

HANDBUCH FÜR

MODUL ZUM IMPORT VERSCHIEDENER DATEIFORMATE

Mark Unger und Siegfried Kienzle

24. November 2016

Erklärung

Die in diesem Projekt verwendete Software unterliegt den rechtlich jeweiligen Bestimmungen der einzelnen Organisationen und Firmen.

Inhaltsverzeichnis

1	Modul	4
1.1	Über die Software	4
1.2	Über das Handbuch	4
2	Grundlagen	5
2.1	Installation	5
2.2	Bestandteile Installationspaket	9
2.3	Modulbestandteile	9
2.4	Erste Schritte	10
2.4.1	Genereller Aufruf	10
2.4.2	Extrahieren von Text auf Konsole	11
2.4.3	Extrahieren von Text in eine Datei ohne Konsolenausgabe	12
2.4.4	Extrahieren von Text in eine Datei mit Konsolenausgabe	13
2.4.5	Hilfe aufrufen	14
3	Technischer Hintergrund	15
3.1	Verwendete Fremdsoftware	15
3.2	Aufbau	15
3.2.1	convertToTxt.py	15
3.2.2	extractTxt.py	15
3.2.3	Die Module	15
4	Kontaktdaten	17

1 Modul

1.1 Über die Software

Dieses Modul dient zur Extrahierung von Text aus Dateien. Sie können dieses Modul für folgende Endungen verwenden:

- .doc
- .docx
- .odt
- .pdf
- rtf

Es wurde für Python 3.4.3 entwickelt und unter Ubuntu 14.04.05 LTS getestet. Zur Installation liegt ein Bash-Script vor.

1.2 Über das Handbuch

Dieses Handbuch beschreibt die Installation und die Handhabung mit dem Modul.

2 Grundlagen

2.1 Installation

1. Installationsscript mittels `./inst.sh` aufrufen:

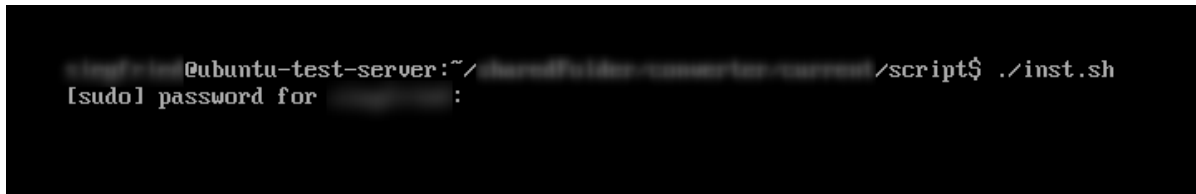


Abbildung 1: Nach Aufruf des Installationscriptes `./inst.sh`

2. sudo-Passwort eintippen und die Enter-Taste drücken.
3. Es werden nun einige Abhängigkeiten installiert, die zur Ausführung dieses Moduls benötigt werden.

4. Geben Sie nun den Pfad an, in den das Modul installiert werden soll. Sollte der Pfad nicht existieren, werden Sie wie in Abbildung 4 gefragt ob der Pfad erstellt werden soll. Existiert der Pfad, entfallen die Schritte 6 bis 8.



Abbildung 2: Nach Aufruf des Installationscriptes `./inst.sh`



Abbildung 3: Nach Eingabe des Installationspfads

5. Wenn der OK-Button blau hinterlegt ist, können Sie mit der Enter-Taste den Pfad bestätigen.

6. Sollte kein Pfad existieren, erscheint folgendes Fenster:



Abbildung 4: Pfad erstellen?

7. Wählen Sie nun mit den Pfeiltasten aus, ob Sie den Pfad erstellen möchten oder nicht und drücken Sie dann die Enter-Taste.



Abbildung 5: Pfad wurde erstellt

8. Es wurde nun der Pfad erstellt. Drücken Sie nun die Enter-Taste, um die Dateien in das entsprechend vorher erstellte Verzeichnis, zu entpacken.



Abbildung 6: Pfad wurde erstellt

9. Die Installation ist nun abgeschlossen. Prüfen Sie nun bitte ob alle Dateien installiert wurden. Eine genaue Auflistung finden Sie unter dem Punkt 2.3.

2.2 Bestandteile Installationspaket

Datei	Beschreibung
inst.sh	Bash-Script für die Ausführung als Super-User (sudo) unter Ubuntu
ubuntu.sh	Bash-Script für die Installation unter Ubuntu
moduls.tar	Tar-Datei, die die Python-Module enthält

Tabelle 1: Bestandteile Installationspaket

2.3 Modulbestandteile

Datei	Verwendung
convertToTxt.py	Datei die für das Extrahieren aufgerufen wird
extractTxt.py	Hauptdatei für die Extrahierung
docTxt.py	Modul für die Dateieindung doc
docxTxt.py	Modul für die Dateieindung docx
odtTxt.py	Modul für die Dateieindung odt
pdfTxt.py	Modul für die Dateieindung pdf
rtfTxt.py	Modul für die Dateieindung rtf

Tabelle 2: Modulbestandteile

2.4 Erste Schritte

2.4.1 Genereller Aufruf

Im Allgemeinen wird das Modul wie folgt aufgerufen:

```
python3 convertToTxt.py <PARAMETER> (<PFAD>)
```

Dabei sollte <PARAMETER> durch Parameter aus Tabelle 3 ersetzt werden. Bei den Parametern -p bzw. --process sowie bei -o bzw. --output sollte dann zusätzlich <PFAD> (das hier in runden Klammern steht) durch den Pfad der Datei, aus der der Text extrahiert werden soll, ersetzt werden. Außerdem ist zwingend darauf zu achten, dass das Modul mit python3 aufgerufen wird.

Parameter (kurz)	Parameter (lang)	Erklärung
-h	--help	Zeigt die Hilfe an
-p	--process	Führt die Textextrahierung durch.
-v	—————	Verbose-Mode: Gibt den Text auf Konsole aus.
-o	--output	Parameter für die Ausgabedatei. Nur anwendbar mit Argument -p bzw. --process

Tabelle 3: Parameterübersicht

Beispielaufrufe finden Sie weiter unten in diesem Kapitel.

2.4.2 Extrahieren von Text auf Konsole

Zum Extrahieren von Text tippen Sie einfach

```
python3 convertToTxt.py -p <DATEIPFAD> -v
```

oder

```
python3 convertToTxt.py --process <DATEIPFAD> -v
```

Als Beispiel sehen Sie im folgenden wie Text aus einer DOCX-Datei extrahiert wird:

```
python3 convertToTxt.py --process endungen/docx.docx -v_
```

oder

```
python3 convertToTxt.py -p endungen/docx.docx -v_
```

```
Guten Tag,
```

```
Hallo
```

Abbildung 7: Beispielausgabe von docx

2.4.3 Extrahieren von Text in eine Datei ohne Konsolenausgabe

Zum Extrahieren von Text in eine Datei, ohne dabei den Text auf die Konsole auszugeben, tippen Sie einfach

```
python3 convertToTxt.py -p <DATEIPFAD> -o <AUSGABEDATEI>
```

oder

```
python3 convertToTxt.py --process <DATEIPFAD> --output <AUSGABEDATEI>
```

Es ist zu beachten, dass wenn die angegebene Ausgabedatei bereits existiert, der Inhalt durch den Text der im Moment extrahiert wird, überschrieben wird. Als Beispiel sehen Sie im folgenden wie Text aus einer DOCX-Datei extrahiert wird:

```
python3 convertToTxt.py -p endungen/docx.docx -o docx.txt_
```

oder

```
python3 convertToTxt.py --process endungen/docx.docx --output docx.txt
```

```
docx.txt
```

Abbildung 8: Beispielausgabe des Befehls ls, nachdem die Datei erstellt wurde

2.4.4 Extrahieren von Text in eine Datei mit Konsolenausgabe

Zum Extrahieren von Text in eine Datei mit Konsolenausgabe, tippen Sie einfach

```
python3 convertToTxt.py -p <DATEIPFAD> -o <AUSGABEDATEI> -v
```

oder

```
python3 convertToTxt.py --process <DATEIPFAD> --output <AUSGABEDATEI> -v
```

Es ist zu beachten, dass wenn die angegebene Ausgabedatei bereits existiert, der Inhalt durch den Text der im Moment extrahiert wird, überschrieben wird. Als Beispiel sehen Sie im folgenden wie Text aus einer DOCX-Datei extrahiert wird:

```
python3 convertToTxt.py -p endungen/docx.docx -o docx.txt -v
```

oder

```
python3 convertToTxt.py --process endungen/docx.docx --output docx.txt -v
```

```
Guten Tag,  
  
Hallo
```

Abbildung 9: Ausgabe nach Aufruf des obigen Befehls

2.4.5 Hilfe aufrufen

Zum Aufrufen der Hilfe einfach wie im folgenden Bild den Parameter -h bzw. --help eingeben.

```
python3 convertToTxt.py -h_
```

oder

```
python3 convertToTxt.py --help
```

Die Ausgabe sollte wie folgt aussehen:

```
arguments:
-h,                --help                show help message and exit
-p [path to file]  --process [path to file] to run the program
-o [path to output-file] --output [path to output-file] to extract text into file
-v                (works only with the argument -p)
                  verbose-Mode
```

3 Technischer Hintergrund

3.1 Verwendete Fremdsoftware

Datiename	verwendete Zusatzsoftware	Entwicklerwebseite
docTxt.py	catdoc	http://freecode.com/projects/catdoc
docxTxt.py	Python-Modul docx2txt	http://docx2txt.sourceforge.net/
pdfTxt.py	pdftotext	https://poppler.freedesktop.org/
odtTxt.py	odt2txt	https://github.com/dstosberg/odt2txt
rtfTxt.py	unrtf	https://www.gnu.org/software/unrtf/unrtf.html

Tabelle 4: Auflistung der verwendeten Software

3.2 Aufbau

Wie schon aus der Tabelle in Kapitel 2.3 zu sehen ist, besteht das Projekt aus einer Hauptdatei `convertToTxt.py` die aufgerufen wird, einer Hilfsdatei `extractTxt.py` in die die Programmlogik ausgelagert wurde und den einzelnen Modulen. Im weiteren werden die einzelnen Dateien technisch erläutert.

3.2.1 `convertToTxt.py`

Die `convertToTxt.py` nimmt alle Anfragen entgegen und beinhaltet die einzelnen Argumente sowie die Hilfe-Funktion. Die Verarbeitung der Argumente und Optionen wurden mittels dem Modul `getopt` realisiert.

3.2.2 `extractTxt.py`

Die Datei `extractTxt.py` enthält die eigentliche Logik des Extrahierungs-Skriptes. Sie enthält die Funktionen `process(path)` und `file(text, a)`. Die `process(path)`-Funktion nimmt den Pfad aus der zu extrahierenden Datei über den Parameter `path` entgegen und übergibt den Pfad dem entsprechenden Modul, indem es sich die Dateieindung betrachtet. Der heraus zu extrahierende Text, der von einzelnen Modulen zurückgegeben wird, wird mit UTF-8 dekodiert und so dann endgültig zurückgegeben. `file(text, a)` speichert den heraus extrahierten Text in eine Datei. Dazu wird dem Parameter `text` der zu speichernde Text und dem Parameter `a` der Speicherort übergeben.

3.2.3 Die Module

Wie im Abschnitt 3.1 in der Tabelle zu sehen ist, gibt es fünf Module. Alle Module rufen, bis auf `docxTxt.py`, ein Konsolen-Programm mittels `subprocess.Popen()` auf. Es wird dazu das Modul `subprocess` importiert. In einer Liste, die dem `subprocess.Popen()` übergeben wird, steht das zu ausführende Programm und der Dateiname. Außerdem wird die Standardausgabe und die Ausgabe für die Fehler in eine Pipe umgeleitet, damit der zu extrahierte Text später weiterverarbeitet werden kann. Am Ende wird dann

`process.stdout.read()` zurückgeliefert. Die Fehlerbehandlung wurde mittels `try-except`-Anweisung realisiert und es wird bei einer auftretenden `Exception` ins Logfile geschrieben.

4 Kontaktdaten

Name	E-Mail
Mark Unger	mrk.unger@gmail.com
Siegfried Kienzle	siegfried.kienzle@gmx.de