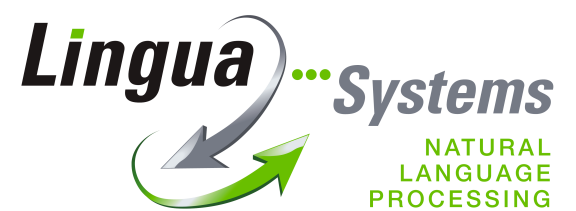


# Software Spezifikation

für

lid

Beschreibt Version 3.1.0



# 1 Einführung

---

Dieses Dokument stellt die Software Spezifikation für `lid` Version 3.1.0 dar.

`lid` ist eine dynamische C/C++ Bibliothek, mit der sich sowohl die Sprache als auch die Zeichenkodierung einer Text-Eingabe bestimmen lässt.

Diese Spezifikation liefert eine allgemeine Beschreibung der Software und eine detaillierte Beschreibung des Leistungsumfangs.

Hinweis auf Konventionen:

An einigen Stellen ist es notwendig, zwischen Zeichenketten zu unterscheiden, die keine `NUL` Zeichen enthalten dürfen (und in Folge dessen kompatibel zu den Funktionen der Standard C Bibliothek sind), und solchen, die auf Grund besonderer Zeichenkodierungen potentiell `NUL` Zeichen aufweisen können. Erstere werden im Folgenden „Strings“ genannt, letzte „Byte Strings“.

## 2 Allgemeine Beschreibung der Software

---

`lid` ist eine dynamische Bibliothek, die Sprache und Zeichenkodierung einer Eingabe bestimmt. Die Eingabe muss in reinem Textformat vorliegen und kann als Datei oder als String übergeben werden. Die Rückgabe der Ergebnisse erfolgt in einer Datenstruktur.

### 2.1 Benutzermerkmale

Die Software kann von jedem Benutzer verwendet werden, der über Grundkenntnisse in der C/C++ Programmierung und der Verwendung von Bibliotheken verfügt.

Es handelt sich bei `lid` um eine Expertensoftware, die theoretisch in jedem Umfeld gewinnbringend zum Einsatz gebracht werden kann, bei dem die Erkennung von Sprache oder Zeichenkodierung eines Textes erforderlich oder hilfreich ist.

### 2.2 Betriebsumgebung

`lid` ist zum Einsatz unter Unix-artigen Betriebssystemen konzipiert, steht aber auch für verschiedene Windows-Betriebssysteme mit dem gleichen Funktionsumfang zur Verfügung. Die Standard-Version der Software kann für die folgenden Betriebssysteme bezogen werden:

- Debian GNU/Linux (x86): *Etch, Lenny*
- Ubuntu GNU/Linux (x86/IA-32): *LTS (10.04)*
- Solaris (Sparc): *Solaris 10*
- FreeBSD (x86): *6, 7, 8*
- Microsoft Windows (x86/IA-32): *XP, Server 2003, Vista, Server 2008, 7*
- Microsoft Windows (x86\_64/IA-64): *7, Server 2008 R2*

Die korrekte Funktionalität wird unter anderen Distributionen und/oder Versionen eines Betriebssystems zu großer Wahrscheinlichkeit ebenfalls gegeben sein, ist allerdings nicht Umfang der Gewährleistung.

Versionen für weitere Betriebssysteme und/oder Distributionen können auf Anfrage ebenfalls bereitgestellt werden.

## 2.3 Abhängigkeiten

`lid` weist außer der jeweiligen Standard-C- und Thread-Bibliothek des Systems keine weiteren Abhängigkeiten auf.

## 2.4 Ressourcenverbrauch

`lid` arbeitet sehr effizient und benötigt im Schnitt – unabhängig von der Größe der Eingabe – nicht mehr als 285 KiB Arbeitsspeicher.

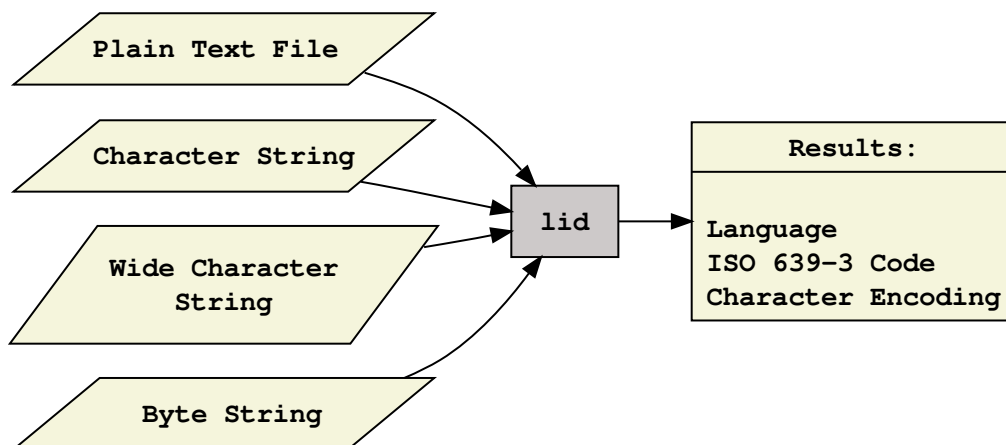
`lid` allokiert zwecks thread-sicherer Fehlerbehandlung zwei Variablen auf dem Thread-Local Storage (TLS) eines jeden Threads, der die bereitgestellten Funktionen verwendet.

## 3 Leistungsumfang der Software

---

Die Bibliothek `lid` bestimmt in der hier beschriebenen Version 26 Sprachen und 35 Zeichenkodierungen. Zusätzlich werden 10 Sprachen in transliterierter Form unterstützt. Als Eingabe können Textdateien und verschiedene String-Datentypen dienen. Das Ergebnis umfasst die erkannte Sprache, ihren ISO 639-3 Code und die erkannte Zeichenkodierung.

`lid` ist thread-safe und kann von mehreren Threads gleichzeitig verwendet werden.



### 3.1 Eingabe

#### 3.1.1 Unterstützte Eingabeformate

`lid` verarbeitet nur Eingaben, die im reinen Textformat (MIME-Type: `text/plain`) vorliegen.

Als Eingabe können dienen:

- Dateien
- Strings
  - Character String (`char *`)
  - Wide Character String (`wchar_t *`)
  - Byte String (`char *`)

Die Länge der Eingabe wird von `lid` nur durch die entsprechenden Datentypen begrenzt.

#### 3.1.2 Unterstützte Sprachen und Zeichenkodierungen

Es können die derzeit 26 Sprachen erkannt werden. Die 35 unterstützten Zeichenkodierungen umfassen sowohl gängige als auch ältere Zeichenkodierungen für die jeweiligen Sprachen.

Die einer UTF-16 oder UTF-32 kodierten Eingabe zugrundeliegende Bytereihenfolge wird ebenfalls bestimmt und durch „UTF-16BE“, „UTF-16LE“, „UTF-32BE“ und „UTF-32LE“ mit angegeben.

Sprache	ISO 639-3 Code	Zeichenkodierungen
Bulgarisch	bul	UTF-32, UTF-16, UTF-8, ISO-8859-5, Windows-1251, MacCyrillic, CP 855, CP 866, KOI8-R
Dänisch	dan	UTF-32, UTF-16, UTF-8, ISO-8859-1, Windows-1252, MacRoman, CP 850, ASCII
Deutsch	deu	UTF-32, UTF-16, UTF-8, ISO-8859-1, Windows-1252, MacRoman, CP 850, ASCII
Englisch	eng	UTF-32, UTF-16, UTF-8, ISO-8859-1, Windows-1252, MacRoman, CP 850, ASCII
Estnisch	est	UTF-32, UTF-16, UTF-8, ISO-8859-4, Windows-1257, MacCentralEurope, CP 775, ASCII
Finnisch	fin	UTF-32, UTF-16, UTF-8, ISO-8859-1, Windows-1252, MacRoman, CP 850, ASCII
Französisch	fra	UTF-32, UTF-16, UTF-8, ISO-8859-1, Windows-1252, MacRoman, CP 850, ASCII
Griechisch	ell	UTF-32, UTF-16, UTF-8, ISO-8859-7, Windows-1253, MacGreek, CP 737
Irish (Gälisch)	gle	UTF-32, UTF-16, UTF-8, ISO-8859-1, Windows-1252, MacRoman, CP 850, ASCII
Italienisch	ita	UTF-32, UTF-16, UTF-8, ISO-8859-1, Windows-1252, MacRoman, CP 850, ASCII
Lettisch	lav	UTF-32, UTF-16, UTF-8, ISO-8859-4, Windows-1257, MacCentralEurope, CP 775, ASCII
Litauisch	lit	UTF-32, UTF-16, UTF-8, ISO-8859-4, Windows-1257, MacCentralEurope, CP 775, ASCII
Maltesisch	mlt	UTF-32, UTF-16, UTF-8, ISO-8859-3
Mandarin (Chinesisch)	cmn	UTF-32, UTF-16, UTF-8, Big5, GB2312
Niederländisch	nld	UTF-32, UTF-16, UTF-8, ISO-8859-1, Windows-1252, MacRoman, CP 850, ASCII
Polnisch	pol	UTF-32, UTF-16, UTF-8, ISO-8859-2, Windows-1250, MacCentralEurope, CP 852
Portugiesisch	por	UTF-32, UTF-16, UTF-8, ISO-8859-1, Windows-1252, MacRoman, CP 850, ASCII
Rumänisch	ron	UTF-32, UTF-16, UTF-8, ISO-8859-2, Windows-1250, MacRomanian, CP 852
Russisch	rus	UTF-32, UTF-16, UTF-8, ISO-8859-5, Windows-1251, MacCyrillic, CP 855, CP 866, KOI8-R
Schwedisch	swe	UTF-32, UTF-16, UTF-8, ISO-8859-1, Windows-1252, MacRoman, CP 850, ASCII
Slowakisch	slk	UTF-32, UTF-16, UTF-8, ISO-8859-2, Windows-1250, MacCentralEurope, CP 852

Sprache	ISO 639-3 Code	Zeichenkodierungen
Slowenisch	slv	UTF-32, UTF-16, UTF-8, ISO-8859-2, ISO-8859-16, Windows-1250, MacCentralEurope, CP 852, ASCII
Spanisch	spa	UTF-32, UTF-16, UTF-8, ISO-8859-1, ISO-8859-15, Windows-1252, MacRoman, CP 850, ASCII
Tschechisch	ces	UTF-32, UTF-16, UTF-8, ISO-8859-2, Windows-1250, MacCentralEurope, CP 852
Ukrainisch	ukr	UTF-32, UTF-16, UTF-8, Windows-1251, MacUkrainian, KOI8-U
Ungarisch	hun	UTF-32, UTF-16, UTF-8, ISO-8859-2, ISO-8859-16, Windows-1250, MacCentralEurope, CP 852

Darüber hinaus werden die folgenden 10 Sprachen auch dann erkannt, wenn sie in einer der 12 unterstützten Transliterationen vorliegen:

Sprache	Transliteration	Zeichenkodierungen
Bulgarisch	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
Deutsch	gebräuchlich	UTF-32, UTF-16, UTF-8, ASCII, ISO-8859-1
Griechisch	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
Polnisch	gebräuchlich	UTF-32, UTF-16, UTF-8, ASCII, ISO-8859-1
Rumänisch	gebräuchlich	UTF-32, UTF-16, UTF-8, ASCII, ISO-8859-1
Russisch	ISO 9	UTF-32, UTF-16, UTF-8
	DIN 1460	UTF-32, UTF-16, UTF-8
Slowakisch	gebräuchlich	UTF-32, UTF-16, UTF-8, ASCII, ISO-8859-1
Slowenisch	gebräuchlich	UTF-32, UTF-16, UTF-8, ASCII, ISO-8859-1
Tschechisch	gebräuchlich	UTF-32, UTF-16, UTF-8, ASCII, ISO-8859-1
Ukrainisch	ISO 9	UTF-32, UTF-16, UTF-8
	DIN 1460	UTF-32, UTF-16, UTF-8

„Gebräuchlich“ denotiert hierbei die üblicherweise angewendeten, nicht standardisierten Transliterationen, die dennoch häufig zur Anwendung kommen.

## 3.2 Rückgabe

### 3.2.1 Ergebnisse

Die folgenden Werte werden als Ergebnisse zurückgeliefert:

- Der englischsprachige Name der erkannten Sprache
- Der ISO 639-3 Code der Sprache
- Die erkannte Zeichenkodierung

### 3.2.2 Format

Die formale Definition der von **lid** zurückgelieferten Datenstruktur lautet:

```
typedef struct lid {
    char *language;      /* z.B. "Bulgarian" */
    char *encoding;     /* z.B. "Windows-1252" */
    char *isocode;     /* z.B. "bul" */
} lid_t;
```

### 3.3 Benutzerschnittstelle

Die gesamte Funktionalität wird durch (Bibliotheks-) Funktionen bereitgestellt.

Für das Bestimmen der Sprache und Zeichenkodierung gibt es jeweils eine Funktion je nach Eingabeformat/Datentyp.

Zusätzlich wird eine Funktion zur Fehlerbehandlung und zum Freigeben des allokierten Speichers bereitgestellt.

Die einzelnen Funktionen mit ihren Parametern und Rückgabewerten werden im Benutzerhandbuch zur Software aufgeführt.

### 3.4 Fehlerbehandlung

**lid** stellt eine fachgerechte Fehlerbehandlung zur Verfügung, die Fehlercodes verwendet und eine Fehlerverwaltung pro Thread erlaubt. Anhand der Fehlercodes können englischsprachige Fehlerbeschreibungen mit Hilfe einer Bibliotheksfunktion generiert werden. Für alle definierten Fehlercodes werden benannte Konstanten bereitgestellt, sodass bei der Anwendungsentwicklung Fehlernamen statt -nummern verwendet werden können.

### 3.5 Sicherheitsaspekte

**lid** wurde auch im Hinblick auf Sicherheitsaspekte entwickelt und weist keine bekannten Sicherheitsmängel auf. Darüber hinaus legt die Software keine temporären Dateien im Dateisystem an und wertet keine Umgebungsvariablen aus.

Von **lid** intern allokiertes Speicher wird auch in allen bekannten Fehlerfällen freigegeben.

### 3.6 Einschränkungen

- Eine 100%ige Erkennung von Sprache und Zeichenkodierung kann nach dem gegenwärtigen Stand der Technik auch von **lid** nicht in jedem Fall uneingeschränkt garantiert werden. Es kann somit bei einigen Dokumenten vorkommen, dass eine falsche Sprache und/oder Zeichenkodierung von der Bibliothek erkannt wird.
- Eine Eingabe kann nur dann sicher verarbeitet werden, wenn sie im reinen Textformat vorliegt oder im Vorfeld konvertiert wurde. Die Konvertierung gehört nicht zum Leistungsumfang der **lid** Bibliothek.

- Es können nur unterstützte Sprachen und Zeichenkodierungen erkannt werden.
- Sprachen und Kodierungen, die nicht unterstützt sind, werden trotzdem der Sprache und/oder Kodierung zugewiesen, der sie am ähnlichsten sind.
- Ebenso können keine mehrsprachigen Dokumente verarbeitet werden, ausgenommen Texte mit kleineren fremdsprachigen Einschüben.
- Eine zu erkennende Eingabe sollte eine bestimmte Mindestlänge von 25 Zeichen in verschiedenen Wörtern aufweisen.
- Die Software liegt nur in englischer Sprache vor.
- Das Handbuch ist sowohl auf Deutsch als auch auf Englisch verfügbar. Es wird ausschließlich in Form einer PDF Datei bereitgestellt.
- Es besteht kein Anspruch auf Abwärtskompatibilität zu vorherigen Versionen.

## 4 Auslieferung

---

Eine vollständige Installation der Software umfasst die kompilierte Bibliothek, alle notwendigen Header-Dateien, das Benutzerhandbuch in deutscher und englischer Sprache und Manual-Seiten (in Englisch). Die Software belegt – je nach System – etwa 1,5 MiB Festplattenspeicher.

## 5 Referenzen

---

- Text Format: MIME-Type: text/plain, RFC 2046 [3]
- Sprachkennungen : ISO 639-3:2007 *Codes for the representation of names of languages*
- Benutzerhandbuch für **lid** Version 3.1.0