

Elke und Michael Niedermaier

**FRANZIS**  
PROFESSIONAL SERIES

Inklusive  
**DVD 9**  
**TeX Collection**

inkl. CTAN-Abzug,  
proTeXt und TeXLive  
in Kooperation mit  
dante e.V.

2. vollständig aktualisierte und überarbeitete Neuauflage

# L<sup>A</sup>T<sub>E</sub>X

## Das Praxisbuch

▶ Inkl. DVD 9 TeX Collection 2004



Das PDF-Dokument wurde dem von Elke und Michael Niedermaier verfassten Werk „LaTeX - das Praxisbuch“ (ISBN 3-7723-6434-9) entnommen, das im Franzis Verlag mit Sitz in 85586 Poing erschienen ist. Die Nutzung des Dokuments ist ausschließlich in Verbindung mit dem Paket „Writer2LaTeX“ gestattet. Sämtliche Rechte verbleiben beim Franzis Verlag. Eine unerlaubte Vervielfältigung ist verboten. Die Druckauflösung wurde im Übrigen auf 150 dpi begrenzt.

## 9 Fremdformate umwandeln

Oft ist es der Fall, dass man eine Menge an Altdokumenten im Word-, OpenOffice- oder anderen Formaten vorliegen hat. Es gibt hier verschiedene Hilfsprogramme, mit denen man aus diesen Formaten  $\LaTeX$ -Dokumente erzeugen kann. In der Dante-FAQ findet man sehr viele Hinweise darüber.

Wir möchten hier speziell das Hilfsprogramm `writer2latex` von Henrik Just<sup>1</sup> vorstellen, da es sehr viele Möglichkeiten bietet und der *Gnu Lesser General Public License* unterliegt.

Besitzt man ein anderes Format, sollte man zuerst in der FAQ nach einer Lösung suchen, dann „Google“ nach Lösungsvorschlägen hierzu befragen und im letzten Fall die Notlösung auf Seite 544 verwenden.

### Inhalt

---

9.1	OpenOffice-/RTF-/Word-Format . . . . .	530
9.2	HTML . . . . .	541
9.3	DocBook . . . . .	543
9.4	Die Notlösung . . . . .	544

---



---

<sup>1</sup> Wir möchten hier Henrik Just ein großes Dankeschön für die Unterstützung und den Einbau der vielen Wünsche unsererseits übermitteln.

## 9.1 OpenOffice-/RTF-/Word-Format

Das Hilfsprogramm `writer2latex` ist ein Java-Programm, welches eigenständig oder als Plugin in OpenOffice verwendet werden kann. Dabei wird die Speicherstruktur von OpenOffice (eine xml-Datei) analysiert und entsprechend in  $\LaTeX$ -Syntax umgesetzt (siehe Abbildung 9.1).

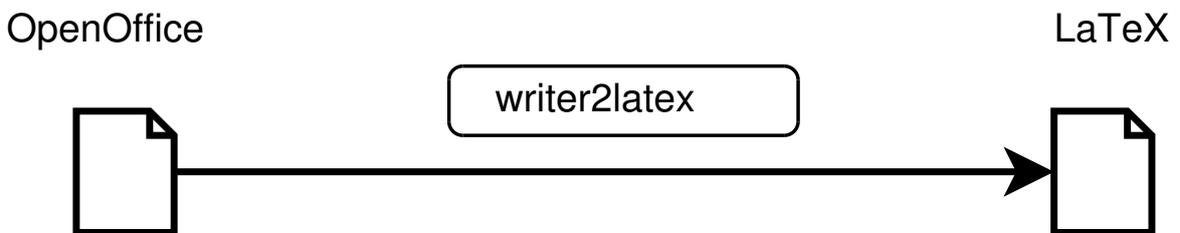


Abbildung 9.1: OpenOffice nach  $\LaTeX$  umwandeln

### RTF-/Word-Format

Lädt man unter [OpenOffice.org](http://OpenOffice.org) eine RTF- bzw. Word-Datei und speichert sie anschließend im OpenOffice-Format, so kann über diesen kleinen Umweg auch in  $\LaTeX$  konvertiert werden.

## Installation

`writer2latex` kann auf der Homepage<sup>2</sup> als ZIP-Archiv heruntergeladen werden. Im Moment ist hier die Version 0.4 aktuell, da aber der Autor sehr fleißig immer weitere Möglichkeiten einbaut, wird schon bei Erscheinen dieses Buches sicherlich eine aktuellere Version vorhanden sein.

Bei der nachfolgenden Installation wird angenommen, dass bereits Java installiert ist<sup>3</sup>.

### Installation unter MS Windows

1. Entpacken Sie die Datei `writer2latex04beta.zip` in ein Verzeichnis Ihrer Wahl. Dabei wird ein Verzeichnis mit dem Namen `writer2latex04` angelegt.
2. Fügen Sie dieses Verzeichnis der Umgebungsvariablen `PATH` hinzu.
3. Öffnen Sie die Datei `w2l.bat` mit einem Texteditor, und ändern Sie das Verzeichnis beim Eintrag `W2LPATH` z. B. folgendermaßen:  

```
set W2LPATH="c:\writer2latex04"
```

<sup>2</sup> <http://www.hj-gym.dk/~hj/writer2latex/>

<sup>3</sup> Mehr zu Java und dessen Installation finden Sie unter <http://java.sun.com/j2se/> und <http://www.selfjava.de/java0033.htm>.

4. `w2l` geht davon aus, dass die Java-Laufzeitumgebung in `PATH` enthalten ist. Ist dies nicht der Fall, so muss der Eintrag `JAVAEXE` entsprechend angepasst werden, z. B. folgendermaßen:

```
set JAVAEXE="C:\j2sdk1.4.0_01\bin\java"
```

Anschließend kann `writer2latex` mit dem Befehl `w2l` aufgerufen werden.

### Installation unter Linux

1. Entpacken Sie die Datei `writer2latex04beta.zip` in ein Verzeichnis Ihrer Wahl. Dabei wird ein Verzeichnis mit dem Namen `writer2latex04` angelegt.
2. Fügen Sie dieses Verzeichnis der Umgebungsvariablen `PATH` hinzu, oder erzeugen Sie einen symbolischen Link in ein Verzeichnis, welches in `PATH` enthalten ist.
3. Öffnen Sie die Datei `w2l` mit einem Texteditor, und ändern Sie das Verzeichnis beim Eintrag `W2LPATH` z. B. folgendermaßen:

```
set W2LPATH="/home/username/writer2latex04"
```

4. `w2l` geht davon aus, dass die Java-Laufzeitumgebung in `PATH` enthalten ist. Ist dies nicht der Fall, so muss der Eintrag `JAVAEXE` entsprechend angepasst werden, z. B. folgendermaßen:

```
set JAVAEXE="/path/to/java/executable/"
```

5. Setzen Sie Ausführungsrechte für die Datei `w2l`:

```
chmod +x w2l
```

Anschließend kann `writer2latex` mit dem Befehl `w2l` aufgerufen werden.

Mehr zur Installation finden Sie in der `writer2latex`-Dokumentation (Abschnitt *Installation*).

### Aufruf

`writer2latex` wird mit `w2l` aufgerufen. Dabei stehen folgende Parameter zur Verfügung:

```
w2l [-ultraclean|-clean|-pdfscreen|-pdfprint|-article]
    [-config <configfile>]
    <writer document to convert> <output-tex>
```

Dabei wird mit `-ultraclean`, usw. festgelegt, wie OpenOffice-Formate in  $\text{\LaTeX}$  übernommen werden. Der Aufruf `-ultraclean` steht für `-config ultraclean.xml` usw. Tabelle 9.1 auf der nächsten Seite zeigt die möglichen Standard-Einstellungen.

### Test

Für unseren Test haben wir ein OO-Dokument erstellt, welches Überschriften, normalen Text, eine Tabelle, ein Bild, eine Aufzählung und ein Listing enthält (siehe Abbildung 9.2 auf der nächsten Seite).

Kurzversion	Langversion	Beschreibung
-ultraclean	-config ultraclean.xml	Es werden so gut wie alle Formatierungen entfernt.
-clean	-config clean.xml	Es werden viele Formatierungen entfernt.
-pdfscreen	-config pdfscreen.xml	Die Ausgabe wird für eine Bildschirmpräsentation (pdf $\LaTeX$ ) optimiert.
-pdfprint	-config pdfprint.xml	Die Ausgabe wird für eine Druckausgabe (pdf $\LaTeX$ ) optimiert.
-article	-config article.xml	Es wird die Klasse <code>article</code> verwendet, und es bleiben sehr viele Formatierungen erhalten.

Tabelle 9.1: Standard-Konfigurationen

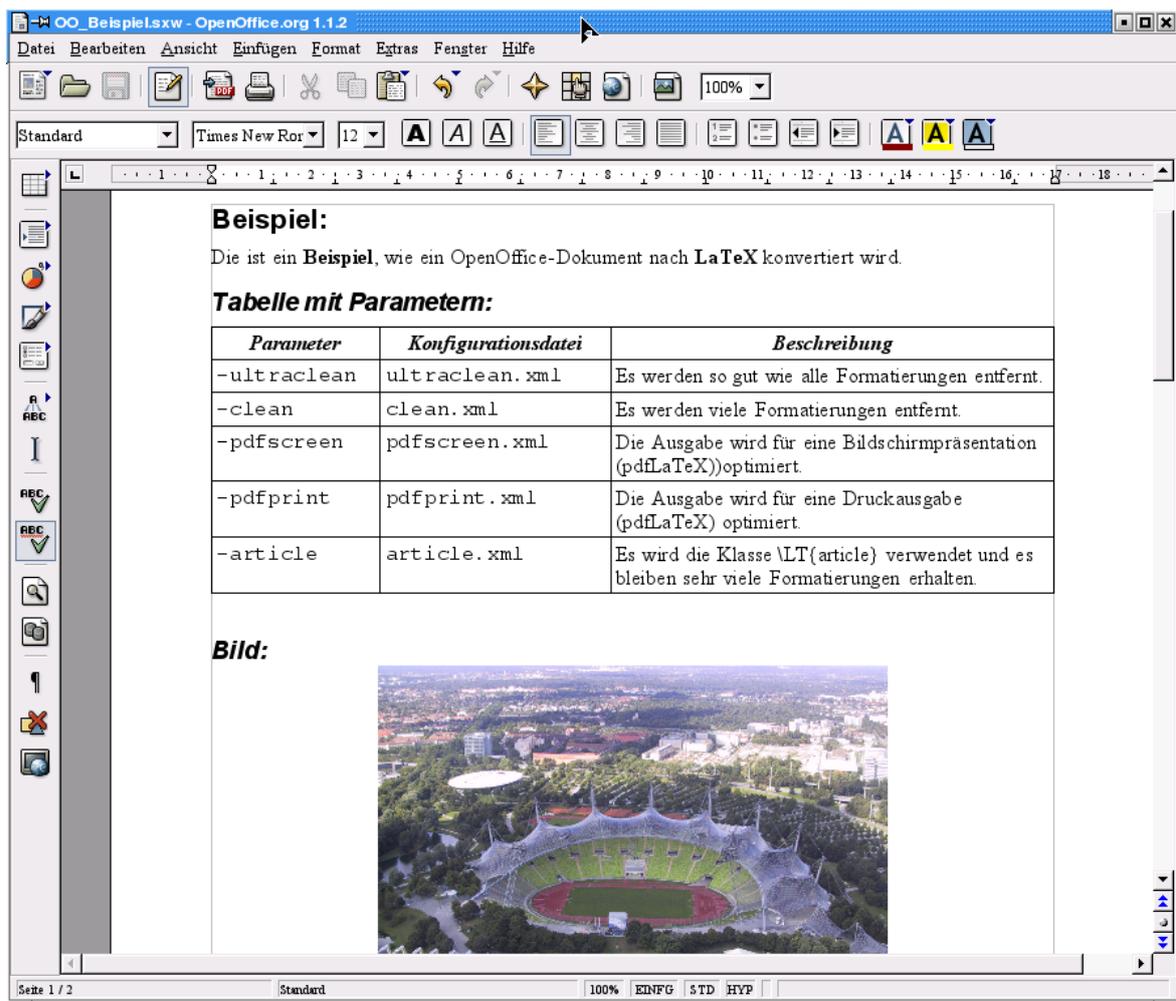


Abbildung 9.2: OO-Beispiel-Dokument

Als Erstes werden wir die Einstellung `ultraclean` verwenden. Dazu lautet der Aufruf:

```
w2l -ultraclean OO_Beispiel.sxw OO_Beispiel_ultraclean.tex
```

Als Ergebnis erhalten wir die Datei `OO_Beispiel_ultraclean.tex` (siehe Beispiel 9.1, 9.2 und 9.3).

```
% This file was converted to LaTeX by Writer2LaTeX ver. 0.3.3g
% see http://www.hj-gym.dk/~hj/writer2latex for more info
\documentclass[12pt]{article}
\usepackage[ascii]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[english]{babel}
\usepackage{amsmath,amssymb,amsfonts,textcomp}
\usepackage{calc}
\usepackage{longtable}
\usepackage{graphicx}
\usepackage{oomath}
```

Beispiel 9.1: Ergebnis mit der Einstellung `ultraclean` (Auszug)  
(source/w2l/OO\_Beispiel\_ultraclean.tex)

```
\begin{document}
\section{Beispiel:}
Die ist ein \textbf{Beispiel}, wie ein OpenOffice{-Dokument nach
\textbf{LaTeX} konvertiert wird.

\subsection{Tabelle mit Parametern:}
\begin{longtable}[c]{|p{3.199cm}|p{4.4430003cm}|p{8.757cm}|}
\hline
{\bfseries\itshape
Parameter
}
```

Beispiel 9.2: Auszug Teil 2

```
\subsection{Formel:}

$$y = \ln(x)$$


\subsection{Aufz\textbackslash{}ahlung:}
\liststyleLi
\begin{itemize}
\item Montag
\item Dienstag
\item Mittwoch
\end{itemize}
```

Beispiel 9.3: Auszug Teil 3

Man sieht, dass hier in der Präambel nur wenige Pakete geladen werden und die verwendeten Einstellungen überschaubar sind. Anschließend kommt der eigentliche Dokumentinhalt, in dem die meisten OO-Einstellungen nicht übernommen worden sind. So ein  $\text{\LaTeX}$ -Dokument lässt sich sehr einfach auf eigene Bedürfnisse anpassen.

## Anpassung der Konfiguration

Damit das spätere Anpassen des Dokumentes auf ein Minimum reduziert wird bzw. evtl. ganz wegfällt, kann man bereits die Konfigurationsdatei entsprechend verändern. Dazu haben wir eine eigene Konfigurationsdatei erstellt (diese wurde von `ultraclean.xml` übernommen und mit zusätzlichen Kommentaren zur Erläuterung versehen).

Zusätzlich sollte die Dokumentation, die im ZIP-Archiv enthalten ist, hinzugezogen werden.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<config>
  <option name="create_user_config" value="false" />
  <!--
    Alle Konfigurationseinstellungen werden in einer
    xml-Datei gespeichert.
  -->
  <option name="use_oomath" value="false" />
  <!--
    Für Formeln wird das Paket 'oomath.sty' verwendet.
  -->
  <option name="documentclass" value="scrartcl" />
  <!--
    Legt die Dokumentenklasse fest.
  -->
  <option name="global_options" value="a4paper" />
  <!--
    Legt die globalen Optionen für die Dokumentenklasse fest.
    z.B. \documentclass[a4paper]{article}
  -->
  <option name="backend" value="pdftex" />
  <!--
    Legt das zu verwendende Backend fest.
    z.B. generic, dvips, pdftex
  -->
  <option name="inputencoding" value="latin1" />
  <!--
    Legt das Inputencoding fest.
    z.B. latin1 fpr \usepackage[latin1]{inputenc}
  -->
  <option name="multilingual" value="false" />
  <!--
    Legt fest, ob das OO-Dokument mehrsprachig ist.
    Wird hier 'true' verwendet, so werden alle
    Sprachinformationen aus OO verwendet.
  -->
  <option name="greek_math" value="true" />
  <!--
    Es werden griechische Symbole im Mathematikmodus verwendet.
  -->
  <option name="use_pifont" value="true" />
  <!--
    Es wird das Paket 'pifont.sty' verwendet.
  -->
  <option name="use_ifsym" value="false" />
  <!--
```

```

    Es wird das Paket 'ifsym.sty' verwendet.
-->
<option name="use_wasysym" value="false" />
<!--
    Es wird das Paket 'wasysym.sty' verwendet.
-->
<option name="use_bbding" value="false" />
<!--
    Es wird das Paket 'bbding.sty' verwendet.
    (eine ähnliche Version zu pifont)
-->
<option name="use_eurosym" value="false" />
<!--
    Es wird das Paket 'eurosym.sty' verwendet.
-->
<option name="use_tipa" value="false" />
<!--
    Es wird das Paket 'tipa.sty' verwendet.
-->
<option name="use_color" value="true" />
<!--
    Es wird Farbe in Zusammenhang mit dem Paket 'hyperref'
    verwendet.
-->
<option name="use_hyperref" value="true" />
<!--
    Es wird das Paket 'hyperref.sty' verwendet.
-->
<option name="use_endnotes" value="false" />
<!--
    Es wird das Paket 'endnotes.sty' verwendet.
-->
<option name="use_ulem" value="false" />
<!--
    Es wird das Paket 'ulem.sty' verwendet.
    In einer späteren Version soll das Paket 'soul'
    unterstützt werden.
-->
<option name="use_bibtex" value="true" />
<!--
    Es wird bibtex verwendet.
-->
<option name="use_lastpage" value="false" />
<!--
    Es wird das Paket 'lastpage.sty' verwendet.
-->
<option name="bibtex_style" value="plain" />
<!--
    Der bibtex-Style wird festgelegt.
-->
<option name="character_formatting" value="ignore_all" />
<!--
    Es wird festgelegt, wie die Formatierung von einzelnen Zeichen
    übernommen oder nicht übernommen werden soll. Mögliche Werte sind:
    - ignore_all      - ignore_fontsize - ignore_font
    - ignore_hard    - convert_all
-->
<option name="paragraph_formatting" value="ignore_all" />

```

```

<!--
  Es wird festgelegt, wie die Formatierung von einzelnen Zeichen
  übernommen oder nicht übernommen werden soll. Mögliche Werte sind:
  - ignore_all  -ignore_hard  -convert_all
-->
<option name="list_formatting" value="ignore_all" />
<!--
  Es wird festgelegt, wie die Formatierung von einzelnen Zeichen
  übernommen oder nicht übernommen werden soll. Mögliche Werte sind:
  -ignore_all  -convert_numbering  -convert_all
-->
<option name="heading_formatting" value="ignore_all" />
<!--
  Es wird festgelegt, wie die Formatierung von einzelnen Zeichen
  übernommen oder nicht übernommen werden soll. Mögliche Werte sind:
  -ignore_all  -convert_numbering  -convert_all
-->
<option name="page_formatting" value="ignore_all" />
<!--
  Es wird festgelegt, wie die Formatierung von einzelnen Zeichen
  übernommen oder nicht übernommen werden soll. Mögliche Werte sind:
  -ignore_all  -convert_header_footer  -convert_all
-->
<option name="ignore_footnotes_configuration" value="true" />
<!--
  Die Formatierung von 'footnotes' und 'endnotes' wird ignoriert.
-->
<option name="ignore_empty_paragraphs" value="true" />
<!--
  Legt fest, ob leere Absätze ignoriert oder durch einen
  'bigskip' ersetzt werden sollen.
-->
<option name="ignore_hard_page_breaks" value="true" />
<!--
  Es werden 'harte' Seitenumbrüche ignoriert.
-->
<option name="ignore_hard_line_breaks" value="true" />
<!--
  Es werden 'harte' Zeilenumbrüche (shift-Enter) ignoriert.
-->
<option name="ignore_double_spaces" value="true" />
<!--
  Es werden doppelte Leerzeichen ignoriert.
  Ansonsten werden diese durch '\ ' ersetzt.
-->
<option name="no_preamble" value="false" />
<!--
  Legt fest, ob das TeX-Dokument eine Präambel habe soll.
  Bei 'true' wird alles bis einschließlich '\begin{document}'
  und am Ende '\end{document}' weggelassen, so dass das Dokument
  als Teildokument mit 'input' bzw. 'include' eingebunden werden
  kann.
-->
<option name="debug" value="false" />
<!--
  Es wird festgelegt, ob debug-Informationen ausgegeben werden sollen.
-->
<heading-map max-level="5">

```

```

<heading-level-map writer-level="1" name="section" level="1" />
<heading-level-map writer-level="2" name="subsection" level="2" />
<heading-level-map writer-level="3" name="subsubsection" level="3" />
<heading-level-map writer-level="4" name="paragraph" level="4" />
<heading-level-map writer-level="5" name="subparagraph" level="5" />
</heading-map>
<!--
  Legt die entsprechenden Überschriftsstufen fest. Dabei wird hier
  als Erstes 'section', dann 'subsection' ... entsprechend
  der Dokumentklasse verwendet.
-->
<custom-preamble>
  \usepackage[expert]{lucidabr}
  \usepackage{listings}
  \lstloadlanguages{java,HTML}
  \definecolor{source}{gray}{0.92}
  \lstset{extendedchars=true,
    basicstyle=\ttfamily\footnotesize,
    backgroundcolor=\color{source},
    captionpos=b,
    usekeywordsintag=true,
    breaklines=true,
    numbersep=2pt,
    numberstyle=\tiny} % numbers=none / left
</custom-preamble>
<!--
  Legt fest, welche zusätzlichen Pakete, Befehle usw. in das
  TeX-Dokument geschrieben werden.
-->
<style-map name="Listing" class="paragraph-block" next="Listing"
  before="\begin{lstlisting}" after="\end{lstlisting}" />
<style-map name="Listing" class="paragraph"
  before="" after="" verbatim="true" />
<!--
  Es wird der Inhalt der Formatvorlage 'Listing' (kompletter Bereich)
  in eine lstlisting-Umgebung gepackt. Dabei wird nur der reine Inhalt
  (ohne Formatierungen usw.) übernommen.
  Achtung: Dazu muss das Paket 'listings.sty' bei 'custom-preamble'
  angegeben werden.
-->
</config>

```

Beispiel 9.4: Angepasste Konfigurationsdatei

(source/w2l/myconfig.xml)

Nun kann das vorhin verwendete Beispiel 9.2 auf Seite 532 mit der neu erstellten Konfiguration übersetzt werden.

```
w2l -config myconfig.xml 00_Beispiel.sxw 00_Beispiel_myconfig.tex
```

Übersetzt man die TeX-Datei mit *pdftex*, so erhält man ein PDF-Dokument mit zwei Seiten (siehe Abbildung 9.3 auf der nächsten Seite).

**Beispiel:**  
Die ist ein **Beispiel**, wie ein OpenOffice-Dokument nach LaTeX konvertiert wird.

**Tabelle mit Parametern:**

Parameter	Konfigurationsdatei	Beschreibung
-ultraclean	ultraclean.xml	Es werden so gut wie alle Formatierungen entfernt.
-clean	clean.xml	Es werden viele Formatierungen entfernt.
-pdfscreen	pdfscreen.xml	Die Ausgabe wird für eine Bildschirmpräsentation (pdfLaTeX) optimiert.
-pdfprint	pdfprint.xml	Die Ausgabe wird für eine Druckausgabe (pdfLaTeX) optimiert.
-article	article.xml	Es wird die Klasse \LaTeX\article verwendet und es bleiben sehr viele Formatierungen erhalten.

**Bild:**



*Abbildung 1 Blick vom Olympiaturm auf das Fußballstadion*

**Formel:**  
 $y = \ln(x)$

**Aufzählung:**

- Montag
- Dienstag
- Mittwoch

**Dies ist ein Listing**

```
public class A1 {
    public static void main(String[] args) {
        int i = 2;
        while (i <= 100) {
            int k = i * i;
            System.out.println("i = " + i + " k = " + k);
            i += 2; // i=i+2
        }
    }
}
```

Abbildung 9.3: Erzeugtes PDF-Dokument

## Weitere Möglichkeiten

Writer2 $\lambda$ tex bietet aber noch weitere Möglichkeiten.

### Als Plugin für OpenOffice

Writer2 $\lambda$ tex lässt sich auch direkt als Plugin in OpenOffice.org verwenden. Dies bedeutet, OO kann als „ $\lambda$ TeX-Editor“ verwendet werden.

Um w2 $\lambda$  als Plugin zu installieren, geht man wie folgt vor. Wichtig dabei ist, dass man eine vollständige OpenOffice.org (mit Java-Unterstützung) besitzt. Bei debian wird die Java-Unterstützung aus lizenztechnischen Gründen nicht mitgeliefert.

1. Im ersten Schritt wird OpenOffice.org ganz normal installiert. Wir haben hier die Version 1.1.3-DE als Netzinstantiation verwendet.
2. Nun muss in das Installationsverzeichnis gewechselt werden, in dem OO installiert worden ist. Bei Linux ist dies standardmäßig das Verzeichnis `/usr/local/OpenOffice.org1.1.3`, bei Windows ist die Installation unter dem Verzeichnis `c:\Programme` zu finden.

Man wechselt nun in das Unterverzeichnis `<OO>/programme` und ruft dort als Administrator (bei Linux als User `root`) das Programm `jvmsetup` auf, um die Java-Unterstützung zu aktivieren (siehe Abbildung 9.4 auf der nächsten Seite).

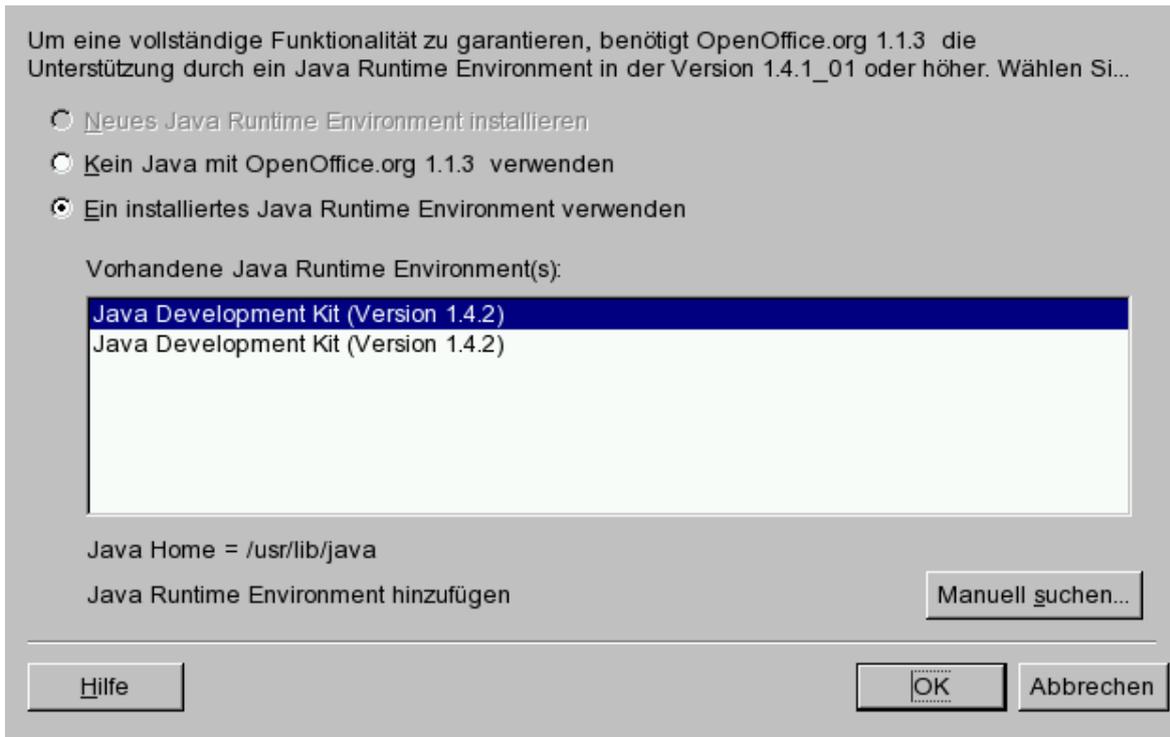


Abbildung 9.4: Java-Erweiterung unter OpenOffice.org

3. Bei der Standardinstallation wird der Filterexport (Mobile Device Filters) normalerweise nicht mitinstalliert. Er lässt sich aber mit dem Setup-Programm leicht nachinstallieren. Dazu wird das Setup-Programm mit dem Parameter `-reinstall` aufgerufen. Hat man zuvor eine Netzinstantallation vorgenommen, muss hier zusätzlich der Parameter `-net` angegeben werden. Danach wird die Auswahl „Filter für mobile Geräte“ aktiviert (siehe hierzu Abbildung 9.5 auf der nächsten Seite). Es wird nun das Paket *xmerge* verwendet, welches `w2l` aufruft.
4. Im nächsten Schritt müssen die entsprechenden `w2l`-Dateien kopiert werden.
  - Die jar-Dateien `writer2latex.jar`, `xmergefix.jar` und die Konfigurationsdatei `writer2latex.xml` (aus dem `w2l`-Verzeichnis) werden in das Unterverzeichnis `<00>/program/classes/` von OpenOffice.org kopiert. Bei Linux muss darauf geachtet werden, dass Leserechte für „alle“ eingerichtet worden sind.
  - Die Datei `xmerge.jar` muss in `oldxmerge.jar` umbenannt werden.
  - Die Datei `xmergefix.jar` wird in `xmerge.jar` umbenannt.
  - Die Datei `w2lfilter.zip` muss in das Unterverzeichnis `<00>/share/uno_packages` kopiert werden.
5. Nun muss OpenOffice.org noch mitgeteilt werden, dass neue Filter vorhanden sind. Dazu wird das Programm `pkgchk` mit dem Parameter `--shared` im Verzeichnis `<00>/programm` aufgerufen. OpenOffice.org darf dabei nicht gestartet sein!

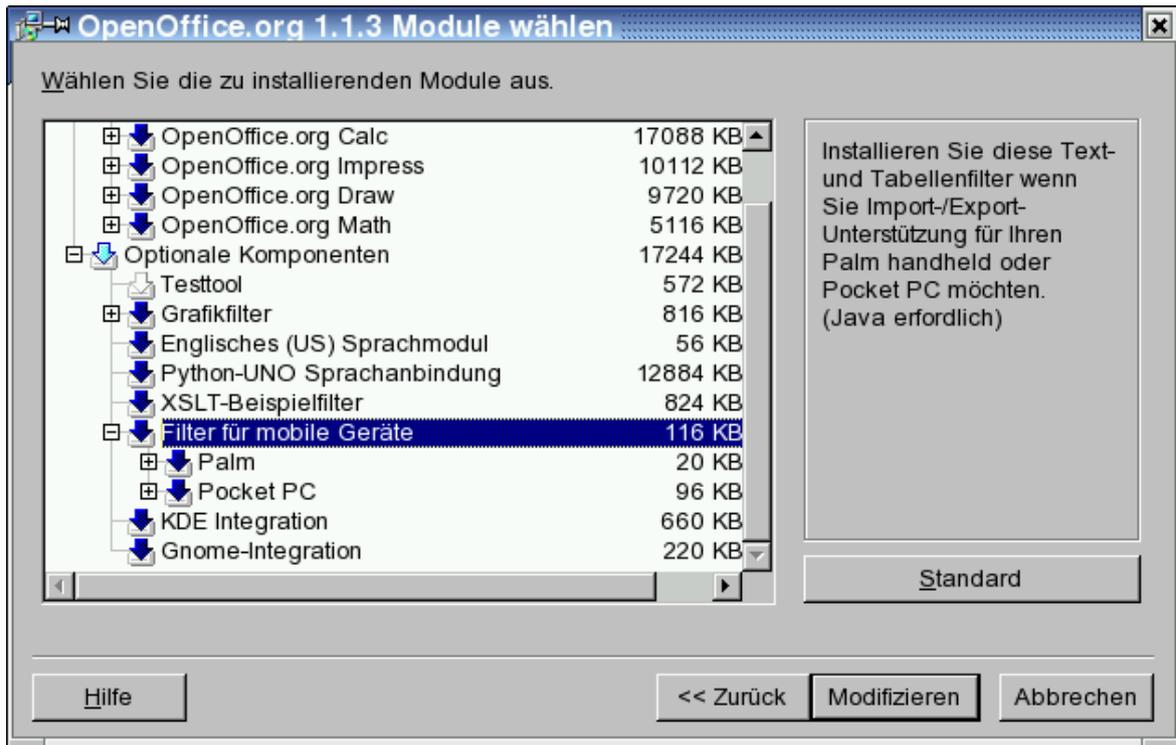


Abbildung 9.5: Filter für mobile Geräte

Nun kann OpenOffice.org gestartet und der Text als  $\LaTeX$ -Datei exportiert werden. Dazu wird unter *Datei* der Menüpunkt *Export* aufgerufen. Unter *Dateiformat* wird dann „LaTeX2e (.tex)“ ausgewählt (siehe Abbildung 9.6 auf der nächsten Seite). Als Ergebnis erhält man dann eine  $\LaTeX$ -Datei.

Standardmäßig verwendet *w2l* die Einstellung, die in der Datei `writer2latex.xml` eingestellt ist. Will man z.B., dass die Konfiguration aus der Datei `clean.xml` verwendet werden soll, so muss diese Datei in das HOME-Verzeichnis mit dem Dateinamen `writer2latex.xml` kopiert werden<sup>4</sup>.

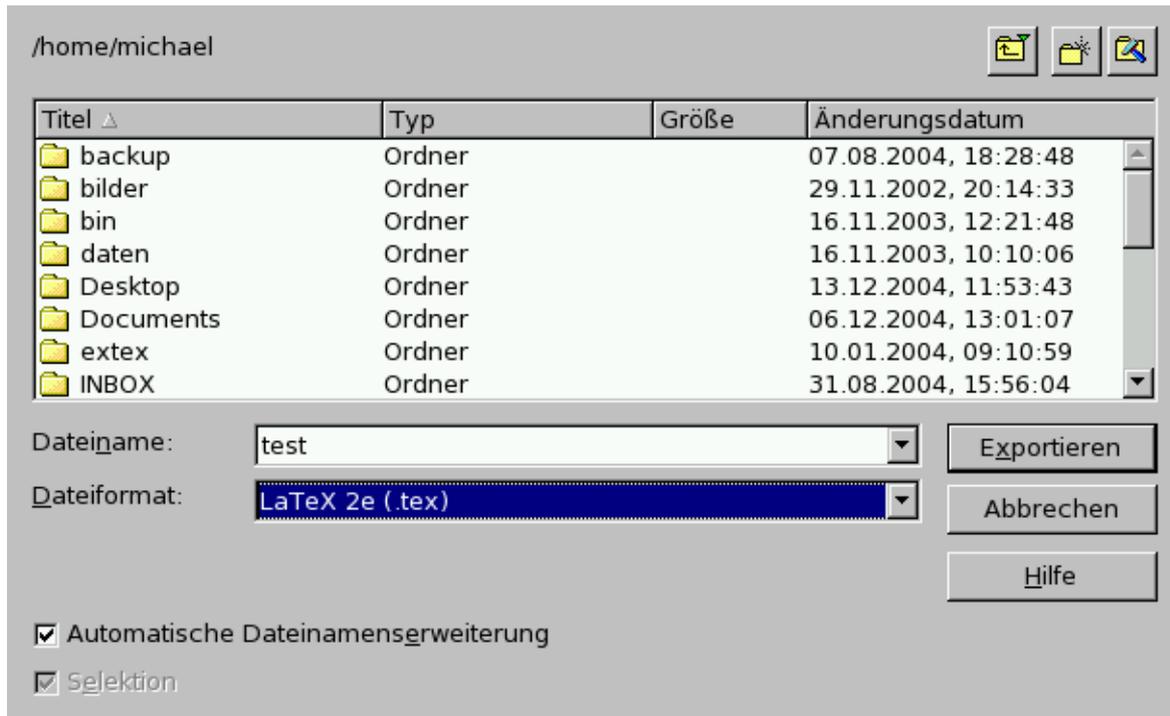
### OpenOffice.org und $\text{bib}\LaTeX$

OpenOffice.org mit *writer2latex* unterstützt auch den Export der Literaturdatenbank im  $\text{bib}\LaTeX$ -Format. Mehr dazu finden Sie in der Dokumentation zu *writer2latex*.

#### Alternativen zu *writer2latex*

Es gibt noch weitere Programme, die aus OpenOffice.org-/Word-Dateien  $\LaTeX$ -Dateien erzeugen. Mehr dazu finden Sie in der FAQ unter <http://www.dante.de/faq/de-tex-faq/html/tools.html#17>.

<sup>4</sup> Der Autor möchte bei der nächsten Version einen entsprechenden Menüpunkt im Export-Fenster einbauen, sodass verschiedene Konfigurationen verwendet werden können.

Abbildung 9.6:  $\LaTeX$ -Export in OpenOffice.org

## 9.2 HTML

Für die Umwandlung von HTML nach  $\LaTeX$  gibt es mehrere Programme. Wir haben hier das perl-Programm *html2latex*<sup>5</sup> getestet. Weitere Programme finden sich in der FAQ unter <http://www.dante.de/faq/de-tex-faq/html/tools.html>.

### Beispiel

Als HTML-Beispiel-Seite haben wir die FAQ-Seite auf dem DANTE-Server verwendet.

Der Aufruf lautet dafür<sup>6</sup>:

```
html2latex -b hr dante-faq-de.html
```

Als Ergebnis erhält man eine  $\LaTeX$ -Datei (siehe Beispiel 9.5).

```
\documentclass[10pt]{article}
\usepackage{fullpage, graphicx, url}
\setlength{\parskip}{1ex}
\setlength{\parindent}{0ex}
\title{DE-TeX-FAQ: German FAQ about TeX and DANTE e.V.}
\begin{document}
\section*{DE-TeX-FAQ -- Fragen und Antworten \textasciitilde{}uber TeX, LaTeX und DANTE e.V.}
.}
```

<sup>5</sup> Siehe <http://html2latex.sourceforge.net/>

<sup>6</sup> Leider hatte das Programm bei unserem Beispiel Probleme mit dem HTML-Tag `hr`, weshalb wir es mit dem Parameter `-b` ausgeschlossen haben.

Dieser Text enth\alt h\aufig gestellte Fragen und passende Antworten zum TeX-System und DANTE e.V.

\begin{quote}

Copyright {\copyright} 1995-2003 Bernd Raichle, Rolf Niepraschk und Thomas Hafner.

Er kann \uber beliebige Medien frei verteilt werden, solange er unver\andert bleibt (inkl. diesem Hinweis). Die Autoren bitten bei Verteilung \uber gedruckte Medien, Datentr\ager wie CD-ROM u.\a. um Zusendung von mindestens drei Belegexemplaren.

\end{quote}

Anregungen, Erg\anzungen, Kommentare und Bemerkungen per E-Mail an detexfaq@dante.de. (Hinweis: Wir beantworten \textbf{keine} Fragen zu und \uber TeX!)

Bernd Raichle, Rolf Niepraschk, Thomas Hafner

Hinweis: Eine Suche in der FAQ ist am einfachsten in der Gesamt-Textversion \uber die Suchfunktion Ihres Browsers m\oglich. Alternativ kann man \uber den Index/das Stichwortverzeichnis (au\ser in der Textversion) die entsprechenden Fragen schnell finden.

\begin{tabular}{ccccc}
Text-Version & HTML-Version \\
(als .tar.gz, als .zip) & DVI-Version & PS-Version \\
(als .gz, als .zip) & PDF-Version
\end{tabular}

Die DE-TeX-FAQ hat folgende Struktur, deren Aufteilung sich aus der Beschr\ankung auf maximal ca. 32 KByte pro Teil ergab:

\begin{itemize}

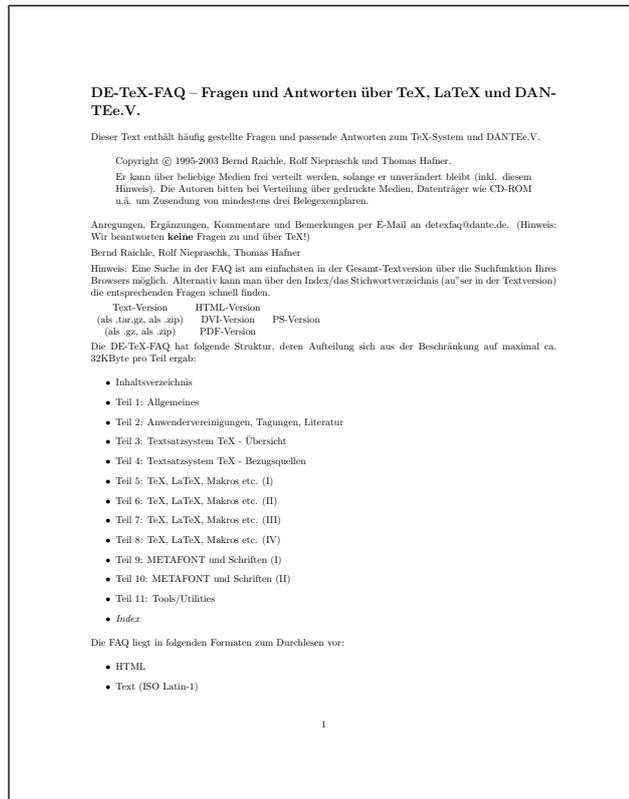
- \item Inhaltsverzeichnis
- \item Teil 1: Allgemeines
- \item Teil 2: Anwendervereinigungen, Tagungen, Literatur
- \item Teil 3: Textsatzsystem TeX - \Ubersicht
- \item Teil 4: Textsatzsystem TeX - Bezugsquellen
- \item Teil 5: TeX, LaTeX, Makros etc. (I)
- \item Teil 6: TeX, LaTeX, Makros etc. (II)
- \item Teil 7: TeX, LaTeX, Makros etc. (III)
- \item Teil 8: TeX, LaTeX, Makros etc. (IV)
- \item Teil 9: METAFONT und Schriften (I)
- \item Teil 10: METAFONT und Schriften (II)
- \item Teil 11: Tools/Utilities
- \item \emph{Index}

`\end{itemize}`

Beispiel 9.5:  $\text{\LaTeX}$ -Ergebnis aus HTML-Datei (Auszug)

Übersetzt man diese Datei mit `pdf $\text{\LaTeX}$` , so sieht die erste Seite wie in Abbildung 9.7 aus.

Abbildung 9.7:  $\text{\LaTeX}$ -Ergebnis aus HTML-Datei



## 9.3 DocBook

DocBook ist ein beliebtes XML-Format, welches für Dokumentationen jeglicher Art entworfen worden ist. DocBook kann mit Hilfe von XSL-Transformationen in verschiedene andere Formate umgewandelt werden.

Für  $\text{\LaTeX}$  sind die XSL-Dateien unter <http://db2latex.sourceforge.net/> zu finden. Diese lassen sich dann mit jedem beliebigen XSL-Prozessor anwenden. Mehr dazu in der entsprechenden Dokumentation.

Alternativ zu XSL kann auch das perl-Programm *dblup* verwendet werden, welches unter <http://dblup.sourceforge.net/> zu finden ist. Allerdings werden zur Zeit nicht alle „Subsets“ von DocBook unterstützt.

## 9.4 Die Notlösung

Wenn man kein Hilfsprogramm gefunden hat, um ein entsprechendes Format in  $\text{\LaTeX}$  umzuwandeln, bleibt noch die Notlösung, eine Textdatei zu verwenden und die entsprechenden  $\text{\LaTeX}$ -Makros von Hand einzugeben.

Die Textdatei kann man mit Hilfe von „cut and paste“ von Hand erstellen oder versuchen, ein Hilfsprogramm zu finden, welches das entsprechende Format in einfachen Text umwandelt.