

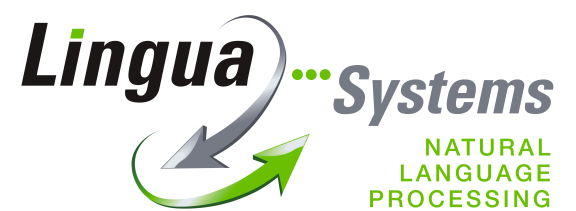
Benutzerhandbuch

für

lidc

Ein Programm zur Bestimmung von Sprache und Zeichenkodierung

Beschreibt Version 1.1.0



lfdc Benutzerhandbuch vom 3. März 2010.

Copyright © 2009 - 2010, Lingua-Systems Software GmbH.

Lingua-Systems Software GmbH, Wiesenstraße 34, 44653 Herne , info@lingua-systems.com

Das vorliegende Handbuch ist urheberrechtlich geschützt. Alle Rechte vorbehalten, insbesondere die Veränderung des Originalwerks oder die Veröffentlichung in Auszügen bedürfen der vorherigen schriftlichen Genehmigung.

Die Rechte zum Erstellen und Verbreiten von unveränderten Kopien auf Datenträgern, Papier oder im Internet, der Übersetzung und des Vortrags werden gewährt.

Genannte Hard- und Software sowie Firmennamen können eingetragene Warenzeichen sein, auch wenn sie nicht gesondert als Marken gekennzeichnet sind. Die fehlende Kennzeichnung berechtigt nicht zu der Annahme, dass diese Namen im Sinne von Warenzeichen- oder Markenschutzgesetzgebung als frei zu betrachten wären.

Das Handbuch wurde mit größter Sorgfalt erstellt. Dennoch können Fehler nicht generell ausgeschlossen werden. Für Folgen, die sich möglicherweise aus Fehlern im Handbuch ergeben, können die Autoren und der Herausgeber keine juristische Verantwortung oder Haftung übernehmen.

Des Weiteren sei darauf hingewiesen, dass alle verlinkten Internetseiten zum Zeitpunkt der Erstellung des Handbuchs eingehend geprüft worden sind. Für den Inhalt der Seiten und im Besonderen Veränderungen der Seiten können die Autoren und der Herausgeber keine Verantwortung oder Haftung übernehmen.

Sollten Ihnen Fehler auffallen oder Sie Probleme mit einer verlinkten Internetseite feststellen, werden Hinweise dankbar entgegen genommen unter support@lingua-systems.de.

Inhaltsverzeichnis

1	Einführung	5
2	Unterstützte Eingabe-Typen	5
3	Unterstützte Sprachen und Zeichenkodierungen	6
4	Installation	8
4.1	Voraussetzungen	8
4.2	Installationsumfang	8
4.3	Die Software installieren	8
4.3.1	Die Software auf Debian GNU/Linux installieren	8
4.3.2	Die Software auf Solaris installieren	9
4.3.3	Die Software auf FreeBSD installieren	9
4.4	Die Software deinstallieren	10
4.4.1	Die Software auf Debian GNU/Linux deinstallieren	10
4.4.2	Die Software auf Solaris deinstallieren	10
4.4.3	Die Software auf FreeBSD deinstallieren	10
5	lidc verwenden	11
5.1	Kurzreferenz	12
5.2	Optionen	12
5.3	Die Ausgabe formatieren	13
5.3.1	Beispiele	14
5.4	Die Ergebnisse zur Zeichenkodierung auswerten	15
5.4.1	Erkannte Zeichenkodierung	15
5.4.2	Angegebene Zeichenkodierung	15
6	Fehler vermeiden	16
A	Referenzen	17

Zu diesem Handbuch

Dieses Handbuch richtet sich an Benutzer, die mit der Verwendung von Programmen auf der Kommandozeile vertraut sind.

In diesem Handbuch wird nach einem kurzen Überblick über das Programm mit seinen unterstützten Eingabe-Typen, Sprachen und Zeichenkodierungen beschrieben, wie *lidc* zu installieren ist. Danach wird auf die Verwendung mit allen Optionen sowie die Formatierung der Ausgabe, ebenso wie die Auswertung der Ergebnisse eingegangen. Im Anschluss daran werden Hinweise gegeben, um mögliche Fehlerquellen zu vermeiden.

Für einen schnellen Einstieg sei direkt auf die Kurzreferenz (Kapitel 5.1, Seite 12) verwiesen.

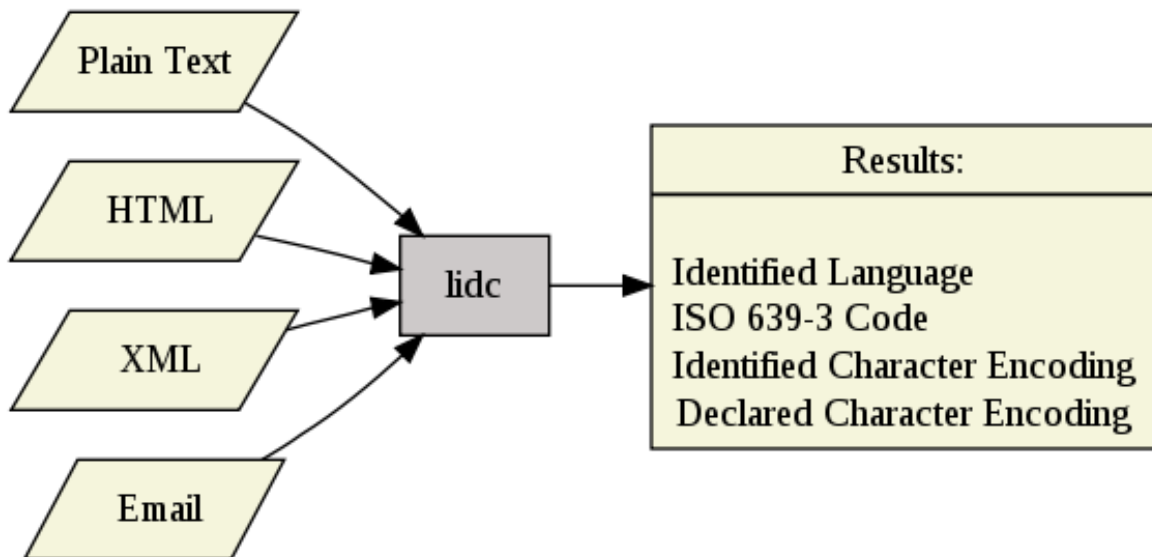
Administratoren, die die Software installieren wollen, finden alle dazu notwendigen Informationen in Kapitel 4 auf Seite 8.

1 Einführung

lidc ist ein Programm für die Kommandozeile, das sowohl die Sprache als auch die Zeichenkodierung einer Texteingabe bestimmt.

Als Eingabe können Dateien verschiedener Typen verwendet werden. Diese umfassen reine Text-Dokumente, *HTML*-, *XML*- Dokumente und E-Mails. Die Eingabe kann sowohl direkt über die Standardeingabe (*stdin*), also auch mittels Kommando-Verkettung (Pipe) übergeben werden.

Die Ausgabe der Ergebnisse umfasst die ermittelte Sprache, ihren dreistelligen Code nach ISO 639-3, die ermittelte Zeichenkodierung und - falls vorhanden - die angegebene Zeichenkodierung des Dokuments. Die Ausgabe kann mit Hilfe eines Formatstrings nach Belieben angepasst werden.



Diese Version von *lidc* unterstützt 25 Sprachen und 33 Kodierungen. Zudem werden zehn Sprachen auch in transliterierter Form erkannt. Die Funktionen zur Bestimmung werden von der zugrundeliegenden Bibliothek *lid* zur Verfügung gestellt.

Das Programm ermittelt die Ergebnisse schnell und zuverlässig. Es weist außer der Standard C-Bibliothek keine weiteren Abhängigkeiten auf. Dadurch lässt sich *lidc* auf allen unterstützten Systemen einfach integrieren und arbeitet auch auf leistungsschwächerer Hardware effektiv.

2 Unterstützte Eingabe-Typen

Das Programm kann als Eingabe sowohl reinen Text, *HTML*, *XML* und E-Mail verarbeiten. Die folgende Auflistung gibt einen detaillierten Überblick über die einzelnen Typen, insbesondere für E-Mail Formate.

1. Plain Text
2. HTML: HTML (4.0, ...), X-HTML
3. XML
4. E-Mail (RFC 822)
5. E-Mail: text/plain, text/html, multipart/mixed, multipart/alternative, multipart/digest, message/rfc822, multipart/parallel (RFC 2045-2049: MIME)
6. E-Mail: multipart/related (RFC 2387)
7. E-Mail: multipart/report (RFC 3462)
8. E-Mail: multipart/signed (RFC 1847)

3 Unterstützte Sprachen und Zeichenkodierungen

Es können die derzeit 23 offiziellen Sprachen der Europäischen Union sowie Russisch und Ukrainisch erkannt werden. Die 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.

Diese Kodierungen sind für den Eingabe-Typ *E-Mail* nicht unterstützt.

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, ISO-8859-15, 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, ISO-8859-15, Windows-1252, MacRoman, CP 850, ASCII
Französisch	fra	UTF-32, UTF-16, UTF-8, ISO-8859-1, ISO-8859-15, 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, ISO-8859-16, 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
Niederländisch	nld	UTF-32, UTF-16, UTF-8, ISO-8859-1, ISO-8859-15, Windows-1252, MacRoman, CP 850, ASCII
Polnisch	pol	UTF-32, UTF-16, UTF-8, ISO-8859-2, ISO-8859-16, Windows-1250, MacCentralEurope, CP 852
Portugiesisch	por	UTF-32, UTF-16, UTF-8, ISO-8859-1, ISO-8859-15, Windows-1252, MacRoman, CP 850, ASCII

Sprache	ISO 639-3 Code	Zeichenkodierungen
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
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

Zusätzlich zu den Sprachen können zehn Sprachen auch in transliterierter Form erkannt werden. Es werden sowohl Transliterationen nach anerkannten Standards, als auch herkömmliche Transliterationen unterstützt, wie sie sich zum Beispiel im E-Mail-Verkehr finden.

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

Eine Erkennung von Sprache und Kodierung ist nur für die Sprachen möglich, die derzeit unterstützt werden. Sollte eine Texteingabe in einer anderen als dieser Sprachen oder Kodierungen vorliegen, wird kein Fehler aufgeworfen. *lidc* wird der Eingabe die ähnlichste Sprache und Kodierung zuweisen und zurückgeben.

4 Installation

4.1 Voraussetzungen

lidc erfordert lediglich die vom System bereitgestellte Standard-C-Bibliothek und weist darüber hinaus keinerlei Softwareabhängigkeiten auf.

Die an die zugrundeliegende Hardware gestellten Anforderungen sind minimal. So benötigt *lidc*, abhängig von Typ und Größe der Eingabe, üblicherweise etwa zwischen 50 und 200 KiB Arbeitsspeicher und ist sehr performant.

4.2 Installationsumfang

Das *lidc* Softwarepaket umfasst neben der Anwendung selbst noch eine detaillierte Man Page sowie dieses Benutzerhandbuch in den Sprachen Deutsch und Englisch.

Nachfolgend findet sich exemplarisch der Verzeichnisbaum einer Installation unter Debian GNU/Linux:

```
/opt/ls/  
|-- bin  
|   '-- lidc  
'-- share  
    |-- doc  
    |   '-- lidc  
    |       |-- copyright  
    |       |-- changelog.gz  
    |       |-- manual_deu.pdf  
    |       '-- manual_eng.pdf  
    '-- man  
        '-- man1  
            '-- lidc.1.gz
```

4.3 Die Software installieren

Das Softwarepaket liegt für die einzelnen unterstützten Betriebssysteme/Distributionen jeweils im nativen Paketformat vor, so dass es sich von einem Administrator unter Zuhilfenahme der systemeigenen Standardwerkzeuge wie gewohnt einspielen lässt.

4.3.1 Die Software auf Debian GNU/Linux installieren

Mit administrativen Rechten lässt sich das Paket wie üblich unter Verwendung von `dpkg(1)` einspielen:

```
debian-box% dpkg --install lidc_1.1.0-1_i386.deb
```

Das Softwarepaket ist kompatibel zum *Filesystem Hierarchie Standard Version 2.3* und installiert alle Dateien unterhalb von `/opt/ls/`.

4.3.2 Die Software auf Solaris installieren

Mit administrativen Rechten lässt sich die Installation von *lidc* unter Solaris mit Hilfe von `pkgadd(1)` wie folgt durchführen:

```
solaris-box% gzip -d LSlidc-1.1.0-sol10-sparc-opt.pkg.gz
solaris-box% pkgadd -d LSlidc-1.1.0-sol10-sparc-opt.pkg

The following packages are available:
  1  LSlidc      lidc (/opt)
      (sparc) 1.1.0

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: 1

Processing package instance <LSlidc> from \
  <LSlidc-1.1.0-sol10-sparc-opt.pkg>

lidc (/opt)(sparc) 1.1.0
Lingua-Systems Software GmbH

...

Installation of <LSlidc> was successful.
```

Die Software wird unterhalb von `/opt/ls/` installiert.

4.3.3 Die Software auf FreeBSD installieren

Mit administrativen Rechten lässt sich die Installation von *lidc* unter FreeBSD mit Hilfe von `pkg_add(1)` wie folgt durchführen:

```
freebsd-box% pkg_add lidc-1.1.0_1.tbz
```

Die Software wird unterhalb von `/usr/local/` installiert.

4.4 Die Software deinstallieren

Da das *lidc* Softwarepaket jeweils im nativen Paketformat der einzelnen unterstützten Betriebssysteme/Distributionen vorliegt, lässt es sich von einem Administrator unter Zuhilfenahme der systemeigenen Standardwerkzeuge wie gewohnt wieder entfernen.

4.4.1 Die Software auf Debian GNU/Linux deinstallieren

Mit administrativen Rechten lässt sich *lidc* wie üblich unter Verwendung von `dpkg(1)` wieder entfernen:

```
debian-box% dpkg --remove lidc
```

4.4.2 Die Software auf Solaris deinstallieren

Mit administrativen Rechten lässt sich *lidc* wie üblich unter Verwendung von `pkgrm(1)` wieder entfernen:

```
solaris-box% pkgrm LSlidc

The following package is currently installed:
  LSlidc  lidc (/opt)
          (sparc) 1.1.0

Do you want to remove this package? [y,n,?,q] y

...

Removal of <LSlidc> was successful.
```

4.4.3 Die Software auf FreeBSD deinstallieren

Mit administrativen Rechten lässt sich *lidc* wie üblich unter Verwendung von `pkg_delete(1)` wieder entfernen:

```
freebsd-box% pkg_delete lidc-1.1.0_1
```

5 lidc verwenden

Der Aufruf des Programms erfolgt über die Kommandozeile. Mit Optionen können die Eingabe und die Ausgabe bestimmt werden. Welche Optionen zur Verfügung stehen, wird im Kapitel „Optionen“ (5.2, Seite 12) beschrieben.

Die Eingabe kann als Datei oder direkt über die Standardeingabe (`stdin`), also auch mittels Kommando-Verkettung (Pipe), übergeben werden. Das folgende Beispiel zeigt den Aufruf mit einer Datei.

```
$ lidc -i website.html -t html -f "%l\n"
```

In diesem Beispiel wird ein *HTML*-Dokument übergeben und entsprechend der Eingabe-Typ gesetzt. Als Ergebnis wird nur die ermittelte Sprache ausgegeben und ein Zeilenumbruch ergänzt. Das Ergebnis könnte also wie folgt aussehen:

```
Danish
```

Insgesamt stehen vier Werte zur Verfügung, die als Ergebnis zurückgegeben werden können: die ermittelte Sprache, ihr ISO 639-3 Code, die ermittelte Zeichenkodierung und die angegebene Zeichenkodierung. Die angegebene Zeichenkodierung ist nicht für das reine Textformat verfügbar. Sie gibt zurück, was ggf. im Dokument als Zeichenkodierung deklariert wurde, z.B. bei *HTML* über ein Meta-Tag. Die Auswertung der Zeichenkodierung wird noch einmal in Kapitel 5.4 aufgegriffen.

Ergebnis	Beschreibung	Beispiel
language	Name der Sprache (Englisch)	„German“
isocode	Code der Sprache gemäß ISO 639-3	„deu“
identified encoding	Name der ermittelten Zeichenkodierung	„UTF-8“
declared encoding	Name der angegebenen Zeichenkodierung	„utf-8“

Abbildung 1: Ergebnisse

Die Ausgabe kann mit einem Formatstring angepasst werden. Welche Parameter dazu bereit gestellt werden, wird im Kapitel „Die Ausgabe formatieren“ (5.3, Seite 13) beschrieben.

Um eine sichere Bestimmung durchzuführen, muss die Eingabe mindestens einen Umfang von etwa 25 Zeichen haben und einen gewissen Grad an Variation aufweisen. Eine Zeichenkette, die beispielsweise lediglich das Wort „Haus“ beinhaltet, ist ebenso keine valide Eingabe wie eine zehnmahlige Wiederholung des Worts, obwohl damit zumindest die Mindestlänge erfüllt wäre.

Bei E-Mails nach dem *MIME*-Standard, die aus mehreren Teilen bestehen, wird nur der erste (Text-) Teil ausgewertet.

5.1 Kurzreferenz

Option	Parameter	Bedeutung
-i	Pfad zur Eingabe-Datei	Eingabe-Dokument
-t	Typ [txt html xml email]	Typ der Eingabe
-f	Formatstring	Formatierung der Ausgabe
	%l	ermittelte Sprache
	%i	ISO 639-3 Code der Sprache
	%e	ermittelte Zeichenkodierung
	%d	angegebene Zeichenkodierung
	%f	Eingabe-Pfad
	\n, \r, \t, \a	Escape-Sequenzen
-v	-	<i>lidc</i> Version
-h	-	Hilfetext

5.2 Optionen

-i PATH

Mit dieser Option wird der Pfad zur Eingabe-Datei gesetzt. Wird diese Option ausgelassen oder explizit auf „-“ gesetzt, wird die Standardeingabe (`stdin`) verwendet.

-t TYPE

Hiermit wird der Typ der Eingabe festgelegt. Mögliche Parameter der Option sind entsprechend der unterstützten Typen (siehe Kapitel 2):

- `txt`
reine Textdokumente ohne Markup
- `html`
HTML Dokumente (z.B. X-HTML, HTML 4.0)
- `xml`
XML Dokumente
- `email`
E-Mails im RFC 822 oder MIME 1.0 Format

Wird der Eingabe-Typ nicht gesetzt, wird `txt` als Standardwert angenommen.

Erfolgt der Aufruf mit einer Datei und wurde kein Typ gesetzt, kann der Typ eventuell anhand der Datei-Endung automatisch bestimmt werden. Die üblichen Datei-Endungen (`.txt`, `.html`, `.xml`, `.eml`) werden erkannt sowie alle *Maildir* Endungen und Erweiterungen des *Dovecot* IMAP Servers.

-f FMT_STR

Diese Option setzt die gewünschte Ausgabe als Formatstring. Die möglichen Parameter werden im nächsten Kapitel ausführlich beschrieben. Die Ausgabe erfolgt auf der Standardausgabe (`stdout`).

Wird diese Option nicht gesetzt, wird der folgende Formatstring verwendet:

"%l,%i,%e\n" (Sprache, ISO 639-3 Code, ermittelte Zeichenkodierung).

-v

Hiermit kann die Version des verwendeten Programms und der zugrundeliegenden Bibliothek angezeigt werden. Es findet keine weitere Verarbeitung statt.

-h

Diese Option zeigt einen kurzen Hilfetext zum Aufruf von *lidc* an. Es findet keine weitere Verarbeitung statt.

5.3 Die Ausgabe formatieren

Mit Hilfe eines Formatstring kann die Ausgabe der Ergebnisse angepasst werden. Es kann festgelegt werden, welche Ergebnisse angezeigt werden sollen (z.B. nur die Zeichenkodierung). Es kann aber zusätzlich auch Text eingefügt werden. Somit ist es auch möglich, eine direkte Ausgabe im *CSV* oder *XML* Format zu erstellen und diese ggf. in eine Datei umzulenken.

Es stehen Platzhalter für die jeweiligen Ergebnisse zur Verfügung, die bei der Ausgabe entsprechend ersetzt werden.

Die alleinige Ausgabe des Pfades der Eingabe-Datei ist nicht vorgesehen, da es sich bei dieser Angabe nicht um ein ermitteltes Ergebnis handelt.

%l

Der Platzhalter %l wird durch die ermittelte Sprache in ihrer englischen Bezeichnung ersetzt, z.B. „German“, „French“ oder „Swedish“.

%i

Dieser Platzhalter steht für den dreistelligen Code der Sprache nach ISO 639-3, z.B. „deu“, „fra“ oder „swe“.

%e

Dieser Parameter steht für die ermittelte Zeichenkodierung, z.B. „ISO-8859-1“, „Windows-1252“ oder „UTF-8“.

%d

Der Parameter %d bezeichnet die angegebene Zeichenkodierung und wird durch ihren Namen in Kleinbuchstaben ersetzt, z.B. „iso-8859-1“, „windows-1252“ oder „utf-8“.

Nicht alle unterstützten Eingabe-Typen haben eine angegebene Zeichenkodierung. In reinen Textdokumenten ist keine Kodierung angegeben. Auch in den anderen Formaten ist eine Angabe zur Kodierung optional. Falls keine Zeichenkodierung angegeben wird oder ermittelt werden kann, wird statt dessen „none“ ausgegeben.

%f

Hiermit kann zusätzlich der Pfad der Eingabe-Datei bzw. „stdin“ ausgegeben werden.

Escape Sequenzen

Zusätzlich werden die Escape-Sequenzen „\n“, „\t“, „\r“ und „\a“ unterstützt, um die Ausgabe optisch aufzubereiten.

5.3.1 Beispiele

Eine einfache Formatierung könnte folgendermaßen aussehen:

```
$ lidc -i myfile.xml -f "%f: %l, %e\n"
myfile.xml: Bulgarian, UTF-8
```

Wenn die Ergebnisse in ein XML Format überführt werden sollen, ist das wie folgt möglich:

```
$ lidc -i german.eml -f \
"<email>\n\t<lang>%l</lang>\n\t<enc>%e</enc>\n</email>\n"
<email>
  <lang>German</lang>
  <enc>ISO-8859-1</enc>
</email>
```

Das folgende Beispiel zeigt eine nützliche Kommandoverkettung, bei der mittels `pdftotext(1)` (Bestandteil von `xpdf`) zunächst der Inhalt einer PDF Datei in ein markupfreies Textformat überführt und auf `stdout` ausgegeben wird, so dass es anschließend durch Verkettung direkt auf `stdin` als Eingabe für `lidc(1)` dienen kann:

```
$ pdftotext manual_deu.pdf - | lidc -f "%l, %e, %d\n"
German, ISO-8859-1, none
```

5.4 Die Ergebnisse zur Zeichenkodierung auswerten

5.4.1 Erkannte Zeichenkodierung

Die Bedeutung der erkannten Zeichenkodierung (Platzhalter %e) unterscheidet sich abhängig vom Eingabe-Typen und muss in Folge dessen auch abhängig von diesem ausgewertet werden.

Bei reinen Text-, *HTML*- und *XML*-Dokumenten ist die erkannte Kodierung stets als *Zeichenkodierung der Datei* zu verstehen.

Bei E-Mails kann dies ebenfalls zutreffend sein, nämlich genau dann, wenn die Nachricht in einer 7- oder 8-Bit Transfer-Encoding vorliegt. Liegt die Nachricht allerdings in *Quoted-Printable* oder *Base64* kodierter Form vor (*MIME*), so muss *lidc* diese Nachricht zunächst in ihre ursprüngliche Form überführen, um Sprache und Zeichenkodierung ermitteln zu können. Daher sollte bei E-Mails die Bedeutung der erkannten Zeichenkodierung generell als *Zeichenkodierung der dekodierten Nachricht* verstanden werden, auch, wenn diese mit der Dateikodierung übereinstimmen kann, wie beispielsweise bei RFC 822 E-Mails.

Typ	Erkannte Zeichenkodierung bezieht sich auf...
txt	Datei
html	Datei
xml	Datei
email	Erste dekodierte Nachricht

Abbildung 2: Interpretation der erkannten Zeichenkodierung

5.4.2 Angegebene Zeichenkodierung

Die angegebene Zeichenkodierung kann nur für Typen ermittelt werden, die eine derartige Spezifikation erlauben. Dies ist bei den Typen `html`, `xml` und `email` (*MIME*) der Fall. Die Spezifikation der Zeichenkodierung ist jedoch bei allen genannten Typen optional.

Die angegebene Zeichenkodierung wird von *lidc* stets in Kleinbuchstaben zurückgegeben und darüber hinaus nicht weiter verarbeitet.

lidc kann auch verwendet werden, um falsche Zeichenkodierungsdeklarationen ausfindig zu machen, wie sie beispielsweise bei *HTML*-Dokumenten immer noch recht häufig vorkommen. Zu diesem Zweck bietet es sich an, die erkannte (%e) mit der angegebenen Zeichenkodierung (%d) zu vergleichen. Dabei muss jedoch beachtet werden, dass nicht jede Abweichung der beiden Werte auf eine fehlerhafte Deklaration hinweisen muss. Dies ist insbesondere dann der Fall, wenn die angegebene Zeichenkodierung eine Obermenge der erkannten Zeichenkodierung ist, wie es beispielsweise bei UTF-8 und ASCII oder ISO-8859-1 und ASCII der Fall ist.

6 Fehler vermeiden

Bei einem Fehler, der eine Weiterverarbeitung unmöglich macht, wird das Programm beendet und eine englischsprachige Fehlermeldung ausgegeben, die die Ursache des Fehlers näher beschreibt. Im Folgenden werden einige Hinweise gegeben, wie die wahrscheinlichsten Fehler zu vermeiden oder zu interpretieren sind.

Um eine Eingabe sicher verarbeiten zu können, ist eine Mindestlänge des Textes erforderlich. Anhaltspunkte für eine ausreichende Länge sind

- ⇒ der Text weist mehr als 25 Zeichen auf und
- ⇒ es kommen mehr als zwei Wörter vor und
- ⇒ die vorkommenden Wörter sind voneinander verschieden.

Des Weiteren muss natürlich überhaupt Text vorliegen. Das ist nicht immer sofort ersichtlich, zum Beispiel, wenn E-Mails oder *XML*-Dokumente verarbeitet werden.

Bei E-Mails liegt keine Eingabe vor, die verarbeitet werden kann, wenn

- ⇒ kein Textinhalt enthalten ist oder
- ⇒ sie in UTF-16 oder UTF-32 kodiert ist oder
- ⇒ kein Teil in einem unterstützten Format vorliegt.

Ebenso bei *XML*-Dokumenten, die eventuell nur aus Tags bestehen und keinen eigentlichen Text-Inhalt enthalten.

Ist der Text zu kurz oder kann kein Text extrahiert werden, kommt es, abhängig von weiteren Faktoren, zu einer der beiden Fehlermeldungen:

```
lidc: Insufficient input length
lidc: No text extracted
```

Sollte zum Beispiel versehentlich ein Bild oder eine Anwendung als Eingabe-Datei verwendet werden, wird folgender Fehler aufgeworfen:

```
lidc: Binary Data
```

Beim Aufruf des Programms kann jede Option nur einmal definiert werden. Sollte eine Option mehrmals angegeben werden, erscheint die Fehlermeldung:

```
lidc: [Parameter] redefined
```

Es können nur die unterstützten Eingabe-Typen (*txt*, *html*, *xml* und *email*) verarbeitet werden. Beim Aufruf eines nicht unterstützten Eingabe-Typs wird ebenfalls ein Fehler aufgeworfen.

```
lidc: Unsupported type
```

Weitere Fehler werden hauptsächlich ausgelöst, wenn die Eingabe in irgendeiner Weise defekt ist, z.B. wenn eine E-Mail fehlerhaft kodiert wurde oder eine fehlerhafte UTF-16 oder UTF-32 Kodierung vorliegt. Die Fehlermeldungen geben Hinweise zur Ursache des Fehlers. In den meisten Fällen lassen sich diese Fehler nicht abfangen.

Kann die Eingabe trotz eines Fehlers verarbeitet werden, wird lediglich eine entsprechende Warnung zu Informationszwecken auf `stderr` ausgegeben und mit der Verarbeitung fortgefahren, so dass ein Ergebnis zurückgeliefert werden kann.

A Referenzen

- ISO 639-3 Standard, <http://www.sil.org/iso639-3/>
- RFC 822, „Standard for the Format of ARPA Internet Text Messages“
- RFC 2045, 2046, 2047, 2049: „Multipurpose Internet Mail Extensions“ (MIME)
- 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“
- Maildir Standard, <http://cr.yp.to/proto/maildir.html>
- Dovecot IMAP Server, <http://www.dovecot.org/>
- Lingua-Systems' *lidc* Produktwebseite, <http://www.lingua-systems.de/language-identifier/lidc-application/>

Index

A	
Abhängigkeiten	8
Aufruf	11
B	
Beispiel	11, 14
Aufruf	11
Formatstring	14
Formatstring (XML)	14
PDF	14
D	
Deinstallation der Software	10
Debian GNU/Linux	10
FreeBSD	10
Solaris	10
dpkg (Linux)	8, 10
E	
Eingabe-Typen	5, 12, 15
email	5, 12, 15
html	5, 12, 15
txt	5, 12, 15
xml	5, 12, 15
Ergebnis	11, 13, 15
angegebene Zeichenkodierung	11, 15
ermittelte Sprache	11
ermittelte Zeichenkodierung	11, 15
formatieren	13
ISO 639-3 Code	11
F	
Fehlerhafte Zeichenkodierungsdeklaration	15
Fehlermeldungen	16
Formatstring	13
I	
Installation der Software	8
Debian GNU/Linux	8
FreeBSD	9
Solaris	9
Installationsumfang	8
M	
Mindestlänge	11, 16
O	
Optionen	12
-f FMT_STR	12
-h	13
-i PATH	12
-t TYPE	12
-v	13
P	
Parameter	<i>siehe</i> Optionen
pkg_add (FreeBSD)	9
pkg_delete (FreeBSD)	10
pkgadd (Solaris)	9
pkgrm (Solaris)	10
R	
Rückgabe	<i>siehe</i> Ergebnis
S	
Sprachen	6
T	
Transliteration	7
V	
Voraussetzungen	8
W	
Warnung	16
Z	
Zeichenkodierung	15
angegebene	15
auswerten	15
erkannte	15
unterstützte	6