

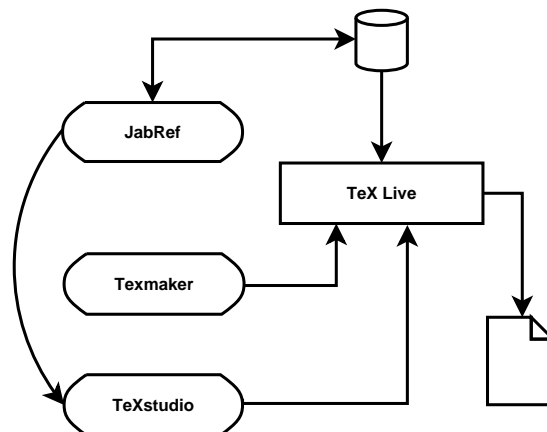
---

# Installation / Konfiguration von T<sub>E</sub>XLive, Texmaker, TeXstudio und JabRef

Eine Anleitung mit manuellen Erweiterungen  
am Beispiel der Distribution Kubuntu 12.04 LTS

---

Günther Zander  
8. Oktober 2013



Lizenz: CC BY-SA

<http://www.lug-balista.de>

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Installation der Programme</b>	<b>3</b>
2.1	Vorarbeiten für die Installation . . . . .	3
2.2	TeX Live . . . . .	3
2.3	Texmaker . . . . .	6
2.4	TeXstudio . . . . .	6
2.5	JabRef . . . . .	7
<b>3</b>	<b>Konfiguration Texmaker</b>	<b>8</b>
3.1	Grundkonfiguration . . . . .	8
3.2	Konfiguration Texmaker mit einem Wörterbuch . . . . .	10
3.3	Das Wörterbuch aktualisieren . . . . .	12
3.4	Texmaker mit einer Datei starten . . . . .	12
3.5	Eigene Makros in Texmaker erstellen . . . . .	13
3.6	Ein Compiler für Texmaker . . . . .	14
3.7	Persönlichen Einstellungen sichern . . . . .	14
<b>4</b>	<b>Konfiguration TeXstudio</b>	<b>15</b>
4.1	Grundkonfiguration . . . . .	15
4.2	Farbeinstellungen . . . . .	15
4.3	Tooltips deaktivieren . . . . .	16
4.4	TeXstudio mit einer Datei starten . . . . .	16
4.5	Das Wörterbuch aktualisieren . . . . .	16
4.6	Eigene Makros in TeXstudio . . . . .	18
4.7	Ein Compiler für TeXstudio . . . . .	18
4.8	Persönlichen Einstellungen sichern . . . . .	19
4.9	Weitere Tipps zu TeXstudio . . . . .	20
4.9.1	Syntaxhervorhebung verhindern . . . . .	20
4.9.2	Eigene Vorlagen erstellen . . . . .	20
4.9.3	Verzeichnisse entrümpeln . . . . .	21
<b>5</b>	<b>Arbeiten mit Makefiles</b>	<b>22</b>
<b>6</b>	<b>JabRef</b>	<b>24</b>
6.1	Benutzerdefinierte Felder . . . . .	24
6.2	Datenaustausch JabRef $\Rightarrow$ Texmaker . . . . .	25
6.3	Datenaustausch JabRef $\Rightarrow$ TeXstudio . . . . .	25
6.3.1	Zitierweise einstellen . . . . .	25
6.4	Schlüsselbereich . . . . .	25
<b>7</b>	<b>Listings</b>	<b>26</b>
7.1	Compiler . . . . .	26
7.2	texmaker.backup . . . . .	30
7.3	texstudio.backup . . . . .	31

# 1 Einleitung

Die \*buntu-Versionen verfügen naturgemäß nicht über die aktuellen Programmversionen von  $\text{T}_{\text{E}}\text{X}$  bzw. von  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ , weswegen Sie diese neu installieren sollten. Damit Sie sich die aktuellsten Programme installieren können, wird Ihnen aufgezeigt, wo sich die Quellen dieser Programme befinden. Dieses Dokument bezieht sich auf folgende Versionen der folgenden Programme:

- TeX Live            2012
- Texmaker           4.0.1
- TeXstudio          2.5.2
- JabRef              2.9.2

Zuerst wird Ihnen aufgezeigt, mit welchen Möglichkeiten Sie die entsprechenden Programme installieren und auf dem neusten Stand halten können. Daran schließen sich vier Kapitel an, in denen auf die Einstellmöglichkeiten der Programme sowie auf einige Arbeitserleichterungen eingegangen wird.

Speziell bei den Programmen Texmaker und TeXstudio ist es ein Ärgernis, dass es nicht möglich ist das verwendete Wörterbuch zu erweitern. Hier wird Ihnen gezeigt, wie es doch möglich ist.

Alle hier vorgestellten Programme wurden ausgewählt, weil es Open-Source-Produkte sind und zudem auf allen Betriebssystemen laufen. Damit müssen Sie sich nur einmal in die Programme einarbeiten, denn das Betriebssystem braucht Sie nicht zu kümmern.

Für die hier beschriebenen Vorgehensweisen und Skripte werden keinerlei Garantien für die Funktionsfähigkeit gegeben. Bevor Sie etwas Testen ist eine Datensicherung unentbehrlich um einen Datenverlust vorzubeugen.
---

## Konventionen in diesem Artikel

In diesem Artikel werden folgende typographische Konventionen verwendet:

- Alle Dateinamen, Links und Verzeichnispfade werden serifenlos dargestellt, wie `/usr/local`.
- Alle Menüs, Menüpfade und Menüeingaben werden fett dargestellt. Die Pfade werden dabei mit einem Punkt getrennt, wie **Datei • Option • Weiteres**.
- Sourcecode sowie Ein- und Ausgaben in einer Shell werden in einer nichtproportionalen Schrift dargestellt, wie `svn checkout`.
- Alle anderen hervorzuhebenden Wörter werden kursiv dargestellt, wie *import*.
- Die Internetlinks mit der Schriftfarbe blau sind im elektronischen Dokument aktiv und können angeklickt werden.

**Über den Autor:** Mit der Version 6.1 von SuSE (1998) ist der Autor auf Linux endgültig umgestiegen. Einigen Jahre arbeitete er mit der Distribution Debian bis er mit der Version 9.04 auf Kubuntu wechselte. Nach einem Studium der Elektrotechnik und den ersten Jahren als Programmierer, ist er heute immer noch im IT-Sektor tätig und arbeitet seit einigen Jahren mit der  $\text{T}_{\text{E}}\text{X}$ -Umgebung zur Erstellung der unterschiedlichsten Dokumente.

## 2 Installation der Programme

### 2.1 Vorarbeiten für die Installation

Stellen Sie zuerst sicher, dass die Pakete *texmaker* und *texlive* nicht auf ihrem Rechner installiert sind. Falls Sie das Paket *jabref* bereits installiert haben, so brauchen Sie dieses nicht zu deinstallieren.

### 2.2 TeX Live

Um die  $\text{T}_{\text{E}}\text{X}$ -Programm, die unter dem Namen "TeX Live" vertrieben werden, aktuell zu halten, laden Sie von der Seite [www.tug.org/texlive/acquire-netinstall.html](http://www.tug.org/texlive/acquire-netinstall.html) die Datei `install-tl-unx.tar.gz` herunter und entpacken diese. Zusätzlich muss das Paket *perl-tk* aus den Paketquellen installiert werden. Wechseln Sie in das Verzeichnis der entpackten Dateien und starten Sie dort, mit Rootrechten das Programm mit den zusätzlichen Parametern, um eine grafische Oberfläche zu erhalten.

```
sudo ./install-tl -gui perlTk
```

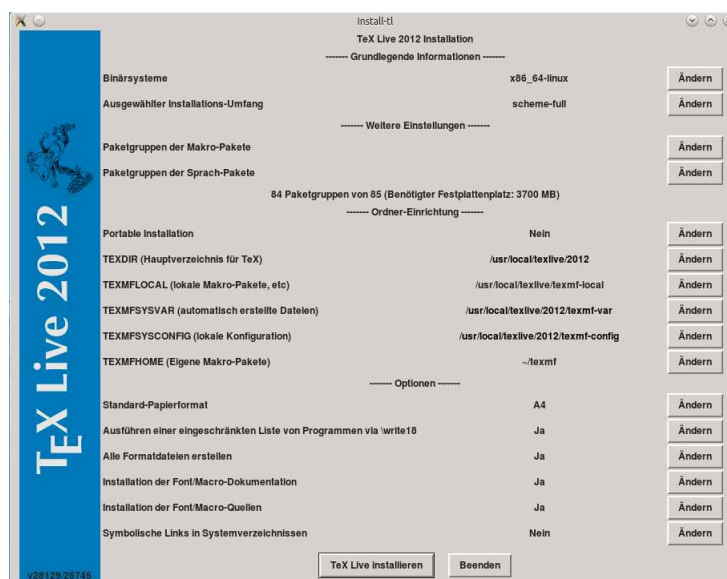


Abbildung 2.1: Die grafische Oberfläche von TeX Live

Nun können Sie alle Änderungen vornehmen, die Sie möchten. Um anschließend mit den Programmen arbeiten zu können, ist es ratsam den letzten Eintrag mit der Bezeichnung **Symbolische Links in Systemverzeichnis** auf den Wert **Ja** abzuändern. Damit werden alle notwendigen Programme und Links im Verzeichnis `/usr/local/bin` angelegt (insgesamt 370 Einträge). Dazu wird eine zweite Abfrage geöffnet, in der Sie nur das obere Feld aktivieren müssen.

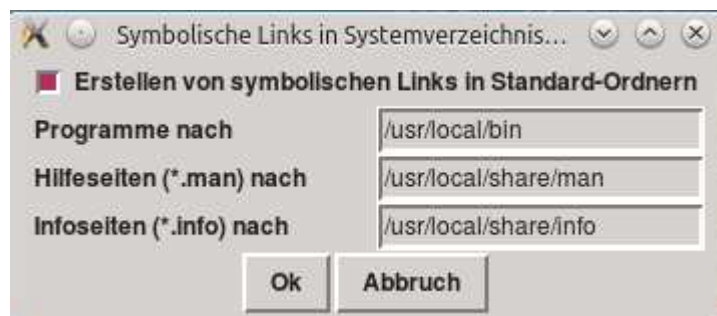


Abbildung 2.2: Die Abfrage der Systempfade

Wenn Sie nun des öfteren im Verzeichnis `/usr/local/bin` arbeiten, können die 370 erzeugten Einträge doch ein wenig störend sein. Abhilfe können Sie schaffen, indem Sie in dem Verzeichnis ein Unterverzeichnis `texlive` anlegen bevor Sie die Installation starten und dieses Verzeichnis in die grafische Oberfläche eintragen.

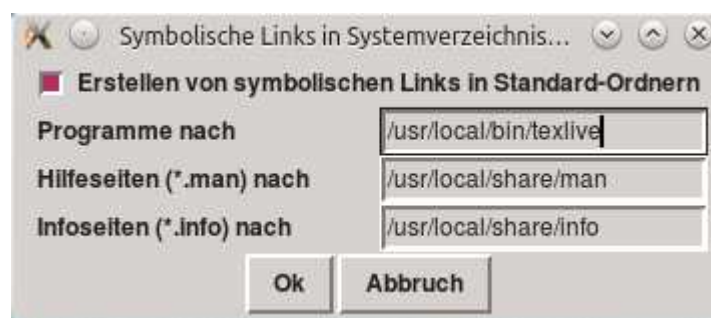


Abbildung 2.3: Die Abfrage der geänderten Systempfade

Dieses neue Verzeichnis müssen Sie noch ihrem System mitteilen, damit die dortigen Einträge mit durchsucht werden können. Dazu öffnen Sie einen Editor mit Rootrechten und erweitern die Datei `/etc/environment` um den Eintrag: `:/usr/local/bin/texlive`. Anschließend melden Sie sich einmal ab und wieder an und von nun an steht Ihnen dieses Verzeichnis zur Verfügung.

Nachdem Sie ihre Einstellungen vorgenommen haben, starten Sie die Installation mit einem Mausklick auf das Feld **TeX Live installieren**. Die Installation benötigt ca. 1 Stunde und belegt etwa 4 GB auf der Platte. Um diese Installation immer auf dem neuesten Stand zu halten, sollten Sie von Zeit zu Zeit folgenden Befehl ausführen:

```
sudo tlmgr update -all
```

Wenn Sie das Verzeichnis `/usr/local/bin/texlive` angelegt haben und den Path für den Benutzer `root` nicht ändern wollen, müssen Sie den Aufruf entsprechend anpassen:

```
sudo /usr/local/bin/texlive/tlmgr update -all
```

Oder Sie setzen einen Link, dann können Sie wie gewohnt damit arbeiten.

```
ln /usr/local/bin/texlive/tlmgr /usr/local/bin/tlmgr
```

**Installationsabbruch:** Sollten Sie die Installation von TeX Live einmal abgebrochen haben, können Sie das Installationsprogramm jederzeit wieder starten. Tex Live lädt dann nur noch die Pakete nach, die noch nicht vorhanden sind, und setzt damit die Installation fort.

**Deinstallation von TeX Live:** Mitunter kann es vorkommen, dass Sie das gesamte Paket löschen möchten. Dafür gibt es kein Deinstallationsprogramm, dieses müssen Sie händisch durchführen. Dazu gehen Sie, mit Rootrechten, wie folgt vor:

- /usr/local/texlive            Verzeichnis löschen
- /usr/local/bin                Alle angelegten Einträge für TeX Live löschen
- /usr/local/bin/texlive        Falls benutzt das gesamte Verzeichnis löschen
- ~/texmf                        Falls vorhanden, löschen

## 2.3 Texmaker

Die aktuelle Version von Texmaker laden Sie unter [www.xmlmath.net/texmaker](http://www.xmlmath.net/texmaker) herunter. Zur Zeit ist dieses die Version 4.0.1 (Paketquelle 3.2.1). Nun wechseln Sie mit einem Dateimanager in ihr Downloadverzeichnis und starten die Installation, indem Sie die Datei `texmaker_ubuntu_12.04_{Versionsnummer}_{i386|amd64}.deb` mit der Maus anklicken. Anschließend wird das Rootpasswort einmal abgefragt, bevor die Installation beginnt.

Nach der Installation wechseln Sie in den Paketmanager und markieren das Programm *texmaker* mit der Option **derzeitige Version behalten**, um Updates aus den Paketquellen auszuschließen.

**Update einspielen für Texmaker:** Um eine neue Version von Texmaker einzuspielen, laden Sie die neue Datei von [www.xmlmath.net/texmaker](http://www.xmlmath.net/texmaker) herunter und starten die Installation, indem Sie mit einem Dateimanager die Datei mit der Maus anklicken. Ihre Einstellungen bleiben dabei erhalten.

## 2.4 TeXstudio

Bevor Sie TeXstudio installieren, sollten Sie das Programm Texmaker deinstallieren und das Verzeichnis `~/.config/xm1` löschen. Wenn Sie dieses nicht getan haben, wird TeXstudio die Einstellungen von Texmaker übernehmen, was für Sie einen erhöhten Konfigurationsaufwand bedeutet. TeXstudio greift auf diese Einstellungen zurück, weil es sich bei diesem Programm um einen *fork* von Texmaker handelt, der zuerst unter dem Namen TexmakerX entwickelt wurde.

In den Paketquellen ist TeXstudio nicht vorhanden. Die aktuelle Version des Programms laden Sie sich von [texstudio.sourceforge.net](http://texstudio.sourceforge.net) herunter. Nun wechseln Sie mit einem Dateimanager in ihr Downloadverzeichnis und starten die Installation indem Sie die Datei `texstudio_{Versionsnummer}_{i386|amd64}.deb` mit der Maus anklicken. Anschließend wird das Rootpasswort einmal abgefragt, bevor die Installation beginnt.



## 2.5 JabRef

Die aktuelle Version von JabRef laden Sie unter [jabref.sourceforge.net](http://jabref.sourceforge.net) herunter. Zur Zeit ist dies die Version 2.9.2 (Paketquelle 2.7b).

Da Sie aus den Paketquellen *jabref* bereits installiert haben, ist auch das gesamte Environment für dieses Programm mit installiert. Mit Rootrechten kopieren Sie die Datei `JabRef-{Versionsnummer}.jar` aus ihrem Downloadverzeichnis nach `/usr/share/jabref`. Dort gibt es nun zwei Dateien: die aktuelle und die aus den Paketquellen.

Der Link `/usr/share/java/jabref.jar` verweist auf die noch alte Datei von JabRef. Diesen Link `../jabref/JabRef-2.7b.jar` können Sie manuell per Konsole ändern oder indem Sie über `ALT+F2` den Dateimanager `kdesu dolphin` mit Rootrechten aufrufen. Nun gehen Sie zu dem Link und mit einem rechten Mausklick öffnen Sie das Kontextmenü, in dem Sie dann in die Auswahl **Eigenschaften** wechseln. Denn dort können Sie den Link auf `../jabref/JabRef-2.9.2.jar` anpassen.

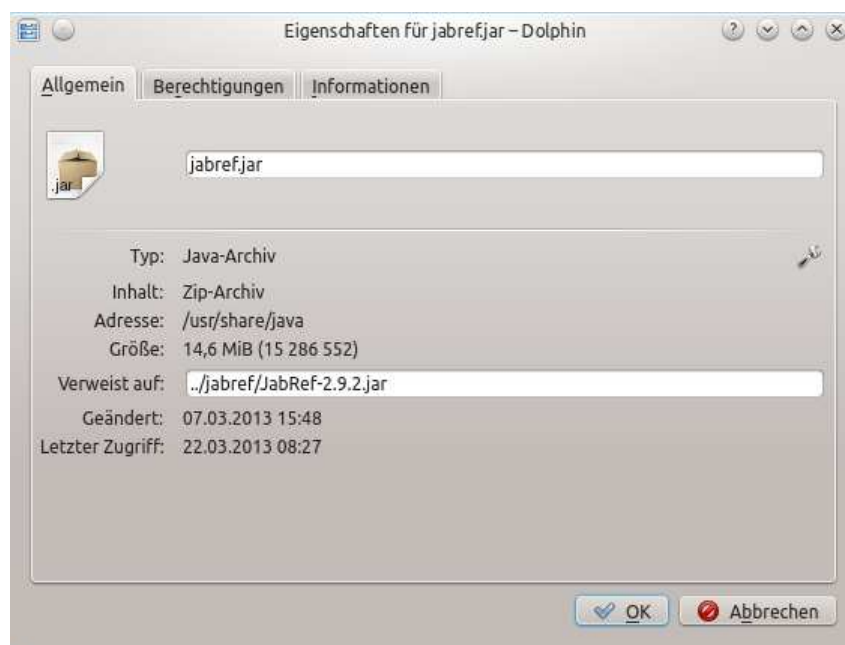


Abbildung 2.4: Linkanpassung für JabRef

Nach der Installation wechseln Sie in den Paketmanager und markieren das Programm *jabref* mit **derzeitige Version behalten**, um Updates aus den Paketquellen auszuschließen.

## 3 Konfiguration Texmaker

### 3.1 Grundkonfiguration

In der Konfigurationsdatei `~/.config/xm1/texmaker.ini` werden alle Einstellungen für Ihre Texmakersitzung abgespeichert. In der Datei `~/.config/xm1/texmakerapp.ini` steht lediglich die bevorzugte Sprache. Nachdem Sie Texmaker das erste Mal aufgerufen haben, müssen Sie über **Optionen • Texmaker Konfigurieren** dieses ihren Vorstellungen anpassen.

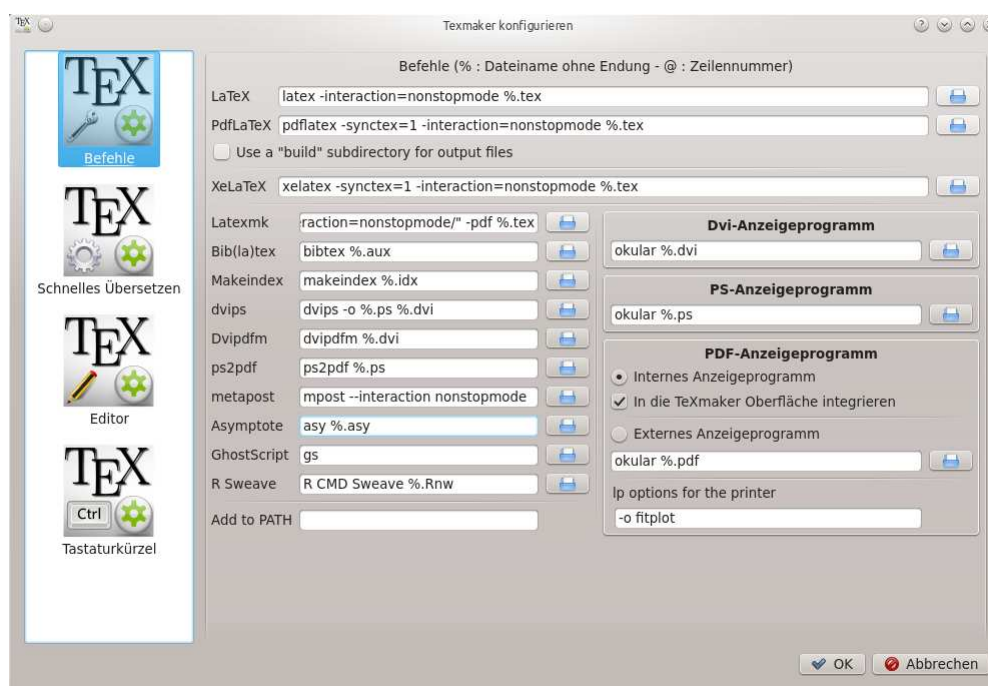


Abbildung 3.1: Die Befehlsübersicht von Texmaker

Unter der Option **Bib(la)tex** sollten Sie **biber** statt **bibtex** einstellen, da das Übersetzungsprogramm **biber** nur noch einen Durchlauf benötigt und den Code UTF-8 beherrscht, was seine Vorgänger nur mäßig können. Biber ist dabei der aktuelle Nachfolger von Bibtex[8].

Die Option **Use a "build" subdirectory for output files** ist gut geeignet, um ein wenig Ordnung in den Verzeichnisbaum zu bringen, da alle nur für die Übersetzung benötigten Dateien in ein separates Verzeichnis `build` erzeugt werden.

Wenn Sie, wie später beschrieben, einen Compiler oder ein Makefile benutzen, ist die Benutzung dieser Option nicht möglich, da das entsprechende Verzeichnis, in dem sich das Dokument befindet, nicht mit übergeben wird, sodass es im Compiler bzw. im Makefile nicht ausgewertet werden kann.

Ihre Einträge in der Präambel der Dokumente sollte hierbei folgendes beinhalten:

```
\usepackage[english,german,ngerman]{babel}
    % Mehrsprachigkeit, bei mehreren Sprachen, muss auch
    % german angegeben werden, damit die Listen nicht in
    % englisch sind

\usepackage[babel,german=quotes]{csquotes}

\usepackage[%
    backend=biber,      % zuständiges Sortierprogramm
    natbib=true,       % natbib-kompatibler Stil (als Beispiel)
    style=draft,       % Ausgabestil draft (als Beispiel)
]{biblatex}

\ExecuteBibliographyOptions{%
    bibencoding=utf8, % wenn bib in utf8 sein soll, sonst ascii
    hyperref=true,   % anklickbare Links erzeugen bei Paket hyperref
}

\bibliography{Quelle} % Quellendatei
```

Im nächsten Abschnitt werden die Einstellungen für das Übersetzen der Dokumente eingestellt. Wenn Sie die Paketen *pstricks* oder *hyperref* verwenden, werden Sie nicht umhinkommen dafür ein eigenes Skript zu entwickeln. Das Skript "*texmaker.compiler*" wird dazu später ausführlich erklärt werden.

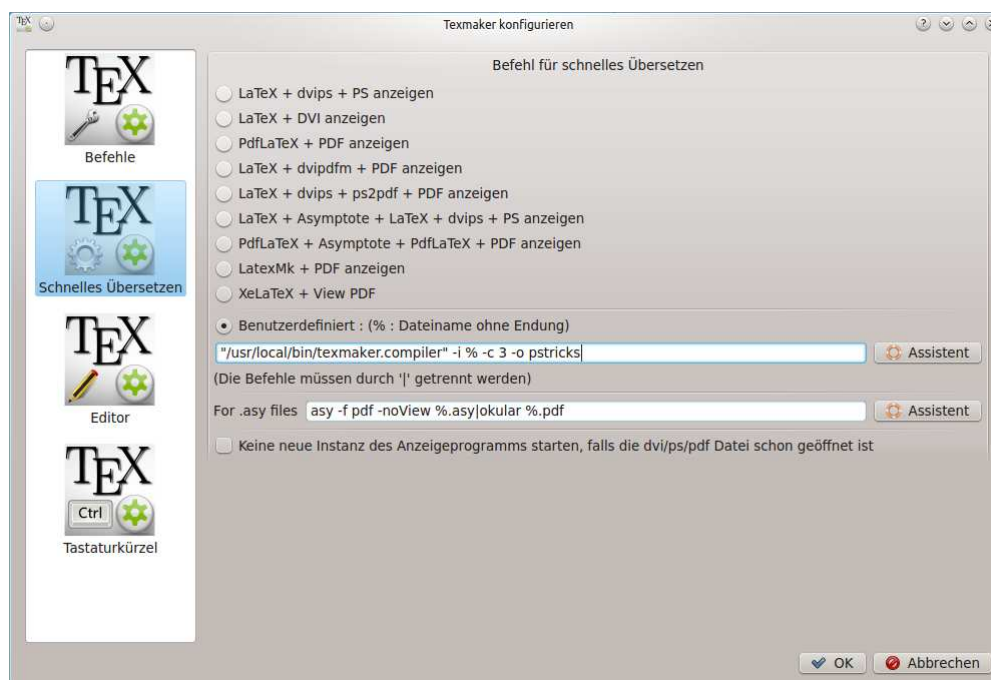


Abbildung 3.2: Auswahl der Übersetzung

## 3.2 Konfiguration Texmaker mit einem Wörterbuch

Das Wörterbuch ist eingestellt auf *myspell*. Dieses sollten Sie abändern auf den Nachfolger *hunspell*, da dieser auch den Code UTF-8 unterstützt. Falls dieses bei Ihnen nicht installiert sein sollte, können Sie dies über den Paketmanager nachholen. Für näher Informationen siehe [de.wikipedia.org/wiki/Hunspell](http://de.wikipedia.org/wiki/Hunspell) oder [wiki.ubuntuusers.de/Hunspell](http://wiki.ubuntuusers.de/Hunspell).

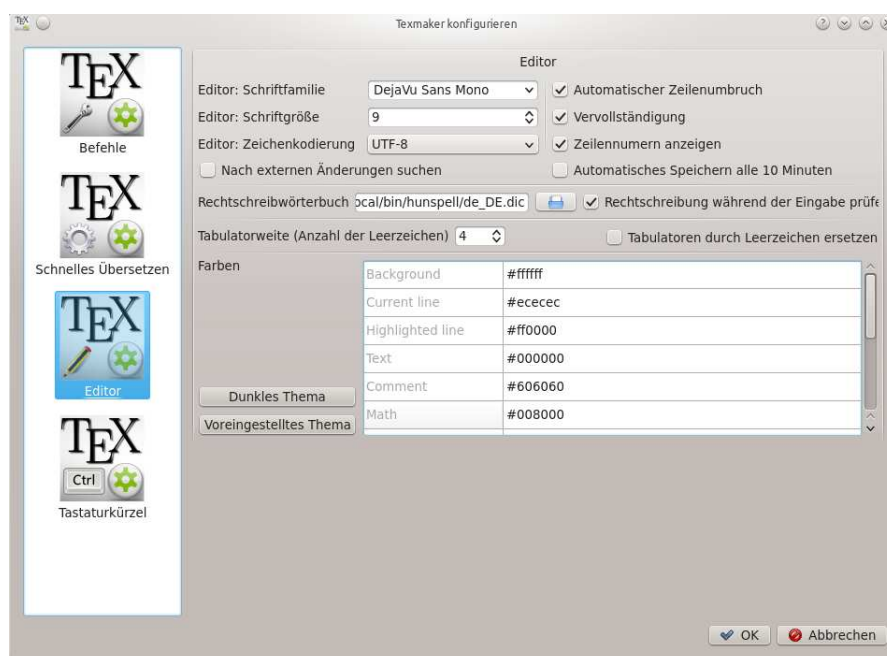


Abbildung 3.3: Auswahl der Übersetzung

Um nun das Wörterbuch durch eigene Ergänzungen zu erweitern, versehen Sie das Verzeichnis `/usr/local/bin` mit vollen Zugriffsrechten (777) und kopieren das dortige Verzeichnis `/usr/share/hunspell` nach `/usr/local/bin/hunspell`. Dadurch können Sie sich erst einmal mit dem Bearbeiten der Dateien vertraut machen, ohne ihr System zu gefährden. Diesen neuen Pfad tragen Sie unter der Option **Rechtschreibwörterbuch** ein.

In der Datei `/usr/local/bin/hunspell/de_DE.dic` können Sie nun Wörter eintragen und nach einem slash `"/` auch mit optionalen Parametern versehen. Für die Datei gelten folgende Regeln:

- Die oberste Zahl gibt die Anzahl der Wörter an, bitte mit ändern.
- Ein Wort pro Zeile.
- Kommentare werden durch einen TAB eingeleitet.

In dem unterem Beispiel sind drei Wörter eingetragen und die oberste Zahl entsprechend angepasst worden.

58792

```
This is the dictionary file of the de_DE Hunspell dictionary
derived from the igerman98 dictionary
```

```
Version: 20120607 (build 20130311)
```

```
Copyright (C) 1998-2011 Bjoern Jacke
```

```
License: GPLv2, GPLv3 or OASIS distribution license agreement There
should be a copy of all of this licenses included with every distribution
of this dictionary. Modified versions using the GPL may only include the
GPL. This dictionary is compiled from igerman98. For the purpose of the
GPL modifications of this dictionary should be made public as a patch
for igerman98 allowing to regenerate the modified dictionary.
```

```
*** ab hier eigene Änderungen ***
```

```
kubuntu
```

```
xubuntu
```

```
ubuntu
```

```
*** Ende *****
```

```
Äbte/Nm
```

Damit können Sie die von Ihnen getätigten Änderungen gut kommentieren, um Ihre Änderungen später sichern oder übertragen zu können. Die geänderte Datei steht erst nach einem Neustart von Texmaker zur Verfügung.

Da Sie sicherlich ihren Editor auf den Code UTF-8 eingestellt haben, ist es nicht ohne weiteres möglich damit das Wörterbuch zu bearbeiten. Dafür müssen Sie den Editor auf den Code ISO-8859-15 einstellen. Um aber nicht immer den Editor umzustellen, können Sie diesen – hier am Beispiel vom kate-Editor – auch mit dem entsprechenden Code starten.

```
kate -e ISO_8859-15 /usr/local/bin/hunspell/de_DE.dic
```

### 3.3 Das Wörterbuch aktualisieren

Die optionalen Parameter in der Wörterbuchdatei \*.dic werden in der Affix-Datei festgelegt. Dazu gibt es einige formale Parameter zum steuern dieser Datei:

- SET setzt die Codetabelle
- TRY definiert einzelne Buchstaben als Klassen
- REP Zeichenersetzungstabelle
- PFX Vorspann / Vorsilben
- SFX Nachsilben / Anhänge

Für weitergehende Informationen zu den Parametern verweise ich auf die englischsprachige Dokumentation [sourceforge.net/projects/hunspell/files/Hunspell/Documentation](http://sourceforge.net/projects/hunspell/files/Hunspell/Documentation) und dort auf die Datei ‚hunspell4.pdf‘.

Da alle Wörterbücher auf der igerman98-Umgebung aufsetzen, können Sie sich das aktuellste Wörterbuch vom 7.6.2012 herunterladen (Paketquelle 9.6.2011) unter der Adresse [www.j3e.de/ispell/igerman98/dict/igerman98-{Datum}.tar.bz2](http://www.j3e.de/ispell/igerman98/dict/igerman98-{Datum}.tar.bz2). Dieses Paket entpacken Sie anschließend und wechseln in das neue Verzeichnis. Zum Kompilieren des Wörterbuches benötigen Sie noch das Paket *ispell* aus den Paketquellen oder per Konsole:

```
sudo apt-get install ispell
```

Damit werden auch die Pakete *ienglish-common* und *iamerican* installiert. Nun können Sie die Wörterbücher erstellen, indem Sie eine Konsole öffnen und in das Verzeichnis der entpackten Dateien wechseln.

```
sudo make hunspell/de_DE.aff  
sudo make hunspell/de_DE.dic
```

Statt de\_DE können Sie die Wörterbücher auch für Österreich de\_AT oder der Schweiz de\_CH erzeugen. Nach dem Erstellen müssen Sie die Datei, wenn diese existieren sollte, de\_DE\_small.dic nach de\_DE.dic umbenennen. Nun können Sie ihre beiden Dateien nach /usr/local/bin/hunspell kopieren.

### 3.4 Texmaker mit einer Datei starten

Mitunter ist es sinnvoll Texmaker gleich mit einer vorher definierten T<sub>E</sub>X-Datei starten zu lassen. Dafür können Sie mittels *kmenuedit* einen neuen Eintrag erstellen, der den Inhalt *texmaker* '{Pfad/Datei}' -master besitzt. Damit startet Texmaker mit der angegebenen Datei und macht diese Datei auch gleichzeitig zur Masterdatei.

### 3.5 Eigene Makros in Texmaker erstellen

Angenehm im Texmaker ist die Tatsache, dass eigene Makros definiert werden können, um die Arbeit bei der Erstellung von Dokumenten zu vereinfachen. Die Makros erreichen Sie über die Punkte **Benutzer/in • Eigene Makros • Eigene Makros editieren**.

Um diese zu erstellen, muss man die dahinter stehenden Parameter berücksichtigen. Ist das erste Zeichen ein %, ist dieses für Texmaker das Zeichen, um eine neue Umgebung mit einem • zu generieren wohin auch der Cursor positioniert wird. Zum Beispiel % Hallo ergibt:

```
\begin{Hallo}
•
\end{Hallo}
```

Nun ist es aber mitunter hilfreich, wenn auch Kommentare mit übergeben werden können. Diese werden mit einem % eingeleitet. Die Makros werten dabei für eine neue Umgebung nur das erste Zeichen aus. Das heißt, Sie setzen ein Leerzeichen vor das % und bekommen einen Kommentar angezeigt. In diesem Beispiel wurde folgende Zeile hinzugefügt <Leerzeichen>% Das ist ein Kommentar.

```
□% Das ist ein Kommentar
\begin{Hallo}
•
\end{Hallo}
```

Sie kennen die Anzeigart der Funktionen \funktion[•]{•} mit ihren Feldern zum Ausfüllen. Dieses können Sie auch nachstellen, indem Sie für jeden angezeigten • das Zeichen @ setzen. Beim Aufruf des Makros wird dann automatisch zum ersten • gesprungen. Falls Sie einmal im Makro das Zeichen @ selbst benötigen, geben Sie @@ ein und es wird ein @ angezeigt werden.

Neben der Theorie nun ein praktisches Beispiel. Mitunter ist die float-Umgebung nicht gerade optimal, weswegen einige Tabellen oder Grafiken in einer nonfloat-Umgebung eingegeben werden müssen. Dazu wurde folgendes Makro erstellt:

```
□% Nonfloat-Umgebung
~\[\intextsep]
\begin{minipage}{\linewidth}
\centering
@
\end{minipage}
\[\intextsep]
```

## 3.6 Ein Compiler für Texmaker

Es kommt vor, dass die voreingestellten Möglichkeiten ein Dokument zu übersetzen, nicht den gewünschten Erfolg bringen. Dann kann man sich sein eigenes Übersetzungsprogramm erstellen, wie in dem hier vorgestellten Script "*texmaker.compiler*". Wenn Sie dieses nun unter der Option **Benutzerdefiniert** eintragen, funktioniert es auch, bis auf einem mit der Zeit doch lästigen Umstand. Um sich die aktuelle Ausgabe anzeigen zu lassen, müssen Sie erst F7 drücken.

Um diesen Umstand zu umgehen, und sich die Ausgabe auch sofort wieder anzeigen zu lassen, werden die Standardeinstellungen von Texmaker ein wenig umdefiniert. In der Option **Befehle** wird nun der Eintrag **XeLaTeX** mit den Einstellungen für den Compiler "*texmaker.compiler*" überschrieben und unter dem Menü **Schnelles Übersetzen** wird der Eintrag **XeLaTeX + View PDF** aktiviert.

Von nun an haben Sie wieder die gewohnte Umgebung zur Verfügung. Statt **XeLaTeX** können Sie die Einträge auch für **LatexMK** abändern.

Falls Sie die Option **Use a "build" subdirectory for output files** aktiviert haben oder mit dieser Option arbeiten möchten, können Sie den "*texmaker.compiler*" nicht einsetzen, da Texmaker nicht die notwendigen Pfadangaben mit übergibt. Auch eine Nachahmung des Compilers mit Hilfe **Benutzer/in • Eigene Befehle** schafft hier keine Abhilfe, da an die dort eingegebenen Programme das build-Verzeichnis nicht korrekt übergeben wird.

Wenn Sie das Script "*texmaker.compiler*" auf Seite 26 in einem Terminal ohne einen Parameter aufrufen, wird ihnen eine Hilfetext angezeigt werden. Ansonsten sollten Sie das Script immer mit zwei Parametern starten. Mit einem Textnamen und der Anzahl der Latexläufe, die Sie haben möchten. Die Ausgabe erfolgt dann in einem dvi-File. Wenn die Anzahl der Latexdurchläufe größer als 1 ist, werden auch die notwendigen Läufe für eine Quellenangabe bzw. Indexerstellung durchlaufen. Mit dem dritten Parameter können Sie das Ausgabeformat steuern.

## 3.7 Persönlichen Einstellungen sichern

Nachdem Sie alle ihre persönlichen Einstellungen vollzogen haben, sollten Sie diese mindestens einmal sichern. Dafür wurde ein kleines Script – siehe Seite 30 – entwickelt, das Ihre Texmaker-, Wörterbuch- und JabRef-Einstellungen in ein Backupfile sichert.

Das Backupfile wird dabei in das HOME-Verzeichnis des aufrufenden Benutzers hinterlegt und hat die Form:

```
(Datum)_(Uhrzeit)_(Aufrufer)_texmaker.backup.tar
```

Wenn Sie dieses Script im Verzeichnis `/usr/local/bin` abspeichern, können dies alle Anwender auf ihrem Rechner benutzen.



## 4 Konfiguration TeXstudio

Bevor Sie dieses Kapitel lesen, sollten Sie das Kapitel *Konfiguration Texmaker* gelesen haben, da dort bereits Themen abgehandelt worden sind, die in diesem Kapitel vorausgesetzt werden.

### 4.1 Grundkonfiguration

Bei dem ersten Aufruf von TeXstudio und dem Laden einer Datei, wurde die exotische Farbenpracht als sehr störend empfunden, da diese vom eigentlichen Zweck, dem Dokument, zu sehr ablenkt.

Bevor Sie nun in die individuelle Konfiguration einsteigen, sollten Sie unbedingt zuerst die Defaultkonfiguration über **Optionen • Profil speichern** als Sicherung abspeichern, um evtl. später drauf zugreifen zu können. TeXstudio speichert alle benutzerdefinierten Einstellungen in der Datei `~/.config/textstudio/textstudio.ini`. Es reicht natürlich aus, wenn Sie diese Datei löschen, damit Sie wieder die Defaulteinstellungen erhalten.

Unter dem Eintrag **Option • TeXstudio konfigurieren • Erzeugen** sollten Sie den Wert des Eintrages **Standardbibliografie** von **BibTeX** auf **Biber** abändern.

### 4.2 Farbeinstellungen

Um eine Editorfarbgestaltung, ähnlich wie die vom Texmaker, zu erhalten, wurden unter der Option **Option • TeXstudio konfigurieren • Syntaxhervorhebung** nachfolgende Einstellungen getätigt:

Formatbezeichnung	Änderung	von	auf
Latex Syntaxfehler	Hintergrundfarbe anpassen	#FFBF9F	#FFFFFF
Mathematisches Schlüsselwort	Textfarbe anpassen	#808000	#008000
Ausrichtungstrennzeichen (&&)	Textfarbe anpassen	#0055FF	#000000
Zeile: Bad Box	Hintergrundfarbe anpassen	#BFD6FF	#FFFFFF
Schlechtes Wort	Linienfarbe anpassen und Durchstreichen deaktivieren	#D600A4	#FFFFFF

Wenn Sie den Menüeintrag **Latex Syntaxfehler** im Original beibehalten wollen, wird er ihnen auch eine richtige Syntax wie zum Beispiel `\dq` als Fehler anzeigen. Mit der Option **Zeile: Bad Box** werden ihnen alle Textbereiche angezeigt werden, die eine Warnung `Underfull / Overfull` beinhalten. Beim Eintrag **Schlechtes Wort** werden Ihnen mitunter eine unverständliche Auswahl markiert.

### 4.3 Tooltips deaktivieren

Wenn Sie die Tooltips – das sind die schwarzen Textblöcke die Ihnen angezeigt werden, wenn Sie zulange den Cursor nicht bewegt haben – abschalten wollen, so gehen Sie in die Konfiguration von TeXstudio und aktivieren – dieses ist im unteren Bereich – das Feld **Zeige erweiterte Optionen**. Unter dem jetzt zusätzlich sichtbaren Menüpunkt **Erweiterter Editor** können Sie die beiden folgenden Felder deaktivieren:

- **Zeige Hilfe als Tooltip auf Text im Editor**
- **Zeige Vorschau als Tooltip auf Text im Editor**

Nachdem Sie diese Einstellungen vorgenommen haben, sollten Sie ihre Arbeit mittels **Option • Speichere aktuelle Konfiguration** einmal speichern. Wenn Sie dieses nicht tun, und TeXstudio beenden, wird beim nächsten Start die Defaulteinstellung geladen werden.

### 4.4 TeXstudio mit einer Datei starten

Mitunter ist es sinnvoll TeXstudio gleich mit einer vorher definierten  $\text{\TeX}$ -Datei starten zu lassen. Dafür können Sie mittels *kmenuedit* einen neuen Eintrag erstellen, der den Inhalt `texstudio '{Pfad/Datei}' --master` besitzt. Damit startet TeXstudio mit der angegebenen Datei und macht diese Datei auch gleichzeitig zur Masterdatei.

### 4.5 Das Wörterbuch aktualisieren

Analog zu Texmaker benutzt auch TeXstudio, die *igerman98/hunspell*-Umgebung, um die Rechtschreibprüfung zu vollziehen. Die Datei `/usr/share/textstudio/de_DE.dic` sollten Sie zuerst mit Schreibrechten (666) für alle Benutzer abändern, um nicht ständig ein Rootpasswort eingeben zu müssen. In dieser Datei können Sie nun Wörter eintragen und nach einem slash `"/` auch mit optionalen Parametern versehen. Für die Datei gelten folgende Regeln:

- Die oberste Zahl gibt die Anzahl der Wörter an, bitte mit ändern.
- Ein Wort pro Zeile.
- Kommentare werden durch einen TAB eingeleitet.

In dem unterem Beispiel sind drei Wörter eingetragen und die oberste Zahl entsprechend angepasst worden.

```
133382
```

```
This is the dictionary file of the de_DE Hunspell dictionary
derived from the igerman98 dictionary
```

```
Version: 20090107+frami20090221 (build 20090221)
```

```
Copyright (C) 1998-2009 Bjoern Jacke <bjoern@j3e.de>for the addon
Franz Michael Baumann <frami.baumann@web.de>
```

```
License: GPLv2, GPLv3 or OASIS distribution license agreement There
should be a copy of all of this licenses included with every distribution
of this dictionary. Modified versions using the GPL may only include the
GPL. This dictionary is compiled from igerman98. For the purpose of the
GPL modifications of this dictionary should be made public as a patch
for igerman98 allowing to regenerate the modified dictionary.
```

```
kubuntu
```

```
xubuntu
```

```
ubuntu
```

```
*** Ende *****
```

```
Äbte/Nm
```

Damit können Sie die von Ihnen getätigten Änderungen gut kommentieren, um diese später sichern oder übertragen zu können. Die geänderte Datei steht erst nach einem Neustart von TeXstudio zur Verfügung.

Da Sie sicherlich ihren Editor auf den Code UTF-8 eingestellt haben, ist es nicht ohne weiteres möglich, damit das Wörterbuch zu bearbeiten. Dafür müssen Sie den Editor auf den Code ISO-8859-15 einstellen. Um aber nicht immer den Editor umzustellen, können Sie diesen – hier am Beispiel vom kate-Editor – auch mit dem entsprechenden Code starten.

```
kate -e ISO_8859-15 /usr/share/textstudio/de_DE.dic
```

## 4.6 Eigene Makros in TeXstudio

Die Makroerstellung ist in TeXstudio ein wenig anders als in Texmaker. Die Makros erreichen Sie über das Menü **Makros • Makros bearbeiten**. Hierbei ist hervorzuheben, dass Sie die Makros auch in JavaScript erstellen können.

Um die T<sub>E</sub>X-Makros zu erstellen, muss man die dahinter stehenden Parameter berücksichtigen. Ist das erste Zeichen ein %, ist dieses für TeXstudio das Zeichen, um mit `begin{Umgebung} ... }` eine neue Umgebung zu generieren. Nun ist es aber mitunter hilfreich, wenn auch Kommentare mit übergeben werden können. Und diese werden mit einem % eingeleitet. Die Makros werten dabei für eine neue Umgebung nur das erste Zeichen aus. Das heißt, Sie setzen ein Leerzeichen vor das % und bekommen einen Kommentar angezeigt. In diesem Beispiel `<Leerzeichen>% Das ist ein Kommentar`.

```
␣% Das ist ein Kommentar
```

Sie kennen die Anzeigart der Funktionen `\funktion[•]{•}` mit ihren Feldern zum Ausfüllen. Dieses können Sie auch nachstellen, indem Sie für jeden angezeigten • die Zeichenkette %| setzen. Beim Aufruf des Makros wird dann automatisch zum ersten %| gesprungen. Leider werden die Zeichen • dabei nicht mehr angezeigt.

Neben der Theorie nun ein praktisches Beispiel. Mitunter ist die float-Umgebung nicht gerade optimal, weswegen einige Tabellen oder Grafiken in einer nonfloat-Umgebung eingegeben werden müssen. Dazu wurde folgendes Makro erstellt:

```
␣% Nonfloat-Umgebung
~\[\intextsep]
\begin{minipage}{\linewidth}
\centering
%|
\end{minipage}
\[\intextsep]
```

Nachdem Sie das Makro aufgerufen haben, springt der Cursor an die Stelle des Zeichen %| und Sie können ihre Eingaben tätigen.

## 4.7 Ein Compiler für TeXstudio

Analog zu Texmaker können Sie auch hier einen eigenen Compiler einbinden. Dazu haben Sie zwei Möglichkeiten. Der Compiler von Texmaker – ab Seite 26 – wird auch hier eingesetzt. Sie können diesen in `"texstudio.compiler"` Umbenennen. Zuerst wird die aufwendige Methode dargestellt, da diese Ihnen vielleicht später einmal von Nutzen sein kann.

Neben den bereits vorhandenen Übersetzungsmöglichkeiten, können Sie noch eigene hinzufügen. Gehen Sie in die Anzeige **Option • TeXstudio konfigurieren • Erzeugen** und dort unten auf das Feld **Zufügen**. Nun werden Ihnen zwei Felder angeboten. Den Eintrag `User0` können Sie mit einem freien Text überschreiben, wie mit `Compiler`. Im Zweiten Feld trage Sie ein:

```
"/usr/local/bin/texstudio.compiler" -i % -c 3 -o pstricks | tks:///view-pdf-internal
```

Einmal mit **OK** bestätigen und ihre neue Übersetzungsmöglichkeit steht ihnen ab sofort zur Verfügung. Vergessen Sie dabei nicht ihre Konfiguration auch zu speichern. Diese Übersetzungsmöglichkeit können Sie nun über zwei unterschiedliche Wege aufrufen. Entweder über das Menü **Tools • Benutzer • Compiler** oder über die Tastenkombination ALT-SHIFT-F1.

Da diese Tastenkombination einigen sicherlich zu aufwendig oder zu umständlich ist, besteht die zweite Möglichkeit aus dem Umdefinieren eines TeXstudio-Parameters. Für dieses Beispiel wird wieder der Eintrag für **XeLaTeX** für unsere Zwecke umdefiniert. Dafür öffnen Sie den Menüpfad **Option • TeXstudio konfigurieren • Befehle** und tragen im Feld **XeLaTeX** ein:

```
"/usr/local/bin/texstudio.compiler" -i % -c 3 -o pstricks
```

Zum Abschluss müssen Sie dem System mitteilen, dass diese neue Übersetzung von nun an standardmäßig benutzt werden soll. Dazu verändern Sie den Wert von **Standardcompiler** unter **Option • TeXstudio konfigurieren • Erzeugen** auf den neuen Wert **XeLaTeX**.

Von nun an haben Sie wieder die gewohnte Umgebung zur Verfügung. Statt **XeLaTeX** können Sie die Einträge auch für **LatexMK** oder **LuaLaTeX** abändern.

## 4.8 Persönlichen Einstellungen sichern

Nachdem Sie alle ihre persönlichen Einstellungen vollzogen haben, sollten Sie diese mindestens einmal sichern. Dafür wurde ein kleines Script – siehe Seite 31 – entwickelt, das Ihre TeXstudio-, Wörterbuch- und JabRef-Einstellungen in einem Backupfile sichert.

Das Backupfile wird dabei in das HOME-Verzeichnis des aufrufenden Benutzers hinterlegt und hat die Form:

```
(Datum)_(Uhrzeit)_(Aufrufer)_texstudio.backup.tar
```

Wenn Sie dieses Script im Verzeichnis `/usr/local/bin` abspeichern, können dies alle Anwender auf ihrem Rechner benutzen.

## 4.9 Weitere Tipps zu TeXstudio

In diesem Abschnitt werden Ihnen noch einige Tipps weitergegeben, die Ihnen beim Erstellen von Dokumenten hilfreich sein können und semantisch nicht in andere Abschnitte passen.

### 4.9.1 Syntaxhervorhebung verhindern

In der Datei `~/config/texstudio/texstudiode_DE.ign` verwaltet TeXstudio seine Wortliste, die nicht durch die Syntaxhervorhebung laufen, wie zum Beispiel das Wort `openany`. Damit können Sie dort weitere Wörter einfügen. Da die Datei im UTF-8-Format vorliegt, brauchen Sie ihren Editor nicht umzustellen.

Es sind nur einige Dinge dabei zu beachten, wie dass jedes Wort in eine eigene Zeile geschrieben wird. TeXstudio darf beim Ändern dieser Datei nicht aktiv sein, da TeXstudio sonst beim Beenden die Datei aus seinem Zwischenspeicher neu schreibt. Beim Starten von TeXstudio wird diese Datei in den Zwischenspeicher eingelesen.

### 4.9.2 Eigene Vorlagen erstellen

Wer öfter Dokumente mit TeXstudio verfasst, hat sich sicherlich einige Grunddokumente für die unterschiedlichsten Bereiche erstellt. Nun kann man bei einem neuen Dokument dieses Grundgerüst im Verzeichnisbaum suchen oder es TeXstudio überlassen.

Nachdem Sie sich Ihr Grunddokument in TeXstudio geladen haben, speichern Sie dieses unter **Datei • Erstelle Vorlage • Eingabe** ab. Anschließend wird Ihnen ein Informationsdialog angezeigt, den Sie auch ausfüllen sollten, um später die Vorlagen unterscheiden zu können. Dabei dürfen im Feld **Name** nur englischsprachige Zeichen verwendet werden.

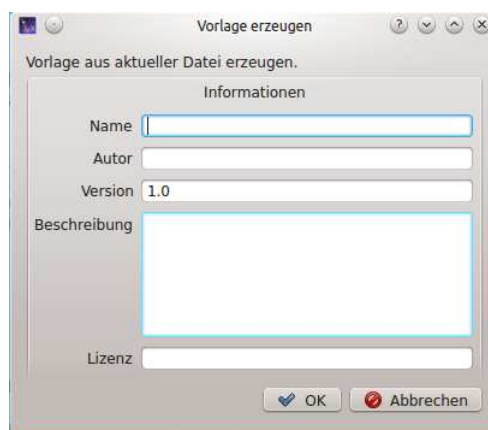


Abbildung 4.1: Der Informationsdialog

Wenn Sie von jetzt an ein neues TeX-Dokument erstellen möchten, brauchen Sie nur noch über **Datei • Neue Datei aus Vorlagen... • Benutzerdefiniert** ihre Vorlage auszuwählen und können sofort anfangen, damit zu arbeiten.

Diese Vorlagen werden im Verzeichnis `~/config/texstudio/templates/user/` in zwei Dateien abgespeichert. Die Dateien `*.json` enthalten ihre Eingaben aus dem Informationsdialog und die Dateien `*.tex` sind ihre eigentlichen Vorlagen im TeX-Format.

Bei der Erstellung eines neuen Dokuments aus den Vorlagen, stellen Sie fest, dass dieses keine Vorlagenvorschau besitzt. Um diese zu erstellen, müssen Sie lediglich ein Bild im PNG-Format unter `~/config/texstudio/templates/user/template_{Dateiname}.png` abspeichern. Beim nächsten Aufruf ihrer Vorlagen erscheint dann auch die grafische Darstellung.



Abbildung 4.2: Vorlagen mit grafischer Darstellung

### 4.9.3 Verzeichnisse entrümpeln

Wenn Sie ihr Arbeitsverzeichnis sichern wollen oder mittels Datenträger auf einem anderem Rechner transportieren möchten, sind die ganzen temporären Übersetzungsdateien überflüssig und können gelöscht werden. Bei einem händischen Vorgehen können Sie dabei die falschen Dateien löschen. Besser ist es, wenn Sie dies TeXstudio überlassen. Zu Erreichen ist dieser Punkt über **Tools • Hilfsdateien aufräumen...**

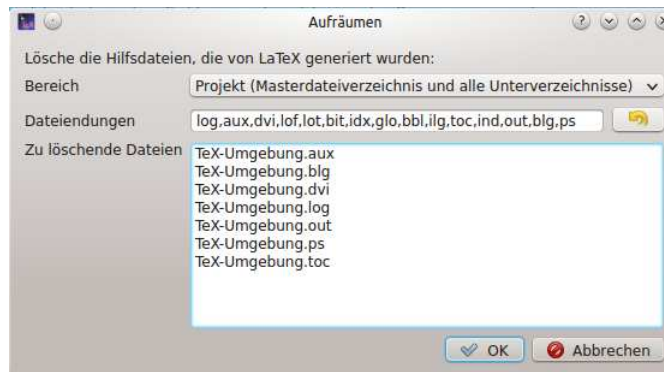


Abbildung 4.3: Das Auswahlmenü der zu löschenden Dateien

Das Feld **Bereich** sollten Sie, damit auch die Unterverzeichnisse aufgeräumt werden, abändern auf **Projekt (Masterdateiverzeichnis und alle Unterverzeichnisse)**, wenn Sie mit dieser Möglichkeit arbeiten wollen. Im Feld **Dateiendungen** können Sie nun alles angeben, was gelöscht werden soll.

Je nachdem, was Sie mit dem Projekt hervorbringen wollen, sollten Sie sich über das Endprodukt ihrer Arbeit Gedanken machen. Wenn Sie als Endprodukt eine PDF-Datei benötigen, können Sie auch die DVI-, und PS-Dateien mit löschen lassen. Die Liste können Sie jederzeit erweitern, solange die Dateiendungen mit einem Komma getrennt sind. Alle hier gemachten Änderungen werden in ihrer Datei `texmaker.ini` abgespeichert, aber erst, wenn Sie TeXstudio beenden.

## 5 Arbeiten mit Makefiles

T<sub>E</sub>X ist dafür ausgelegt, größere Werke wie Bücher, Dissertationen oder technische Dokumentationen zu erfassen und darzustellen. Wenn Sie dieses Werk jedes mal komplett neu übersetzen lassen, kostet es nicht nur unnötig Zeit für die Übersetzung, sondern auch unnötig Energie – stellen Sie sich vor Sie sitzen in der Universität am Laptop, ohne Stromanschluss, und die Festplatte muss ständig laufen –. Bei diesen Vorhaben ist es angebracht, sich Gedanken über »*Makefiles*« zu machen.

"make (engl. machen, erstellen) ist ein Build-Management-Tool, das Kommandos in Abhängigkeit von Bedingungen ausführt. Es wird hauptsächlich bei der Softwareentwicklung als Programmierwerkzeug eingesetzt.

Genutzt wird es beispielsweise, um in Projekten, die aus vielen verschiedenen Dateien mit Quellcode bestehen, automatisiert alle Arbeitsschritte (Übersetzung, Linken, Dateien kopieren etc.) zu steuern, bis hin zum fertigen, ausführbaren Programm. make ist so mächtig, dass es nicht nur zur automatisierten Steuerung der Übersetzung geeignet ist, sondern beliebige Aufgaben übernehmen kann, bei denen die zeitliche Abhängigkeit von Dateien eine Rolle spielt.

make liest ein sogenanntes Makefile, in dem die Abhängigkeiten des Übersetzungsprozesses von Programmen formalisiert erfasst sind. Diese Formalisierung beschreibt, welche Quelltextdateien auf welche Weise durch den Compiler oder durch andere Programme zu welchen Objektdateien bzw. Ergebnissen verarbeitet werden, bzw. welche Objektdateien vom Linker zu Programmbibliotheken oder ausführbaren Programmen verbunden werden. Alle Schritte erfolgen unter Beachtung der im Makefile erfassten Abhängigkeiten.

Wenn das Makefile vom make-Programm abgearbeitet wird, wird eine Umwandlung etwa einer Quelldatei in eine Objektdatei nur dann vorgenommen, wenn die Quelldatei neuer als die bereits vorliegende Version der Ergebnisdatei ist, bzw. wenn keine Ergebnisdatei vorhanden ist. Bei der Entwicklung großer Programmpakete mit vielen Quelldateien und vielen ausführbaren Programmen entfällt so die Notwendigkeit, bei einer Reihe kleiner Veränderungen alle Kompilierungen erneut durchzuführen. Der Erfolg dieser Sparmaßnahme hängt natürlich von der korrekten Angabe der Dateiabhängigkeiten ab. Alle Abhängigkeiten müssen vollständig und korrekt im Makefile beschrieben sein.

Die von make ausgeführten Befehle sind nicht auf Compiler oder Linker beschränkt. Jedes Werkzeug, das der Kommandozeileninterpreter zur Verfügung stellt, kann benutzt werden, um die gewünschten Dateimanipulationen zu erreichen. So kann man z. B. automatisch Bilder aus Texten erzeugen."<sup>1</sup>

Das bedeutet also, es werden nur die Quelldateien neu übersetzt, die auch geändert worden sind. Die restlichen Dateien liegen bereits in einer übersetzten Form vor. Da für jedes Projekt ein eigenes Makefile erzeugt werden muss, muss dieses Makefile, analog zum `texmaker.compiler`, bei der Option **schnelles Übersetzen** auch eingetragen werden.

Um eine Einführung über diese Thematik zu erhalten, können Sie eine Internetrecherche mit den Wörtern "**make + Einführung**" durchführen. Dieses wird Ihnen ein Vielzahl von Dokumente anzeigen. Dann haben Sie natürlich noch das make-Manual, aufzurufen mit `man make` in einer Konsole.

---

<sup>1</sup>vgl.: [de.wikipedia.org/wiki/Make](http://de.wikipedia.org/wiki/Make), letzter Zugriff am 25.3.2013



Hier können keine Beispiele für derart projektbezogene Abläufe angegeben werden, deswegen wird auf die Literatur<sup>2</sup> verwiesen. Für dieses Dokument sieht das Makefile wie unten abgebildet aus. Dieses ist noch überschaubar, da das Dokument wie auch die Bilder im gleichen Verzeichnis vorhanden sind und keine weiteren Abhängigkeiten existieren.

```
MAIN = TeX-Umgebung
MAINDIR = $HOME/Dokumente/TeX-Umgebung
EPSFILES = $(wildcard *.eps)
```

```
all: $(MAIN).pdf
#      okular $(MAIN).pdf
```

```
$(MAIN).pdf: $(MAIN).tex $(EPSFILES)
        latex $(MAIN).tex
        latex $(MAIN).tex
        dvips $(MAIN).dvi
        ps2pdf $(MAIN).ps
```

Der PDF-Betrachter ist hierbei deaktiviert, da der interne Betrachter benutzt werden soll. Eintragen können Sie dieses unter der Option **XeLaTeX** mit den Parametern *make*, solange die Übersetzungsdatei den Namen *Makefile* besitzt und im Projektverzeichnis steht. Ansonsten müssten Sie den Pfad und den Dateinamen *make -f {Pfad}/Name* noch mit angeben.

---

<sup>2</sup>LaTeX Das umfassende Handbuch, von Karsten Günther, Galileo Press GmbH, 1. Auflage 2004, ISBN 3-89842-510-X, Seite 517 - 527

## 6 JabRef

Ihre persönlichen Einstellungen, wie die anzuzeigenden Felder, speichert JabRef im Verzeichnis `~/ .java/.userPrefs/net/sf/jabref` ab.

### 6.1 Benutzerdefinierte Felder

Im Programm JabRef wurde ein Bibliotheksfeld für eigene Zwecke umdefiniert. Das Feld *note* wird benutzt, um den Aufstellungsort einzelner Bücher zu hinterlegen. Zum Beispiel für die Hamburger Staatsbibliothek *Stabi: H/Gen 035*. Diese Feld erscheint auch im Literaturverzeichnis, um die Arbeit mit dem Dokument zu vereinfachen.

Im fertigen Dokument soll dieses Feld natürlich nicht mehr erscheinen. Deswegen wurden in der Präambel des Dokuments folgende Einträge erstellt:

```
\usepackage{ifthen}

\newcommand{\entwurf}{true} % Entwurf (true), Endversion (false)

% Bei der Endausgabe ohne das Feld "Note" die Literaturliste erstellen
% -----
\ifthenelse{\boolean{\entwurf}}{\}{%
\DeclareSourceMap{
  \maps[datatype=bibtex]{
    \map{
      \step[fieldset=note, null]
    }
  }
}}
```

Nachdem das Paket *ifthen* geladen ist, wurde ein Schalter *entwurf* definiert, der nur die beiden Werte *true/false* annehmen kann. Anschließend wird der Wert des Schalters geprüft und das Feld *note* entweder so belassen oder mit *null* gefüllt.

## 6.2 Datenaustausch JabRef $\Rightarrow$ Texmaker

Zwischen JabRef und Texmaker ist ein direkter Austausch der cite-Felder nicht möglich. Nachdem Sie sich eine Quelle markiert haben, können Sie, innerhalb von JabRef, dieses Feld entweder mit STRG+K oder über **Markieren • cite{BiBTeX key} kopieren** in die Zwischenablage kopieren. In dem Programm Texmaker können Sie diese Zwischenablage entweder mit STRG+V oder über die rechte Maustaste und **Einfügen** einkopieren.

## 6.3 Datenaustausch JabRef $\Rightarrow$ TeXstudio

Nachdem Sie in JabRef als **Externe Anwendung auswählen** das Programm **TeXstudio** ausgewählt haben – über das senkrechte Icon –, können Sie einen beliebigen Datensatz markieren und mit einem Klick auf das Feld **Auswahl in TeXstudio einfügen** wird der cite-Eintrag in ihr Dokument kopiert, an die Stelle wo der Cursor steht. Sie können dieses natürlich auch mit der Tastenkombination STRG-L erledigen.



Abbildung 6.1: Editorauswahl einstellen

### 6.3.1 Zitierweise einstellen

Normalerweise werden Zitate mit dem Eintrag `cite{}` an TeXstudio übergeben. Wer mit einem anderen Befehl arbeiten möchte, wie zum Beispiel `autocite{}`, kann dieses unter dem Pfad **Option • Einstellungen • externe Programme • Einstellungen für TeXstudio** nach seinen Wünschen entsprechend abändern.

## 6.4 Schlüsselbereich

Leider ist es zur Zeit noch nicht möglich, den *bibtexkey* im Format UTF-8 anzugeben. Hier sollten nur die englischsprachigen Schriftzeichen sowie die Zeichen `:-_` vergeben werden.

## 7 Listings

### 7.1 Compiler

```
#!/bin/bash
#
# Übersetzungsscript für Texmaker/TeXstudio
# -----
#
# Author: Günther Zander, Hamburg 3/2013
# License: GPL
# File: /usr/local/bin/[texmaker|texstudio].compiler
# Version: 0.5

function_help(){
echo ""
echo " `basename $0` -i Eingabedatei [-c Laufanzahl | -o Ausgabeformat | -d]"
echo ""
echo " Dieses Script soll den TeX-Übersetzungslauf automatisieren."
echo " Folgende Parameter sind zu übergeben:"
echo "   -i Eigabedatei   Das TeX-Dokument ohne die Endung .tex"
echo "   -c Laufanzahl    Anzahl der Übersetzungsläufe"
echo "   -o Ausgabeformat dvi      -> TeX-Standard"
echo "                   ps       -> dvips"
echo "                   pdf       -> dvipdfm"
echo "                   pstricks -> dvips + ps2pdf"
echo ""
echo "   -d                Zum Abschluß werden nicht mehr benötigte"
echo "                   Dateien gelöscht."
echo ""
echo " Im Übersetzungslauf sind die Abläufe für das Literatur-"
echo " verzeichnis und das Indexverzeichnis mit integriert."
echo ""
echo " Die Reihenfolge der Parameter ist dabei unerheblich"
echo " Die Rückgabewerte sind ungleich 0 für Fehler"
echo ""
echo " Die Logdatei finden Sie unter:"
echo "   $LOG"
echo ""
}

# -- Grundeinstellungen -----
#
# Pfad zur LOG-Datei und das alte LOG löschen
LOG="/tmp/tex-compiler.log"
rm -f "$LOG"
# Wie oft soll TeX, standardmäßig, ausgeführt werden
LAUF=1
# Standardendung der übergebenen Datei
ENDUNG=".tex"
#Standardausgabe
AUSGABE="dvi"
```

```
#
# -----

# Hilfe anzeigen
if [ "$1" == "" ]; then
    function_help
    exit 0
fi

# Übergabeparameter setzen
while getopts dc:o:i: VAR
do
    case $VAR in
        d) # Temporäre Dateien löschen
            DEL=true
            echo "Löschparameter ist gesetzt." >> "$LOG";;

        c) # Anzahl TeX durchläufe
            if ! [ "$OPTARG" == "" ]; then
                LAUF=$OPTARG
            fi
            echo "Laufanzahl auf $LAUF gesetzt." >> "$LOG";;

        o) # Ausgabeformat festlegen
            if ! [ "$OPTARG" == "" ]; then
                AUSGABE=$OPTARG
            fi
            echo "Ausgabeformat auf $AUSGABE gesetzt." >> "$LOG";;

        i) # Eingabedatei
            TEX=$OPTARG;;

        *) # Hilfstext aufrufen
            function_help
            exit 1;;

        esac
    done

# Ohne Eingabedatei keine Bearbeitung
if ! [ "$TEX" == "" ]; then
    declare -i i=0
    while [[ $((++i)) -le $LAUF ]]; do
        # TeX-Läufe
        latex -interaction=nonstopmode "$TEX$ENDUNG"
        ERROR="$?"
        if ! [ "$ERROR" == "0" ]; then
            echo -n "Bei TeX-Lauf "$i" ist der Fehler "$ERROR" aufgetreten" >> "$LOG"
            echo ", Datei: "$TEX$ENDUNG >> "$LOG"
            exit 1
        fi
        echo "TeX-Lauf "$i" mit "$TEX$ENDUNG" war erfolgreich" >> "$LOG"
    done

# Biberlauf
if [ "$i" == "1" ]; then
    biber "$TEX"
    ERROR="$?"
fi
```

```
if [ "$ERROR" == "1" ]; then
    echo -n "Beim Biberlauf ist der Fehler \"$ERROR\" aufgetreten," >> "$LOG"
    echo " Datei: \"$TEX >> \"$LOG"
    exit 1
fi
echo -n "Biberlauf beim TeX-Lauf \"$i\" mit Returncode \"$ERROR >> \"$LOG"
echo " war erfolgreich" >> "$LOG"
fi

# Indexlauf
if [ "$i" == "1" -a -e "$TEX.idx" ]; then
    makeindex "$TEX".idx
    ERROR="$?"
    if [ "$ERROR" == "1" ]; then
        echo -n "Beim Indexlauf ist der Fehler \"$ERROR\" aufgetreten," >> "$LOG"
        echo " Datei: \"$TEX >> \"$LOG"
        exit 1
    fi
    echo -n "Indexlauf bei TeX-Lauf \"$i\" mit Returncode \"$ERROR >> \"$LOG"
    echo " war erfolgreich" >> "$LOG"
fi
done

# In PS-File umwandeln
if [ "$AUSGABE" == "ps" ]; then
    dvips "$TEX".dvi
    ERROR="$?"
    if ! [ "$ERROR" == "0" ]; then
        echo -n "Bei der PS-Umsetzung ist der Fehler \"$ERROR >> \"$LOG"
        echo " aufgetreten, Datei: \"$TEX >> \"$LOG"
        exit 1
    fi
    echo "Die PS-Umsetzung von der Datei \"$TEX\" war erfolgreich" >> "$LOG"
fi

# In PDF-File umwandeln
if [ "$AUSGABE" == "pdf" ]; then
    dvi2pdf "$TEX".dvi
    ERROR="$?"
    if ! [ "$ERROR" == "0" ]; then
        echo -n "Bei der PDF-Umsetzung ist der Fehler \"$ERROR >> \"$LOG"
        echo " aufgetreten, Datei: \"$TEX >> \"$LOG"
        exit 1
    fi
    echo "Die PDF-Umsetzung von der Datei \"$TEX\" war erfolgreich" >> "$LOG"
fi

# In PDF-File umwandeln für pstricks
if [ "$AUSGABE" == "pstricks" ]; then
    dvips "$TEX".dvi
    ERROR="$?"
    if ! [ "$ERROR" == "0" ]; then
        echo -n "Bei der pstricks-Umsetzung [ps] ist der Fehler \"$ERROR >> \"$LOG"
        echo " aufgetreten, Datei: \"$TEX >> \"$LOG"
        exit 1
    fi
fi
```

```
ps2pdf "$TEX".ps
ERROR="$?"
if ! [ "$ERROR" == "0" ]; then
    echo -n "Bei der pstricks-Umsetzung [ps2pdf] ist der Fehler " >> "$LOG"
    echo "$ERROR" aufgetreten, Datei: "$TEX" >> "$LOG"
    exit 1
fi
echo -n "Die pstricks-Umwandlung von der Datei "$TEX" war" >> "$LOG"
echo " erfolgreich" >> "$LOG"
fi
fi

# Aufräumen
if [ "$DEL" == "true" ]; then
    for I in aux bbl blg lot lof toc out
    do
        rm -f *.$I
        echo "Lösche *.$I" >> "$LOG"
    done

    if [ "$AUSGABE" == "ps" ]; then
        rm -f *.dvi
        echo "Lösche *.dvi" >> "$LOG"
    fi

    if [ "$AUSGABE" == "pdf" ]; then
        rm -f *.dvi
        echo "Lösche *.dvi" >> "$LOG"
    fi

    if [ "$AUSGABE" == "pstricks" ]; then
        rm -f *.dvi
        rm -f *.ps
        echo "Lösche *.dvi und *.ps" >> "$LOG"
    fi

    if [ "$AUSGABE" == "dvi" ]; then
        rm -f *.pdf
        rm -f *.ps
        echo "Lösche *.pdf und *.ps" >> "$LOG"
    fi
fi
fi

exit 0
```

## 7.2 texmaker.backup

```
#!/bin/bash
#
# Texmaker Sicherung
# -----
#
# Author: Günther Zander, Hamburg 3/2013
# License: GPL
# File: /usr/local/bin/texmaker.backup
# Version: 0.1
#
# Es wird eine Sicherungsdatei erzeugt, im HOME-Verzeichnis des
# Aurufers im Format:
#
# YYYY-MM-TT_(Uhrzeit)_(Aufrufer)_texmaker.backup.tar
#
#
# Verzeichnisse zum sichern
# -----
# Texmaker
TEX="$HOME/.config/xml/"
# JabRef
JAB="$HOME/.java/.userPrefs/net/sf/jabref/"
# Wörterbuch
BUCH="/usr/local/bin/hunspell/"
#
# Programme sichern
# -----
# Compiler
COMP="/usr/local/bin/texmaker.compiler"
#
# Backupname erstellen
# -----
BACKUP=`date +%F_%H%M`_"$USER"_texmaker.backup"
#
# Sicherung
# -----
#
tar cvf $HOME/"$BACKUP" $TEX $JAB $BUCH $COMP
```



## 7.3 texstudio.backup

```
#!/bin/bash
#
# Texstudio Sicherung
# -----
#
# Author: Günther Zander, Hamburg 3/2013
# License: GPL
# File:    /usr/local/bin/texstudio.backup
# Version: 0.1
#
# Es wird eine Sicherungsdatei erzeugt, im HOME-Verzeichnis des
# Aurufers im Format:
#
# YYYY-MM-TT_(Uhrzeit)_(Aufrufer)_texstudio.backup.tar
#
#
# Verzeichnisse zum sichern
# -----
# Texmaker
TEX="$HOME/.config/texstudio/"
# JabRef
JAB="$HOME/.java/.userPrefs/net/sf/jabref/"
#
# Programme sichern
# -----
# Compiler
COMP="/usr/local/bin/texstudio.compiler"
#
# Wörterbuch
BUCH="/usr/share/texstudio/de_DE.dic"
#
# Backupname erstellen
# -----
BACKUP=`date +%F_%H%M`_"$USER"_texstudio.backup"
#
# Sicherung
# -----
#
tar cvf $HOME/"$BACKUP" $TEX $JAB $BUCH $COMP
```