

# Konferenzmanagement mit $\LaTeX$

Uwe Ziegenhagen

Die Organisation von Veranstaltungen wie Konferenzen, Tagungen oder Workshops erfordert eine ganze Reihe von Rechnungen, Teilnehmerlisten, Namensschildern und anderen Dokumenten.

$\LaTeX$  stellt für alle diese Dokumenttypen geeignete Klassen bereit, über das `datatool` Paket von Nicola Talbot kann man ihnen den Zugriff auf CSV-Dateien ermöglichen.

In diesem Artikel beschreibe ich anhand eines praktischen Beispiels die genutzten Pakete und ihr Zusammenwirken.

## Einleitung

Ausgangspunkt der Betrachtungen sei eine fiktive Tagung, für deren Organisation die folgenden Dokumente benötigt werden:

- aktuelle Teilnehmerlisten mit dem Zahlungsstatus der Konferenzgebühr
- ein Tagungsband, der die einzelnen Beiträge der Teilnehmer enthält
- Rechnungen und Teilnahmebestätigungen
- Namensschilder

Die Grundlage für die Erstellung nahezu all dieser Dokumente bildet eine Excel bzw. OpenOffice Tabelle, die dann über den Umweg einer CSV-Datei von  $\LaTeX$  verarbeitet wird.

## Das `datatool`-Paket

Das `datatool` Paket [2] von Nicola Talbot, das ihr `cvstools` Paket [1] ablöst, stellt eine einfache Schnittstelle zur Verarbeitung von CSV-Dateien (Comma-Separated Values) bereit. Das Paket besteht eigentlich aus mehreren Style-Dateien, in diesem Artikel liegt der Fokus jedoch auf `datatool.sty`, für die anderen Teile sei auf die sehr gute Dokumentation des Pakets verwiesen.

Tabelle 1: Bestandteile des `Datatool` Pakets

<code>datatool.sty</code>	erstellt und exportiert Daten-Dateien, importiert extern erstellte Daten-Dateien
<code>datapie.sty</code>	erstellt Kuchen-Diagramme (pie charts)
<code>dataplot.sty</code>	generiert zweidimensionale Punkt- oder Liniendiagramme
<code>databar.sty</code>	erstellt Balkendiagramme
<code>databib.sty</code>	konvertiert BibTeX Bibliographien in Daten-Dateien

```

1  vorname,nachname,strasse,ort,zuzahlen,bezahlt
2  Nicole,Möller,Schillerplatz 61,18419 Vogelow,100.00,100.00
3  Tom,Lehmann,Nachtigallgasse 11,29098 Altaue,100.00,100.00
4  Tim,Wagner,Amselplatz 92,46917 Langenhausen,100.00,0.00
5  Moritz,Müller,Waldallee 71,55348 Kirchstein,100.00,0.00
6  Susi,Mayer,Sonnenweg 27a,83675 Heidehausen,100.00,100.00
7  Ines,Mayer,Wasserallee 83a,26118 Kirchfurt,100.00,100.00
8  Uwe,Meier,Sonnenplatz 7,07514 Vogelburg,100.00,0.00
9  Mandy,Berger,Goetheweg 25,03783 Wolfental,100.00,100.00
10 Tim,Grünwald,Wiesenplatz 9a,90778 Moosow,100.00,50.00
11 Jenny,Köster,Finkenallee 29c,53522 Wiesenow,100.00,100.00
12 Marko,Mayer,Amselweg 11c,32108 Grünstein,100.00,100.00
13 Jenny,Berger,Wiesenallee 82,72044 Moosau,0.00,0.00

```

Listing 1: Datenbankdatei `datad.csv`

Listing 2 zeigt ein minimales Beispiel, das die Vor- und Nachnamen der Teilnehmer in einer Liste (siehe Tabelle 2) ausgibt. Die Daten werden aus der Datei `datad.csv` entnommen, siehe Listing 1.

Mit `\DTLloaddb{list}{datad.csv}` werden der Name der Datenquelle und die Quelldatei der Datenbank festgelegt, hier `list` als Name für die Datei `datad.csv`.

Mit `\DTLforeach{arg1}{arg2}{arg3}` wird spezifiziert, was mit den aus der Datenbank gelesenen Datensätzen geschehen soll. `arg1` spezifiziert den Namen der Datenquelle, `arg2` nimmt die Zuordnung der Spalten zu den entsprechenden  $\LaTeX$ -Befehlen vor und `arg3` legt die Ausgabe fest.

```

1 \documentclass{article}
2 \usepackage{datatool}
3 \usepackage[latin1]{inputenc}
4 \begin{document}
5 \DTLloaddb{list}{datad.csv}
6
7 \begin{tabular}{ll}
8 \bfseries Vorname & \bfseries Nachname
9 \DTLforeach{list}{\first=vorname,\last=nachname}{%
10 \\ \first & \last }
11 \end{tabular}
12
13 \end{document}

```

Listing 2: `listed.tex`, Quellcode für Tabelle 2

Vorname	Nachname
Nicole	Möller
Tom	Lehmann
Tim	Wagner
Moritz	Müller
Susi	Mayer
Ines	Mayer

Tabelle 2: `listed.pdf`, Ausgabe von Listing 2 (Auszug)

Listing 3 enthält die notwendigen Befehle, um alle Spalten der Datenbank auszuwerten und führt zwei neue Befehle ein, `\DTLsort` und `\DTLsumforkeys`.

`\DTLsort{key}{database}` sortiert die Einträge der Datenbank anhand des Parameters `key`. Es können auch mehrere Werte an `key` übergeben werden, dies ist sinnvoll wenn gleiche Nachnamen in der Datenbank auftreten und deshalb zusätzlich nach dem Vornamen sortiert werden soll.

Die Version des Befehls ohne Stern `*` berücksichtigt Groß- und Kleinschreibung (Großbuchstaben vor Kleinbuchstaben), die Version mit Stern ignoriert diese beim Vergleich von Zeichenketten. Der Befehl akzeptiert als optionales Argument eine Liste von Schlüsseln, auf die zurückgegriffen wird, wenn ein Eintrag unzureichend ist.

`\DTLsumforkeys{database}{key}{\command}` summiert die Werte aller `key` Felder und speichert diese im Befehl `\command`.

Mit diesen beiden Befehlen können wir jetzt die Teilnehmerliste nach dem Nachnamen sortieren und die erwarteten und eingegangenen Zahlungen summieren.

```

1  \documentclass{article}
2  \usepackage{datatool}
3  \usepackage[latin1]{inputenc}
4  \usepackage{eurosym}
5  \begin{document}
6
7  \DTLloaddb{list}{datad.csv}
8  \DTLsort{nachname}{list}
9  \DTLsumforkeys{list}{zuzahlen}{\soll}
10 \DTLsumforkeys{list}{bezahlt}{\haben}
11
12 \begin{tabular}{rllrr}
13 Nr. & \bfseries Vorname & \bfseries Nachname &
14       & \bfseries Gebühr & \bfseries Zahlung
15 \DTLforeach{list}{\first=vorname,\last=nachname,
16               \fee=zuzahlen, \paid=bezahlt}{%
17   \DTLiffirstrow{\ \hline}{\}%
18   \theDTLrowi & \first & \last & \fee\,\euro & \paid\,\euro
19 }
20 \end{tabular}
21
22 \soll\,\euro{} sind zu bezahlen, \haben\,\euro{} sind bezahlt.
23 \end{document}

```

Listing 3: `liste2d.tex`, Quellcode für Tabelle 3

Um besser zwischen einzelnen Zeilen unterscheiden zu können, kann das `colortbl` Paket zusammen mit dem `\DTLifoddrrow` Befehl genutzt werden. Listing 4 zeigt den Quellcode, der für Tabelle 4 benötigt wird.



Tabelle 4: Ausgabe von Listing 4

ID	Vorname	Nachname	Gebühr	Zahlung
1	Mandy	Berger	100,00€	100,00€
2	Jenny	Berger	0,00€	0,00€
3	Tim	Grünwald	100,00€	50,00€
4	Jenny	Köster	100,00€	100,00€
5	Tom	Lehmann	100,00€	100,00€
6	Susi	Mayer	100,00€	100,00€
7	Ines	Mayer	100,00€	100,00€
8	Marko	Mayer	100,00€	100,00€
9	Uwe	Meier	100,00€	0,00€
10	Nicole	Möller	100,00€	100,00€
11	Moritz	Müller	100,00€	0,00€
12	Tim	Wagner	100,00€	0,00€

1,100€ sind zu bezahlen, 750€ sind bezahlt.

Für andere CSV-Formate als das englische können wir `datatool` Parameter Achtung für Spaltentrenner und Dezimalzeichen übergeben.

dieser  
Teil  
war im  
DTK  
Artikel  
fehler-  
haft.

Listing 5 zeigt die CSV-Datei, exportiert aus einer deutschen Version von Excel, Listing 6 den L<sup>A</sup>T<sub>E</sub>X Code, Tabelle 5 die generierte Ausgabe. Das Listing führt einen neuen Befehl ein, einer Erklärung bedarf:

- `\DTLdecimaltocurrency{number}{command}` konvertiert eine Dezimalzahl in das lokale Format

```

1 vorname;nachname;strasse;ort;zuzahlen;bezahlt
2 Nicole;Möller;Schillerplatz 61;18419 Vogelow;100,000;100,00
3 Tom;Lehmann;Nachtigallgasse 11;29098 Altaue;100,00;100,00,
4 Tim;Grünwald;Wiesenplatz 9a;90778 Moosow;100,00;50,00
5 Jenny;Köster;Finkenallee 29c;53522 Wiesenow;100,00;100,00
6 Marko;Mayer;Amselweg 11c;32108 Grünstein;100,00;100,00
7 Jenny;Berger;Wiesenallee 82;72044 Moosau;0,00;0,00

```

Listing 5: Die Datenquelle `data2d.csv` mit deutscher Formatierung

```

1 \documentclass[10pt]{scrartcl}
2 \usepackage[latin1]{inputenc}
3 \usepackage[landscape,a4paper]{geometry}
4 \usepackage{datatool,eurosym}
5 \usepackage[ngerman]{babel}
6 \begin{document}
7
8 \section*{Teilnehmer, Stand \today}
9 \DTLsetseparator{;} \DTLsetnumberchars{}{,}

```



## Briefe und Rechnungen

Für formelle Briefe und Rechnungen gibt es die `g-brief` Klasse und die `scr1ttr2`, mit deren Hilfe sich den deutschen Formvorschriften entsprechende Dokumente erstellen lassen.

Persönlich nutze ich `scr1ttr2`, da sich mit dieser Dokumentenklasse auch komplexe Layouts gut umsetzen lassen, sowie ein selbstentwickeltes Paket, `varsfromjobname.sty`. Mit diesem Paket [3] ist es möglich, einzelne Elemente aus dem Namen der L<sup>A</sup>T<sub>E</sub>X-Datei zu extrahieren und in Variablen abzulegen, die dann im Dokument gesetzt werden können.

Dazu erwartet das Paket den Dateinamen in der Form `arg1-arg2-....tex`, insgesamt werden bis zu neun Argumente unterstützt, die dann über `\getfromjobname{argx}` bzw. über die Befehle `\getonefromjobname` bis `\getninefromjobname` zur Verfügung gestellt werden. Auf diese Weise ist es zum Beispiel möglich, Daten wie Rechnungsdatum und Rechnungsnummer fest im Dateinamen einzubetten.

Für die Erstellung von Serienbriefen gibt es mehrere Möglichkeiten. Das `KOMA` Paket selbst verfügt über die Möglichkeit, mit Adressdateien umzugehen, allerdings müssen diese in einem bestimmten Format vorhanden sein, das sich vom für `datatools` genutzten CSV-Format unterscheidet.

Um nur eine Datenquelle pflegen zu müssen, nutze ich daher auch das `datatool` Paket. Listing 7 zeigt ein Beispiel für die Erstellung eines Serienbriefes mit `datatool`.

Die `letter` Umgebung wird dabei in den Schleifen-Teil des `\DTLforeach` gesetzt, der Adressparameter für den Brief wird dann mit `{\first \last \address \\town}` gefüllt.

```

1 \documentclass[a5paper]{scr1ttr2}
2 \usepackage[english]{babel}
3 \usepackage[latin1]{inputenc}
4 \usepackage{datatool}
5 \setkomavar{title}{Teilnahmebestätigung}
6
7 \begin{document}
8 \DTLloaddb{list}{datad.csv}
9 \DTLforeach{list}{\first=vorname,\last=nachname,\address=strasse,
10 \town=ort,\fee=zuzahlen,\paid=bezahlt}{% Beginn Schleife
11

```

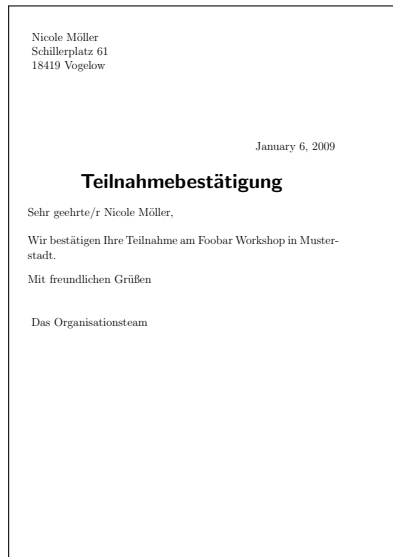
```

12 \begin{letter}{\first~\last \address \town}%
13 \opening{Sehr geehrte/r \first~\last,}%
14
15 Wir bestätigen Ihre Teilnahme am FooBar
16 Workshop in Musterstadt.
17
18 \setkomavar{fromname}{Das Organisationsteam}%
19 \closing{Mit freundlichen Grüßen}%
20 \end{letter}
21 }% Ende Schleife
22 \end{document}

```

Listing 7: `scrlettsample.tex`, Quelltext für Abbildung 1

Abbildung 1: Ausgabe von Listing 7



Dies ist nur ein einfaches Beispiel, über die zahlreichen weiteren `datatool` Befehle sind noch viel detailliertere Briefe möglich. So lässt sich beispielsweise über ein zusätzliches Feld `Geschlecht` in der Datendatei, das mit 'm' oder 'w' gefüllt ist, und das Kommando `\DTLifstringeq` eine geschlechtsspezifische Anrede erstellen.

## Namensschilder mit `ticket.sty`

Mit `ticket.sty` lassen sich Namensschilder, Adressaufkleber und ähnliche Dokumente einfach setzen. Listing 8 zeigt den Aufbau einer entsprechenden Datei.

Nach dem Laden diverser Pakete wird das standardmäßig definierte Ticket zurückgesetzt, damit die eigene Definition erfolgen kann. Die Kommandos zwischen `\makeatletter` und `\makeatother` setzen bzw. unterdrücken Rahmen, Schnitt- und Falzmarken. Der Parameter `\@boxedtrue` ist sehr nützlich während der Entwurfsphase der Etiketten, vor dem Druck sollte er auf `\@boxedfalse` gesetzt werden.

Der Parameter `badges` beim Laden von `ticket.sty` verweist auf die Datei `badges.tdf` (siehe Listing 9), in der die Anzahl und Größe des einzelnen Schilder sowie Angaben über den druckerspezifischen horizontalen und vertikalen Offset stehen, also den Abstand des ersten Tickets vom oberen und linken Rand. Die Angabe des Offset ist notwendig, da die Ränder über das `geometry` Pakets auf null gesetzt sind.

Der Befehl `\mylabel` wird anschließend definiert, um eine bequeme Schnittstelle zu den einzelnen Namensschildern zu haben, die nur noch Parameter für Name und Ort benötigt.

```

1 \documentclass[a4paper,12pt]{letter}
2 \usepackage[total={210mm,297mm},top=0mm, %
3     left=0mm, includefoot]{geometry}
4 \usepackage[badges]{ticket}
5 \usepackage{graphicx,palatino}
6 \usepackage[latin1]{inputenc}
7 \usepackage{xcolor}
8
9 \renewcommand{\ticketdefault}{}%
10 \makeatletter
11 \@boxedfalse % Rahmen um Ticket
12 \@emptycrossmarkfalse % Falzmarken
13 \@cutmarktrue % Schnittmarken
14 \makeatother
15
16 \newcommand{\mylabel}[2]{\ticket{%
17     \put(7,35){\scalebox{2}{\textbf{#1}}}
18     \put(7,25){\scalebox{1.5}{\textbf{#2}}}
19     \put(7,5){\scalebox{1}{\textcolor{gray}{%

```

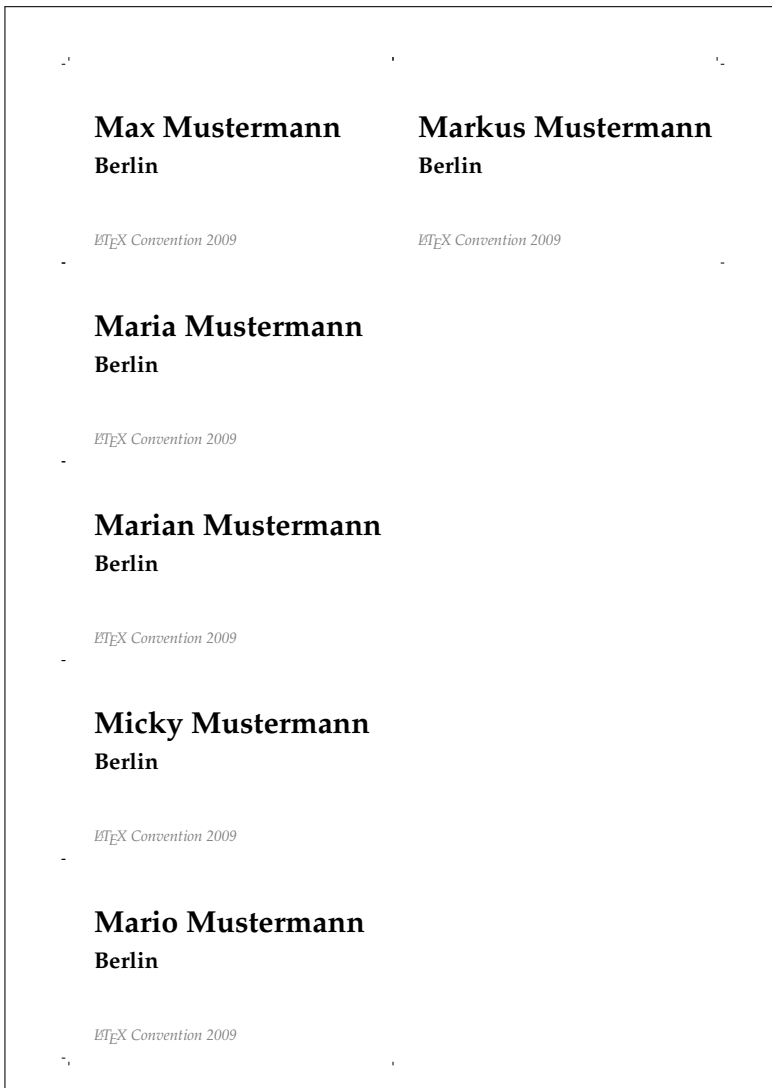
```
20 \textit{\LaTeX{} Convention 2009}}}}
21 }}
22
23 \begin{document}
24 \mylabel{Max Mustermann}{Berlin}
25 \mylabel{Maria Mustermann}{Berlin}
26 \mylabel{Marian Mustermann}{Berlin}
27 \mylabel{Micky Mustermann}{Berlin}
28 \mylabel{Mario Mustermann}{Berlin}
29 \mylabel{Markus Mustermann}{Berlin}
30 \end{document}
```

Listing 8: Quelltext für Abbildung 2

```
1 \unitlength=1mm
2 \hoffset=-10mm
3 \voffset=-16mm
4 \ticketNumbers{2}{5}
5 \ticketSize{90}{55.2} % unitlength => mm
6 \ticketDistance{0}{0} % unitlength => mm
```

Listing 9: `badges.tdf`, Definition des Papierbogens `badges.tdf`

Abbildung 2: Ausgabe von Listing 8



Natürlich ist auch hier das Ziel, die Namensschilder durch `datatools` automatisch setzen zu lassen, Listing 10 zeigt die entsprechend angepasste Datei mit den entsprechenden `datatool` Befehlen.

```

1 \documentclass[a4paper,12pt]{letter}
2 \usepackage[total={210mm,297mm},top=0mm,
3     left=0mm, includefoot]{geometry}
4 \usepackage[badges]{ticket}
5 \usepackage[graphicx,palatino,marvosym]
6 \usepackage[latin1]{inputenc}
7 \usepackage{xcolor}
8 \usepackage{datatool}
9
10 \renewcommand{\ticketdefault}{}%
11 \makeatletter
12   \@emptycrossmarkfalse
13   \@cutmarkfalse
14   \@boxedtrue
15 \makeatother
16
17 \newcommand{\mylabel}[2]{\ticket{%
18   \put(7,35){\scalebox{2}{\textbf{#1}}}
19   \put(7,25){\scalebox{1.5}{\textbf{#2}}}
20   \put(7,5){\scalebox{1}{\textcolor{gray}{%
21     \textit{\LaTeX} Convention 2009}}}}
22 }}
23
24 \begin{document}
25
26 \DTLloaddb{list}{datad.csv}
27 \DTLforeach{list}{\first=vorname,\last=nachname,
28 \ort=ort,\fee=zuzahlen,\paid=bezahlt}
29 {%
30 \mylabel{\first \last}{\ort}
31 }
32
33 \end{document}

```

Listing 10: `badgesd.tex`, Quelltext für Abbildung 3

Abbildung 3: Ausgabe von Listing 10

<p><b>NicoleMöller</b> 18419 Vogelow</p> <p><small>IT&amp;X Convention 2009</small></p>	<p><b>InesMayer</b> 26118 Kirchfurt</p> <p><small>IT&amp;X Convention 2009</small></p>
<p><b>TomLehmann</b> 29098 Altaue</p> <p><small>IT&amp;X Convention 2009</small></p>	<p><b>UweMeier</b> 07514 Vogelburg</p> <p><small>IT&amp;X Convention 2009</small></p>
<p><b>TimWagner</b> 46917 Langenhausen</p> <p><small>IT&amp;X Convention 2009</small></p>	<p><b>MandyBerger</b> 03783 Wolfental</p> <p><small>IT&amp;X Convention 2009</small></p>
<p><b>MoritzMüller</b> 55348 Kirchstein</p> <p><small>IT&amp;X Convention 2009</small></p>	<p><b>TimGrünwald</b> 90778 Moosow</p> <p><small>IT&amp;X Convention 2009</small></p>
<p><b>SusiMayer</b> 83675 Heidehausen</p> <p><small>IT&amp;X Convention 2009</small></p>	<p><b>JennyKöster</b> 53522 Wiesenow</p> <p><small>IT&amp;X Convention 2009</small></p>

## Dokumente zusammenführen mit `combine.sty`

Bei größeren Workshops oder Tagungen ist es oft üblich, dass die Beiträge der Teilnehmer in Tagungsbänden, den Proceedings, gesammelt werden. Die Hauptarbeit der Editoren besteht dann darin, die einzelnen Beiträge zu einem Dokument zusammenzusetzen, was sehr aufwändig sein kann.

Als Alternative zu dieser manuellen Integration läßt sich das Paket `combine.sty` nutzen, das eigenständig kompilierbare Dokumente zu einem Text zusammenbaut.

Listing 11 zeigt einen minimalen Beitrag eines Teilnehmers, der -zusammen mit gleichartigen Dateien- zu einem einzigen Dokument zusammengesetzt werden soll.

Eine einfache `combine` Datei, die zwei dieser Einzelbeiträge in eine Datei kompiliert und ein Inhaltsverzeichnis ablegt, ist in Listing 12 abgedruckt. Lokal in den Dateien geladene Pakete werden in dieses Hauptdokument übernom-

men, daher empfiehlt es sich, eine für die Teilnehmer verpflichtende Vorlage zu erstellen und keine weiteren Pakete zuzulassen.

```

1 %% LaTeX2e file 'import1.tex'
2 %% generated by the 'filecontents' environment
3 %% from source 'combined' on 2009/02/16.
4 %%
5 \documentclass{article}
6
7 \author{Leonardo da Vinci}
8 \title{Inventionen}
9
10 \begin{document}
11 \maketitle
12
13 Text des ersten Artikels.
14
15 \end{document}

```

Listing 11: `import1.tex`, Quelltext eines zu importierenden Dokuments

```

1 \documentclass[a5paper]{combine}
2 \pagestyle{combine}
3 \begin{document}
4
5 \tableofcontents
6 \begin{papers}
7   \coltoctitle{Anmerkungen zum Euro} % Erster Titel für ToC
8   \coltocauthor{Max Mustermann} % Erster Autor für ToC
9   \label{import1}
10  \import{import1}
11
12  \coltoctitle{Anmerkungen zur D-Mark} % Zweiter Titel für ToC
13  \coltocauthor{Maria Mustermann} % Zweiter Autor für ToC
14  \label{import2}
15  \import{import2}
16 \end{papers}
17 \end{document}

```

Listing 12: `combineit.tex`, Quelltext eines `combine` Dokuments

## Zusammenfassung

Von den Möglichkeiten, die L<sup>A</sup>T<sub>E</sub>X im Satz von dynamischen Dokumenten bietet, konnte dieser Artikel nur einige wenige zeigen. Insbesondere der Satz von XML-basierten Dokumenten oder Inhalten, die nur per Datenbanksystem zugänglich sind, bietet spannende Projekte im Satz mit L<sup>A</sup>T<sub>E</sub>X und wird eventuell noch in einer der nächsten Ausgaben nachgereicht.

Für Kommentare, Hinweise und Anmerkungen bin ich jederzeit dankbar, am einfachsten erreicht man mich über das Kontaktformular meiner Homepage <http://www.uweziegenhagen.de>. Dort werden auch die genutzten Dateien und Errata für diesen Artikel bereitgestellt.

## Literatur

- [1] Nicola Talbot: *CSV Tools*; 2007; CTAN: `tex-archive/macros/latex/contrib/csvtools`.
- [2] Nicola Talbot: *datatool v 1.01: Databases and data manipulation*; 2007; CTAN: `tex-archive/macros/latex/contrib/datatool`.
- [3] Uwe Ziegenhagen: *varsfromjobname v 0.5*; 2009; CTAN: `tex-archive/macros/latex/contrib/varsfromjobnames`.