66
10.1 Interfaces for libsane ... 66
10.2 Data Definitions for libsane ... 66
VI Package Format and Installation ... **75**
11 Software Installation ... 76
11.1 Package Dependencies ... 76
Annex A Alphabetical Listing of Interfaces by Library ... **77**
A.1 libsane ... 77
A.2 libcups ... 77
A.3 libcupsimage ... 79

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 Imaging module of the Linux Standard Base (LSB). This module provides the fundamental system interfaces, libraries, and runtime environment upon which conforming applications and libraries requiring the LSB Imaging module depend.

Interfaces described in LSB Imaging are mandatory except where explicitly listed otherwise. Interfaces described in the LSB Imaging module supplement those described in the LSB Core module. They do not depend on other LSB modules.

2 Normative References

The specifications listed below are referenced in whole or in part by the LSB Imaging specification. Such references may be normative or informative; a reference to specification shall only be considered normative if it is explicitly cited as such. The LSB Imaging specification may make normative references to a portion of these specifications (that is, to define a specific function or group of functions); in such cases, only the explicitly referenced portion of the specification is to be considered normative.

Table 2-1 Normative References

Name	Title	URL
CUPS API Reference	CUPS 1.2 API Reference	http://www.cups.org/documentation.php/doc-1.2/
Filesystem Hierarchy Standard	Filesystem Hierarchy Standard (FHS) 3.0	http://refspecs.linuxbase.org/fhs
ISO C (1999)	ISO/IEC 9899:1999 - Programming Languages -- C	
PPD Specification	PostScript Printer Description File Format Specification version 4.3	http://partners.adobe.com/public/developer/en/ps/5003.PPD_Spec_v4.3.pdf
PPD Specification Update	Update to PPD Specification Version 4.3	http://partners.adobe.com/public/developer/en/ps/5645.PPD_Update.pdf
SANE Standard Version 1.04	SANE Standard Version 1.04	http://www.sane-project.org/html/

3 Requirements

3.1 Relevant Libraries

The libraries listed in Table 3-1 shall be available on a Linux Standard Base system, with the specified runtime names. This list may be supplemented or amended by an architecture-specific specification.

Table 3-1 Standard Library Names

Library	Runtime Name
libcups	libcups.so.2
libcupsimage	libcupsimage.so.2
libsane	libsane.so.1

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

4 Terms and Definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 2382, ISO 80000–2, and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

4.1

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.

4.2

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.

4.3

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.

4.4

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.

4.5

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.

4.6

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.

4.7

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 PPD Format Extensions

The Postscript Printer Description (PPD) format is used in a text file to describe device capabilities for a printing device. PPD files shall conform to the format described by PPD Specification and PPD Specification Update. In addition, several extensions to the standard attribute list are recognized, as listed below. The "cupsVersion" attribute is required in a compliant PPD, while the other attributes are optional.

cupsColorProfile

This string attribute specifies an sRGB-based color profile consisting of gamma and density controls and a 3x3 CMY color transform matrix.

The attribute has the following parameter usage:

*cupsColorProfile Resolution/MediaType: "density gamma m00 m01 m02 m10 m11 m12 m20 m21 m22"

The Resolution and MediaType values may be "-" to act as a wildcard. Otherwise, they must match one of the Resolution or MediaType attributes defined in the PPD file.

The density and gamma values define the gamma and density adjustment function such that (in terms of C math):

f(x) = density * pow(x, gamma)

The m00 through m22 values define a 3x3 transformation matrix for the CMY color values. The density function is applied after the CMY transformation:

| m00 m01 m02 | | m10 m11 m12 | | m20 m21 m22 |

cupsFax

This boolean attribute specifies whether the PPD defines a facsimile device. The default is false.

cupsFilter

The attribute has the following parameter usage:

*cupsFilter: "source/type cost program"

This string attribute provides a conversion rule from the given source type to the printer's native format using the filter "program". A source type is specified according to the conventions of the MIME specification, using "type/subtype" nomenclature, and may refer to a standard MIME type or a CUPS-specific MIME type using the prefix "vnd.cups-" in the subtype. If a printer supports the source type directly, the special filter program "-" may be specified. The cost is an arbitrary positive integer, used to calculate the relative impact a print job has on system load.

cupsManualCopies

This boolean attribute notifies the RIP filters that the destination printer does not support copy generation in hardware. The default value is false.

cupsModelNumber

This integer attribute specifies a printer-specific model number. This number can be used by a filter program to adjust the output for a specific model of printer.

cupsVersion

The attribute has the following parameter usage:

*cupsVersion: "major.minor"

This required attribute describes which version of the CUPS PPD file extensions was used. Currently it must be the string "1.0" or "1.1". The strings "1.2" and "1.3" represent newer versions of the CUPS PPD API that are not covered in this version of the specification, and are currently not allowed, although they may be found in non-conforming PPDs which use a newer version of the CUPS PPD specification.

FoomaticIDs

The attribute has the following parameter usage:

*FoomaticIDs printer driver

The parameters correspond to the IDs in the Foomatic database for the printer and driver, respectively.

FoomaticNoPageAccounting

This boolean attribute tells foomatic-rip whether or not to insert accounting information into the PostScript data stream. By default, foomatic-rip will insert this information.

FoomaticRIPCommandLine

The attribute has the following parameter usage:

*FoomaticRIPCommandLine "code"

This attribute defines the command line in the "code" parameter for the renderer that is called by foomatic-rip. The command must take PostScript on standard input and provide the job data stream in the printer's native language on standard output. The command must exit with status 0 if the conversion was successful and exit with another status if an error occurs. The "code" parameter may contain option setting wildcards, as described below under "FoomaticRIPOption".

FoomaticRIPDefault

The attribute has the following parameter usage:

*FoomaticRIPDefaultOptionName value

This attribute sets a default for a Foomatic option. The name of the attribute should contain the name of the option appended to "FoomaticRIPDefault", with the desired default value as the only parameter.

This option is only used to provide numeric options in the PPD, which are not supported by the Adobe spec, via enumerated options, and should not be used except for that purpose.

FoomaticRIPOption

The attribute has the following parameter usage:

*FoomaticRIPOption name: type style spot [order]

This attribute sets options for the command line specified in the "FoomaticRIPCommandLine" attribute. The "name" parameter specifies the option name, the "type" parameter specifies the option type, the "style"

parameter specifies one of "CmdLine", "JCL", "PS", or "Composite", and the "spot" parameter specifies a letter, which is prepended with a "%" and used in the "FoomaticRIPCommandLine" attribute to indicate where the option should go in the command line. The optional "order" parameter indicates an order number for one-choice options.

FoomaticRIPOptionAllowedChars

The attribute has the following parameter usage:

*FoomaticRIPOptionAllowedChars name: "code"

This option sets a list of allowed characters in a string option. The "name" parameter identifies the option, while the "code" parameter is a list of allowed characters.

FoomaticRIPOptionAllowedRegExp

The attribute has the following parameter usage:

*FoomaticRIPOptionAllowedRegExp name: "code"

This option causes the option named by "name" to be validated by the Perl-style regular expression in "code".

FoomaticRIPOptionMaxLength

The attribute has the following parameter usage:

*FoomaticRIPOptionMaxLength name: length

For string or password options, this attribute sets a maximum length which can be returned. The "name" parameter identifies the option, and the "length" parameter is the maximum number of characters allowed.

FoomaticRIPOptionPrototype

The attribute has the following parameter usage:

*FoomaticRIPOptionPrototype name: "code"

For string, password, or simulated numeric options, this attribute sets a code prototype to be inserted into the output. This works for options where the FoomaticRIPOption "style" parameter is set to CmdLine, JCL, or PS. The value of the option can be represented with the string "%s" in the "code" parameter.

FoomaticRIPOptionRange

The attribute has the following parameter usage:

*FoomaticRIPOptionRange name: min max

This attribute adds a minimux and maximum limit to numeric options (that are simulated by Foomatic via emumerated options). The "name" parameter identifies the option, while the "min" and "max" parameters set the minumum and maximum allowed values, respectively, for the option.

FoomaticRIPOptionSetting

The attribute has the following parameter usage:

*FoomaticRIPOptionSetting name=choice: "code"

