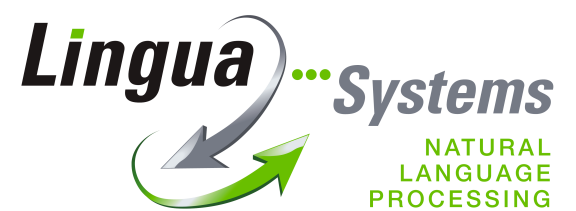


Software Specification

for

lidc

Covers version 1.3.0



1 Introduction

This document provides the software specification for `lidc`, version 1.3.0.

`lidc` is an application that identifies both language and character encoding of text, HTML, XML or email documents.

The specification serves as a general introduction to the software on the one hand and as a detailed description on the application's scope of service on the other hand.

2 Overview on the software

`lidc` is a command line application for Unix-like operating systems, that identifies language and character encoding of an input. It supports a variety of common input formats and allows for user-definable output.

2.1 User's requirements

Every user familiar with the basic concepts of using applications on the command line of Unix-like operating systems, like Linux or Solaris, will be able to use `lidc` right away.

`lidc`'s field of application is generally every field that benefits from knowing the language and/or character encoding of its relevant documents.

2.2 Operating environment

`lidc` is a command line application suited for Unix-like operating systems. Its standard version is currently available for the following operating systems:

- Debian GNU/Linux (x86): *Lenny*
- Ubuntu GNU/Linux (x86): *LTS (10.04)*
- Solaris (Sparc): *10*
- FreeBSD (x86): *7, 8*
- Mac OS X (x86): *Tiger, Leopard, Snow Leopard*

The software may very well work on other versions and or distributions without modification although `lidc` is only supported in the environments specified above.

Versions for other distributions and/or operating systems may be made available upon request as well.

2.3 Dependencies

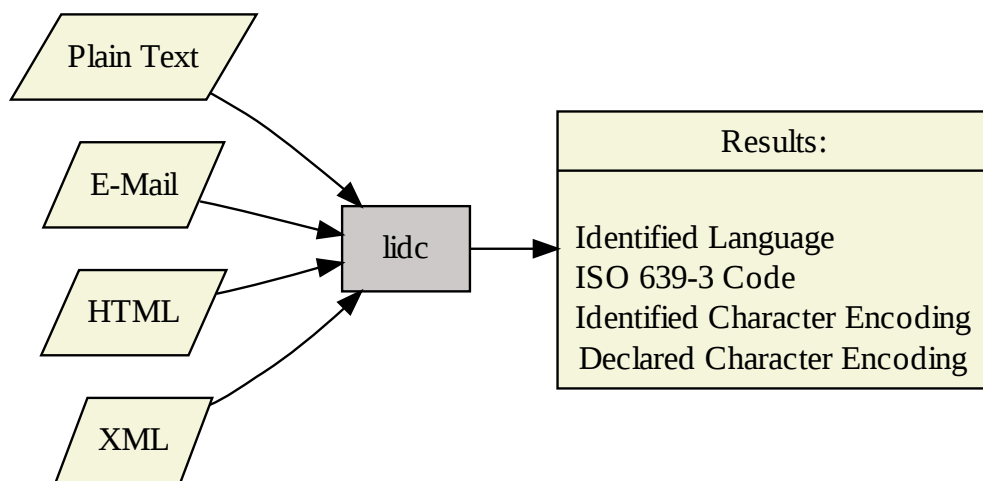
`lidc` does not require any additional software (libraries) to be installed except for the standard C and thread library shipped with the particular system.

2.4 Ressource Usage

The necessary amount of memory (RAM) operatively depends on the size and type of the actual input. At least 30 KiB will be required for operation.

3 Scope of Service

The command line application `lidc` in the current version supports 28 languages and 35 character encodings. Additionally 10 languages are identified in transliterated form as well. The input may be given in a variety of different formats. The result is user-definable and may contain information on the identified language, its ISO 639-3 code, the identified character encoding and –if provided– the declared character encoding.



The subsequent chapters provide information on the supported input formats, languages, character encodings and output capabilities. Besides that rather functional description, security and quality aspects are presented as well as restrictions of the software.

3.1 User Interface

`lidc` is a commandline application. Its functionality can be accessed and customized with a set of arguments. All arguments are described in detail in the User Manual to this software.

3.2 Input

3.2.1 Supported Input Formats

`lidc` utilizes an internal filter to process the following formats:

1. Plain Text (MIME-Type: `text/plain`)

2. HTML: HTML (any version), X-HTML
3. XML
4. Email (RFC 822)
5. Email: text/plain, text/html, multipart/mixed, multipart/alternative, multipart/digest, message/rfc822, multipart/parallel (RFC 2045-2049: MIME)
6. Email: multipart/related (RFC 2387)
7. Email: multipart/report (RFC 3462)
8. Email: multipart/signed (RFC 1847)

The type may be set as an argument on the command line or, if omitted, be automatically determined by evaluating the file's extension.

`lidc` reads its input either from file or from standard input and can thus be used as part of a pipe.

`lidc` handles the processing of UTF-16 and UTF-32 encoded files, in both Little- and Big-Endian byte order, for any input type except email.

3.2.2 Supported Languages and Character Encodings

Currently, 28 languages can be identified. The supported encodings cover commonly used encodings as well as traditional ones.

The used byte order of any UTF-16 and UTF-32 input is determined as well. In detail, these encodings are determined as either "UTF-16BE", "UTF-16LE", "UTF-32BE" or "UTF-32LE".

Language	ISO 639-3 Code	Character Encoding
Bokmål (Norwegian)	nob	UTF-32, UTF-16, UTF-8, ISO-8859-1, Windows-1252, MacRoman, CP 850, ASCII
Bulgarian	bul	UTF-32, UTF-16, UTF-8, ISO-8859-5, Windows-1251, MacCyrillic, CP 855, CP 866, KOI8-R
Czech	ces	UTF-32, UTF-16, UTF-8, ISO-8859-2, Windows-1250, MacCentralEurope, CP 852
Danish	dan	UTF-32, UTF-16, UTF-8, ISO-8859-1, Windows-1252, MacRoman, CP 850, ASCII
Dutch	nld	UTF-32, UTF-16, UTF-8, ISO-8859-1, Windows-1252, MacRoman, CP 850, ASCII
English	eng	UTF-32, UTF-16, UTF-8, ISO-8859-1, Windows-1252, MacRoman, CP 850, ASCII
Estonian	est	UTF-32, UTF-16, UTF-8, ISO-8859-4, Windows-1257, MacCentralEurope, CP 775, ASCII
Finnish	fin	UTF-32, UTF-16, UTF-8, ISO-8859-1, Windows-1252, MacRoman, CP 850, ASCII
French	fra	UTF-32, UTF-16, UTF-8, ISO-8859-1, Windows-1252, MacRoman, CP 850, ASCII
German	deu	UTF-32, UTF-16, UTF-8, ISO-8859-1, Windows-1252, MacRoman, CP 850, ASCII

Language	ISO 639-3 Code	Character Encoding
Greek	ell	UTF-32, UTF-16, UTF-8, ISO-8859-7, Windows-1253, MacGreek, CP 737
Hungarian	hun	UTF-32, UTF-16, UTF-8, ISO-8859-2, ISO-8859-16, Windows-1250, MacCentralEurope, CP 852
Irish (Gaelic)	gle	UTF-32, UTF-16, UTF-8, ISO-8859-1, Windows-1252, MacRoman, CP 850, ASCII
Italian	ita	UTF-32, UTF-16, UTF-8, ISO-8859-1, ISO-8859-16, Windows-1252, MacRoman, CP 850, ASCII
Lithuanian	lit	UTF-32, UTF-16, UTF-8, ISO-8859-4, Windows-1257, MacCentralEurope, CP 775, ASCII
Latvian	lav	UTF-32, UTF-16, UTF-8, ISO-8859-4, Windows-1257, MacCentralEurope, CP 775, ASCII
Maltese	mlt	UTF-32, UTF-16, UTF-8, ISO-8859-3
Mandarin (Chinese)	cmn	UTF-32, UTF-16, UTF-8, Big5, GB2312
Nynorsk (Norwegian)	nno	UTF-32, UTF-16, UTF-8, ISO-8859-1, Windows-1252, MacRoman, CP 850, ASCII
Polish	pol	UTF-32, UTF-16, UTF-8, ISO-8859-2, ISO-8859-16, Windows-1250, MacCentralEurope, CP 852
Portuguese	por	UTF-32, UTF-16, UTF-8, ISO-8859-1, ISO-8859-15, Windows-1252, MacRoman, CP 850, ASCII
Romanian	ron	UTF-32, UTF-16, UTF-8, ISO-8859-2, Windows-1250, MacRomanian, CP 852
Russian	rus	UTF-32, UTF-16, UTF-8, ISO-8859-5, Windows-1251, MacCyrillic, CP 855, CP 866, KOI8-R
Swedish	swe	UTF-32, UTF-16, UTF-8, ISO-8859-1, Windows-1252, MacRoman, CP 850, ASCII
Slovak	slk	UTF-32, UTF-16, UTF-8, ISO-8859-2, Windows-1250, MacCentralEurope, CP 852
Slovenian	slv	UTF-32, UTF-16, UTF-8, ISO-8859-2, ISO-8859-16, Windows-1250, MacCentralEurope, CP 852, ASCII
Spanish	spa	UTF-32, UTF-16, UTF-8, ISO-8859-1, ISO-8859-15, Windows-1252, MacRoman, CP 850, ASCII
Ukrainian	ukr	UTF-32, UTF-16, UTF-8, Windows-1251, MacUkrainian, KOI8-U

In addition to the languages, 10 languages can be recognized in the transliterated forms. The supported transliterations are given here. "common" denotes transliterations that are not according to a standard but are commonly used.

Language	Transliteration	Character Encodings
Bulgarian	ISO 9	UTF-32, UTF-16, UTF-8, ASCII
	DIN 1460	UTF-32, UTF-16, UTF-8, ASCII, Windows-1250
	Streamlined System	UTF-32, UTF-16, UTF-8, ASCII, Windows-1250
Czech	common	UTF-32, UTF-16, UTF-8, ASCII, ISO-8859-1
German	common	UTF-32, UTF-16, UTF-8, ASCII, ISO-8859-1
Greek	ISO 843	UTF-32, UTF-16, UTF-8, ASCII
	DIN 31634	UTF-32, UTF-16, UTF-8, ASCII
	Greeklisch	UTF-32, UTF-16, UTF-8, ASCII, ISO-8859-1
Polish	common	UTF-32, UTF-16, UTF-8, ASCII, ISO-8859-1
Romanian	common	UTF-32, UTF-16, UTF-8, ASCII, ISO-8859-1

Language	Transliteration	Character Encodings
Russian	ISO 9	UTF-32, UTF-16, UTF-8
	DIN 1460	UTF-32, UTF-16, UTF-8
Slovak	common	UTF-32, UTF-16, UTF-8, ASCII, ISO-8859-1
Slovenian	common	UTF-32, UTF-16, UTF-8, ASCII, ISO-8859-1
Ukrainian	ISO 9	UTF-32, UTF-16, UTF-8
	DIN 1460	UTF-32, UTF-16, UTF-8

3.3 Output

3.3.1 Results

The following pieces of information are available as a result:

- The identified language
The name of the language is available as well as its ISO 639-3 code.
- The identified character encoding
The encoding is given according to the list specified in chapter 3.2.2
- The declared character encoding (if available)
If a document contains a character encoding declaration, its value will be made available (in lowercase letters) as a result as well.
- The name of the used input

3.3.2 Output handling

`lidc` prints its results according to a (user-definable) format string on the command line (`stdout`).

If the format string should be suited to any special requirements, `lidc` provides a set of flags to reference all computed results. While sending the output to the standard output stream, `lidc` will substitute all flags with their associated values: the identified language (`%1`), its ISO

639-3 code (%i), the identified character encoding (%e), the (probably) declared character encoding (%d) and the input file's path (%f). Beside that, the format string may contain ordinary characters and escape sequences, so obtaining an immediate CSV, HTML or XML output is possible and easy to achieve.

3.4 Error Handling

`lidc` prints an appropriate message whenever an error is encountered and terminates with an error code.

3.5 Security Considerations

`lidc` was designed and implemented with security in mind and has no single known safety defect. Beside that, the application does not create any temporary files or evaluate any environment variables.

3.6 Quality Characteristics

`lidc` works both fast and reliable and can handle a variety of common input formats. The performance depends on the underlying hardware as well as on the overall business of the machine.

3.7 Restrictions

At the present state of the art, an identification of language and/or character encoding cannot be guaranteed to be accurate in any case without restrictions. There may be documents that lead to wrong identification results.

An input can only be processed accurately if it is available in an supported input type (see chapter 3.2.1) or has been preprocessed to such a format before.

Only supported languages and character encodings (see chapter 3.2.2) can be identified.

Unsupported languages and character encodings will be identified as the most similar language and/or character encoding.

Multilingual documents cannot be processed unless they contain no more than a few passages in another language.

Any input should contain at least 25 characters of textual data that form different words.

While processing multipart *emails* (*MIME*), only the first message body will be evaluated.

The software itself is provided in English only.

The manual is solely provided as a PDF document, in both an English and German version.

4 Deliveries

A complete installation of the software consists of the command line application, its man page and the user manual in an English and German version. The installation requires about 2 MiB of disk space.

5 References

- Lingua-Systems' [lidc](http://www.lingua-systems.com/language-identifier/lidc-application/) product website, <http://www.lingua-systems.com/language-identifier/lidc-application/>
- Concerning E-Mail:
 - RFC 2045, 2046, 2047, 2049 - "Multipurpose Internet Mail Extensions"
 - RFC 822 - "Standard for the Format of ARPA Internet Text Messages"
 - RFC 2387 - "The MIME Multipart/Related Content-type"
 - RFC 1847 - "Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted"
 - RFC 3462 - "The Multipart/Report Content Type for the Reporting of Mail System Administrative Messages"
- Concerning HTML:
 - W3C XHTML 1.0 Specification: "The Extensible HyperText Markup Language"
 - W3C HTML 4.01 Specification
- Concerning XML:
 - W3C Extensible Markup Language (XML) 1.0 Specification
- Language Codes : ISO 639-3:2007 - "Codes for the representation of names of languages"