

Tipps zu L<sup>A</sup>T<sub>E</sub>X

# Das L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>-Sündenregister oder Veraltete Befehle, Pakete und andere Fehler

Mark Trettin\*

Version 1.6 vom 1. Oktober 2003

## Zusammenfassung

Angeregt durch eine Diskussion in der deutschsprachigen T<sub>E</sub>X-Newsgruppe<sup>1</sup> über das wiederholte Auftauchen von veralteten und „schlechten“ Paketen und Befehlen, habe ich mich entschlossen, diese kleine Übersicht zu schreiben.

Ich versuche in diesem Artikel die gängigsten Fehler zu zeigen und Alternativen anzubieten. Diese Übersicht soll weder Einführungen wie l2kurz [7] noch die De-TeX-FAQ [4] ersetzen, sondern lediglich einen kleinen Überblick bieten.

Für Vorschläge, Verbesserungen und Kommentare bin ich dankbar. Ach ja, bevor Anfragen kommen: Ja, ich habe Times/Helvetica<sup>2</sup>/Courier benutzt, allerdings nur um die Datei möglichst klein zu halten. ; - )

Copyright © 2003 by Mark Trettin.

This material may be distributed only subject to the terms and conditions set forth in the *Open Publication License*, v1.0 or later (the latest version is presently available at <http://www.opencontent.org/openpub/>).

Ich bedanke mich bei Christoph Bier, Christian Faulhammer, Jürgen Fenn<sup>3</sup>, Yvonne Hoffmüller, David Kastrup, Markus Kohm, Thomas Lotze, Frank Mittelbach, Heiko Oberdiek, Walter Schmidt, Stefan Stoll und Reinhard Zierke für Tipps, Anmerkungen und Korrekturen. Falls ich jemanden vergessen haben sollte, bitte ich um eine Mail.

---

\*E-Mail: [Mark.Trettin@gmx.de](mailto:Mark.Trettin@gmx.de)

<sup>1</sup>[de.comp.text.tex](http://de.comp.text.tex)

<sup>2</sup>Arial in der Darstellung des Acrobat Reader

<sup>3</sup>Englische Übersetzung: l2tabuen.

## Inhaltsverzeichnis

<b>1</b>	<b>„Todsünden“</b>	<b>3</b>
1.1	<i>a4.sty</i> , <i>a4wide.sty</i> . . . . .	3
1.2	Layoutänderungen . . . . .	3
1.3	Änderungen von Paketen und Klassen . . . . .	3
1.4	Änderung des Zeilenabstandes mittels <code>\baselinestretch</code> . . . . .	3
1.5	Absatzeinzug und -abstand ( <code>\parindent</code> , <code>\parskip</code> ) . . . . .	4
1.6	Abgesetzte Formeln mit <code>\$\$...\$\$</code> . . . . .	4
1.7	<code>\def</code> vs. <code>\newcommand</code> . . . . .	5
1.8	Verwendung von <code>\sloppy</code> . . . . .	5
<b>2</b>	<b>Obsoletes</b>	<b>6</b>
2.1	Befehle . . . . .	6
2.1.1	Änderung des Schriftstils . . . . .	6
2.1.2	Mathematische Brüche ( <code>\over</code> vs. <code>\frac</code> ) . . . . .	7
2.1.3	Zentrierung mit <code>\centerline</code> . . . . .	7
2.2	Klassen und Pakete . . . . .	7
2.2.1	<i>scrlettr.cls</i> vs. <i>scrlettr2.cls</i> . . . . .	7
2.2.2	<i>epsf.sty</i> , <i>psfig.sty</i> , <i>epsfig.sty</i> vs. <i>graphics.sty</i> , <i>graphicx.sty</i> . . . . .	8
2.2.3	<i>doubleSPACE.sty</i> vs. <i>setspace.sty</i> . . . . .	8
2.2.4	<i>fancyheadings.sty</i> , <i>scrpage.sty</i> vs. <i>fancyhdr.sty</i> , <i>scrpage2.sty</i> . . . . .	8
2.2.5	<i>isolatin.sty</i> , <i>umlaut.sty</i> vs. <i>inputenc.sty</i> . . . . .	8
2.2.6	<i>t1enc.sty</i> vs. <i>fontenc.sty</i> . . . . .	10
2.2.7	<i>natdin.bst</i> vs. <i>dinat.bst</i> . . . . .	10
2.3	Schriften . . . . .	10
2.3.1	<i>times.sty</i> . . . . .	10
2.3.2	<i>mathptm.sty</i> . . . . .	10
2.3.3	<i>pslatex.sty</i> . . . . .	10
2.3.4	<i>palatino.sty</i> . . . . .	11
2.3.5	<i>mathppl.sty</i> . . . . .	11
2.3.6	Aufrechte griechische Buchstaben . . . . .	11
<b>3</b>	<b>Verschiedenes</b>	<b>13</b>
3.1	Gleitumgebungen – „figure“, „table“ . . . . .	13
3.2	Der Anhang . . . . .	13
3.3	Mathematiksatz . . . . .	13
3.4	Die <code>\*name</code> -Makros . . . . .	14

## 1 „Todsünden“

In diesem Abschnitt habe ich die wohl schlimmsten Fehler zusammengetragen, die in schöner Regelmäßigkeit in `de.comp.text.tex` auftauchen und den dortigen Regulars entweder die Zornesröte ins Gesicht oder die Tränen in die Augen treiben. ; - )

### 1.1 *a4.sty*, *a4wide.sty*

Diese „beiden“ Pakete sollten nicht mehr verwendet und ersatzlos aus dem L<sup>A</sup>T<sub>E</sub>X-Quelltext gestrichen und durch die Klassenoption `a4paper` ersetzt werden. Abgesehen davon, dass das Layout der meisten dieser Pakete typografisch mehr als fragwürdig ist, existieren mehrere verschiedene, zu einander inkompatible Versionen. Man kann sich also nicht einmal sicher sein, dass auf einem anderen Rechner das Dokument gleich (schlecht?) aussieht.

### 1.2 Layoutänderungen

Die von den Standardklassen (*article.cls*, *report.cls*, *book.cls*) verwendeten Satzspiegel wirken häufig zu groß. Abhilfe bieten die entsprechenden Klassen (*scrartcl.cls*, *scrreprt.cls*, *scrbook.cls*) aus dem KOMA-Script-Paket, oder das ebenfalls dort enthaltene *typearea.sty*. Die dazugehörige Dokumentation *scrguide* [3] enthält viele weiterführende Informationen.

Wenn man wirklich einen anderen Satzspiegel als den von z. B. *typearea.sty* erzeugten benötigt, dann sollte man bitte die Pakete *geometry.sty* oder *vmargin.sty* verwenden und nicht versuchen, „zu Fuß“ `\oddsidemargin` & Co. zu ändern.

Unter gar keinen Umständen sollte man an `\hoffset` bzw. `\voffset` herumfummeln, außer man kennt sich sehr gut mit den T<sub>E</sub>X-Internas aus.

### 1.3 Änderungen von Paketen und Klassen

Niemals Dokumentklassen (*article.cls*, *scrbook.cls* usw.) oder Stylefiles (*varioref.sty*, *color.sty*) direkt ändern! Entweder man bastelt sich Containerklassen bzw. -styles oder, man *kopiert* die Klassen/Styles, ändert die Kopie und speichert diese unter *anderem* Namen ab.

Ein Beispiel zur Erstellung von Containerklassen findet sich in der FAQ [4, Punkt 5.1.5].

**Hinweis** Generell sollte man solche zusätzlich installierte Klassen und Pakete entweder in den lokalen oder den `$HOME`-T<sub>E</sub>X-Baum speichern, damit bei einem Upgrade der T<sub>E</sub>X-Distribution diese Änderungen nicht überschrieben werden. Braucht man diese Änderungen nur in dem speziellen Projekt und möchte es weitergeben, dann könnte man die angepasste Klasse auch im aktuellen Projektverzeichnis speichern.

### 1.4 Änderung des Zeilenabstandes mittels `\baselinestretch`

Wenn man einen größeren Durchschuss benötigt (z. B. eineinhalbzeilig oder zweizeilig), dann bietet das Paket *setspace.sty* die einfachste Möglichkeit. Möchte man hingegen nur kleine Anpassungen für andere Schriften als Computer Modern (z. B. Palatino) machen, dann kann bzw.

sollte man `\linespread{<Faktor>}` verwenden. Bei der Schrift Palatino würde sich zum Beispiel `\linespread{1.05}` anbieten.

## 1.5 Absatzeinzug und -abstand (`\parindent`, `\parskip`)

Den Absatzeinzug (`\parindent`) zu ändern, kann manchmal sinnvoll sein. Man sollte dabei aber beachten, dass

- man den Einzug mit einer schriftabhängigen Größe (em) und nicht mit einer absoluten Größe (mm) ändert. „Schriftabhängig“ bedeutet hier *nicht*, dass sich der Absatzeinzug bei Schriftgrößenänderungen automatisch anpasst, sondern dass der Wert der aktuell aktiven Schrift benutzt wird.
- man  $\LaTeX$ -Syntax verwendet, da diese die wenigsten Probleme birgt. Zum Beispiel ist sie für externe Programme/Skripte leichter zu parsen<sup>4</sup>, für den Benutzer besser zu warten und es gibt keine Inkompatibilitäten mit anderen Paketen (*calc.sty*).

Ersetze: `\parindent=1em` durch `\setlength{\parindent}{1em}`

Wenn man hingegen *keinen* Absatzeinzug, dafür aber einen zusätzlichen Durchschuss als Absatzkennzeichnung wünscht, sollte man *nicht* einfach

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{\baselineskip}
```

verwenden. Da sich `\parskip` auch auf Listen, Verzeichnisse und Überschriften auswirkt, ist diese Vorgehensweise nicht zu empfehlen.

Das Paket *parskip.sty* und die KOMA-Script-Klassen treiben einigen Aufwand, um diese Begleiterscheinungen zu vermeiden. Zur Verwendung der verschiedenen Optionen (`parskip`, `halfparskip` usw.) von KOMA-Script siehe den *scrguide* [3]. Wenn eine der KOMA-Script-Klassen verwendet wird, muss das Paket *parskip.sty* *nicht* noch zusätzlich geladen werden.

## 1.6 Abgesetzte Formeln mit `$$ . . . $$`

Bitte nicht! `$$ . . . $$` ist ein `plainTeX`-Befehl und sollte in  $\LaTeX$  vermieden werden, da dadurch die vertikalen Abstände bei abgesetzten Formeln inkonsistent werden (siehe auch Abschnitt 3.3 auf Seite 13, insbesondere die Warnung bezüglich `displaymath` im Zusammenhang mit *amsmath.sty*). Ferner funktioniert die Klassenoption `fleqn` nicht mehr.

Ersetze: `$$ . . . $$` durch `\[ . . . \]`  
 oder  
`\begin{displaymath}`  
 `. . .`  
`\end{displaymath}`

---

<sup>4</sup>syntaktisch analysieren, aufspalten

### 1.7 `\def` vs. `\newcommand`

Makros sollte man *immer* mittels `\newcommand{\<name>}{...}` definieren und *nicht* mit `\def\<name>{...}`. Das Hauptproblem von `\def` ist, dass keine Überprüfung auf die Existenz eines Makros durchgeführt wird. Es wird deshalb ggf. ohne Fehlermeldung/Warnung überschrieben.

Bereits existierende Makros können mit `\renewcommand{\<name>}{...}` undefiniert werden.

Wer genau weiß, *warum* er `\def` benötigt, weiß auch um dessen Nach- bzw. Vorteile und kann diesen Unterpunkt getrost ignorieren.

### 1.8 Verwendung von `\sloppy`

Der Schalter `\sloppy` sollte nicht verwendet werden. Schon gar nicht global in der Präambel. Wenn man in einzelnen Absätzen Probleme mit dem Umbruch hat, gilt:

1. Überprüfen, ob die entsprechenden Trennmuster (z. B. *(n)german.sty*) und T1-Schriften geladen sind (siehe auch FAQ [4, Punkt 5.3 ff.]).
2. Umformulieren. Man muss nicht unbedingt den Satz in dem das Umbruchproblem auftritt umformulieren, oft reicht es schon einen der vorhergehenden oder nachfolgenden Sätze umzuformulieren/umzustellen.

Erst wenn diese Punkte nicht geholfen haben, kann man versuchen mit `\sloppy` den nachfolgenden Absatz „lockerer“ setzen.

## 2 Obsoletes

Markus Kohm hat ein Perl-Script geschrieben, mit dem man online auf <http://kohm.de.tf/markus/texidate.html> seine Dateien auf die häufigsten Fehler überprüfen kann.

Allerdings ist zu beachten, dass es keinen vollständigen  $\TeX$ -Parser enthält und deshalb nur auf die „offensichtlichen“ Fehler überprüfen kann. Erst testen, dann posten.

### 2.1 Befehle

#### 2.1.1 Änderung des Schriftstils

In Tabelle 1 sind die alten und aktuellen Befehle zur Änderung des Schriftstils gegenüber gestellt. Die als „lokal“ bezeichneten Makros wirken nur auf ihr Argument, wohingegen die als „global/Schalter“ bezeichneten, sich auf den gesamten folgenden Text bis zum Ende der aktuellen Gruppe auswirken.

**Tabelle 1:** Befehle zur Änderung des Schriftstils

veraltet	Ersatz in $\LaTeX 2_{\epsilon}$	
	lokal	global/Schalter
<code>{\bf ...}</code>	<code>\textbf{...}</code>	<code>\bfseries</code>
<code>{\em ...}</code>	<code>\emph{...}</code>	<code>\em<sup>a</sup></code>
<code>{\it ...}</code>	<code>\textit{...}</code>	<code>\itshape</code>
—	<code>\textmd{...}</code>	<code>\mdseries</code>
<code>{\rm ...}</code>	<code>\textrm{...}</code>	<code>\rmfamily</code>
<code>{\sc ...}</code>	<code>\textsc{...}</code>	<code>\scshape</code>
<code>{\sf ...}</code>	<code>\textsf{...}</code>	<code>\sffamily</code>
<code>{\sl ...}</code>	<code>\textsl{...}</code>	<code>\slshape</code>
<code>{\tt ...}</code>	<code>\texttt{...}</code>	<code>\ttfamily</code>
—	<code>\textup{...}</code>	<code>\upshape</code>

<sup>a</sup>Kann in Makrodefinitionen nützlich sein

**Warum sollte man die alten Befehle nicht verwenden?** Die alten Befehle unterstützen nicht die Schriftverwaltung<sup>5</sup> von  $\LaTeX 2_{\epsilon}$ . `{\bf foo}` zum Beispiel setzt alle schon vorhandenen Schriftattribute zurück, bevor es „foo“ fett druckt. Das führt dazu, dass man nicht einfach einen fett-kursiven Stil durch `{\it \bf Test}` definieren kann. (Die angegebene Definition erzeugt: **Test**). Die aktuellen Befehle `\textbf{\textit{Test}}` hingegen verhalten sich wie erwartet und erzeugen: **Test**. Ferner gibt es bei den alten Befehlen keine „Italic-Korrektur“, z. B. *fünfhundert* (`{\it fünf}hundert`) und *fünfhundert* (`\textit{fünf}hundert`).

<sup>5</sup>NFSS: New Font Selection Scheme

### 2.1.2 Mathematische Brüche (`\over` vs. `\frac`)

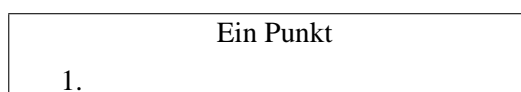
Der Befehl `\over` sollte vermieden werden. `\over` ist ein  $\TeX$ -Befehl, der durch die von  $\LaTeX$  abweichende Syntax schlechter bzw. nicht parsbar ist. Insbesondere das Paket *amsmath.sty* definiert `\frac{}{}` um und man erhält Fehlermeldungen bei der Verwendung von `\over`. Ein weiteres Argument für die Verwendung von `\frac{}{}` ist die für den Benutzer – vor allem bei komplexeren Brüchen – einfachere Zuordnung von Zähler und Nenner.

Ersetze: `$a \over b$` durch  `$\frac{a}{b}$`

### 2.1.3 Zentrierung mit `\centerline`

Der Befehl `\centerline` ist ebenfalls ein  $\TeX$ -Befehl und sollte in  $\LaTeX$  vermieden werden. `\centerline` ist einerseits inkompatibel zu einigen  $\LaTeX$ -Paketen (z. B. *color.sty*) und andererseits kann die Verwendung zu unerwünschten bzw. unerwarteten Effekten führen:

```
\begin{enumerate}
\item \centerline{Ein Punkt}
\end{enumerate}
```



Ersetze: `\centerline{...}` durch `{\centering ...}`  
 oder  
`\begin{center}`  
`...`  
`\end{center}`

**Anmerkung** Hinweise zur Zentrierung von Grafiken und Tabellen sind in Abschnitt 3.1 auf Seite 13 zu finden.

## 2.2 Klassen und Pakete

### 2.2.1 *scrlettr.cls* vs. *scrlettr2.cls*

Die Klasse *scrlettr.cls* aus dem KOMA-Script-Paket ist veraltet und wurde durch *scrlettr2.cls* ersetzt. Um ein *ähnliches* Layout wie die alte Klasse zu erreichen, kann man die Klassenoption `KOMAold` verwenden, die einen Kompatibilitätsmodus zur Verfügung stellt.

Ersetze: `\documentclass{scrlettr}` durch `\documentclass[KOMAold]{scrlettr2}`

**Anmerkung** Für neue Vorlagen und Briefe, sollte aber die neue Schnittstelle benutzt werden, da diese deutlich flexibler ist.

Eine Erklärung der Unterschiede des Benutzerinterfaces würde den Rahmen dieser Übersicht sprengen, deshalb muss ich hier auf den *scrguide* [3] verweisen.

### 2.2.2 *epsf.sty*, *psfig.sty*, *epsfig.sty* vs. *graphics.sty*, *graphicx.sty*

Die Pakete *epsf.sty* und *psfig.sty* sind durch *graphics.sty* oder *graphicx.sty* zu ersetzen. *epsfig.sty* ist nur ein Wrapper<sup>6</sup> um *graphicx.sty* für alte Dokumente, die mit *psfig.sty* erstellt wurden.

Da *epsfig.sty* intern *graphicx.sty* benutzt, kann man es noch verwenden, sollte aber für neu erstellte Dokumente auf das modernere *graphics.sty* oder *graphicx.sty* umsteigen. Die Syntax der beiden letztgenannten Pakete ist deutlich flexibler. Das Paket *epsfig.sty* wird hauptsächlich aus Kompatibilitätsgründen mitgeliefert.

Zu den Unterschieden zwischen den Paketen *graphics.sty* und *graphicx.sty* siehe grfguide [2]. Hinweise zur Zentrierung von Grafiken siehe Abschnitt 3.1 auf Seite 13.

Ersetze: `\usepackage{psfig}` durch `\usepackage{graphicx}`  
`\psfig{file=Bild,...}` `\includegraphics[...]{Bild}`

### 2.2.3 *doubleSPACE.sty* vs. *setspace.sty*

Um den Durchschuss zu ändern, sollte man das Paket *setspace.sty* verwenden. *doubleSPACE.sty* ist veraltet und wurde durch *setspace.sty* ersetzt. Siehe auch Abschnitt 1.4 auf Seite 3.

Ersetze: `\usepackage{doubleSPACE}` durch `\usepackage{setspace}`

### 2.2.4 *fancyheadings.sty*, *scrpage.sty* vs. *fancyhdr.sty*, *scrpage2.sty*

Das Paket *fancyheadings.sty* wurde durch *fancyhdr.sty* ersetzt. Eine weitere Alternative für angepasste Kopfzeilen bietet das Paket *scrpage2.sty* aus dem KOMA-Script-Bundle. Auch hier ist zu beachten, dass nicht *scrpage.sty* verwendet wird. Die Anleitung zu *scrpage2.sty* befindet sich im scrguide [3].

Ersetze: `\usepackage{fancyheadings}` durch `\usepackage{fancyhdr}`

Ersetze: `\usepackage{scrpage}` durch `\usepackage{scrpage2}`

### 2.2.5 *isolatin.sty*, *umlaut.sty* vs. *inputenc.sty*

**Generelles** Im Prinzip gibt es vier Möglichkeiten, Umlaute und andere nicht-ASCII-Zeichen einzugeben:

1.  $\text{H}\{\backslash"u\}ll\text{e}$  Der Vorteil dieser Art der Eingabe ist, dass sie immer und auf jedem System funktioniert.

Die Nachteile hingegen sind, dass das Kerning<sup>7</sup> zwischen den Buchstaben zerstört wird, es in einem deutschsprachigen Text äußerst umständlich ist und dass es äußerst schlecht lesbar ist.

Diese Variante sollte man – auf Grund des Kerningproblems – *immer* vermeiden.

2. Die Eingabe der Form  $\text{H}\backslash"ull\text{e}$  bzw.  $\text{H}\backslash\{u\}ll\text{e}$  hat die oben genannten Kerningprobleme nicht und ist ebenfalls auf jedem System nutzbar.

<sup>6</sup>Hier: Ein Stylefile, welches ein oder mehrere andere aufruft und damit Funktionen nachbildet.

<sup>7</sup>Einfügen positiver bzw. negativer Abstände zwischen Zeichen in Abhängigkeit der Zeichenkombination



## 2 Obsoletes

Die Nachteile bei dieser Art sind auch hier die aufwändige Eingabe und schlechtere Lesbarkeit.

Diese Variante ist die sinnvollste für Makrodefinitionen und Stylefiles, da sie encoding- und paketunabhängig ist.

3. Mit *(n)german.sty* bzw. der Option `(n)german` beim Paket *babel.sty* kann man die Umlaute etwas einfacher (`H"ulle`) eingeben. Der Vorteil ist auch hier wieder, dass es auf allen Systemen funktioniert. Da *babel.sty* bzw. *(n)german.sty* auf allen T<sub>E</sub>X-Installationen zu finden ist, sollte es auch keine Kompatibilitätsprobleme geben.

Die Nachteile sind auch hier die umständlichere Eingabe und schlechtere Lesbarkeit.

Diese Variante ist für Fließtext relativ gut verwendbar. Sollte aber in Makrodefinitionen und Präambeln vermieden werden.

4. Die direkte Eingabe (Hülle). Die Vorteile liegen auf der Hand. Der Text ist „normal“ schreib- und lesbar.

Der Nachteil ist, dass man L<sup>A</sup>T<sub>E</sub>X mit der verwendeten Eingabekodierung bekannt machen muss und dass es beim Austausch von Dateien zwischen verschiedenen Systemen evtl. zu Problemen kommen kann. Das ist *kein* Problem für T<sub>E</sub>X bzw. L<sup>A</sup>T<sub>E</sub>X selbst, aber es kann evtl. zu *Darstellungsproblemen* in den Editoren auf den verschiedenen Systemen führen. Zum Beispiel könnte ein in iso-8859-15 (latin9) kodierter € in einem Editor unter Windows (CP1252) als  $\text{€}$  dargestellt werden.

Diese Variante ist sehr gut für Fließtext verwendbar. Sollte aber in Makrodefinitionen und Präambeln vermieden werden.

Zusammenfassend kann also gesagt werden, dass man in Makros, Präambeln und Stylefiles die Form `H"ulle` oder `H\{u}lle` verwenden sollte und im übrigen Text entweder `H"ulle` oder Hülle.

**Eingabekodierung** Um die verwendete Kodierung L<sup>A</sup>T<sub>E</sub>X bekannt zu machen, sollte man *nicht* *isolatin1.sty* bzw. *isolatin.sty* oder *umlaut.sty* verwenden! Diese Pakete sind veraltet bzw. nicht auf allen Systemen vorhanden.

Korrekt ist das Paket *inputenc.sty* mit den Optionen

**latin1/latin9** für unixoide Systeme (latin1 ist auch unter Windows verwendbar)

**ansinew** für Windows

**applemac** Macs

**cp850** OS/2

zu benutzen.

Ersetze: `\usepackage{isolatin1}` durch `\usepackage[latin1]{inputenc}`

Ersetze: `\usepackage{umlaut}` durch `\usepackage[latin1]{inputenc}`

### 2.2.6 *t1enc.sty* vs. *fontenc.sty*

Dieses Thema ist in der FAQ [4, Punkt 10.1 ff.] eigentlich ausreichend erörtert. Hier nur kurz der Hinweis, dass das Paket *t1enc.sty* veraltet ist und deshalb durch *fontenc.sty* ersetzt werden sollte!

Ersetze: `\usepackage{t1enc}` durch `\usepackage[T1]{fontenc}`

### 2.2.7 *natdin.bst* vs. *dinat.bst*

Das Stylefile *natdin.bst* wurde durch *dinat.bst* ersetzt.

Ersetze: `\bibliographystyle{natdin}` durch `\bibliographystyle{dinat}`

## 2.3 Schriften

Das Thema „Schriften und L<sup>A</sup>T<sub>E</sub>X“ ist ein Quell ewiger „Freude“ in `de.comp.text.tex`, meistens ausgelöst durch die Frage, warum denn die Schrift im Acrobat<sup>®</sup> Reader so pixelig sei. Die häufigsten *falschen* Antworten auf diese Frage verweisen auf *times.sty* bzw. *pslatex.sty*. Durch die Nutzung dieser Pakete werden gänzlich andere Schriften eingestellt.

Um „schöne“ Schriften (Computer Modern) im AR zu erhalten, sei hiermit auf die FAQ [4, Punkt 10.1.7/10.1.8] verwiesen.

### 2.3.1 *times.sty*

Das Paket *times.sty* ist veraltet (siehe `psnfss2e` [6]). Es stellt `\rmdefault` auf die Schrift „Times“, `\sfdefault` auf „Helvetica“ und `\ttdefault` auf „Courier“ um, *ohne* jedoch die passenden Mathematikschriften einzubinden. Ferner wird die Helvetica nicht korrekt skaliert und wirkt zu groß. Wenn man die Kombination Times/Helvetica/Courier benutzen möchte, dann folgendermaßen:

Ersetze: `\usepackage{times}` durch `\usepackage{mathptmx}`  
`\usepackage[scaled=.90]{helvet}`  
`\usepackage{courier}`

**Anmerkung** Der Skalierungsfaktor für *helvet.sty* in Kombination mit der Times sollte zwischen 0.90 und 0.92 liegen.

### 2.3.2 *mathptm.sty*

Das Paket *mathptm.sty* ist der Vorgänger von *mathptmx.sty*.

Ersetze: `\usepackage{mathptm}` durch `\usepackage{mathptmx}`

### 2.3.3 *pslatex.sty*

Das Paket *pslatex.sty* arbeitet intern wie *mathptm.sty* + *helvet.sty* (skaliert), wobei allerdings eine zu eng laufende Courier gewählt wird. Der Hauptnachteil von *pslatex.sty* ist, dass es *nicht* mit T1- und TS1-Encoding funktioniert.

Ersetze: `\usepackage{pslatex}` durch `\usepackage{mathptmx}`  
`\usepackage[scaled=.90]{helvet}`  
`\usepackage{courier}`

**Anmerkung zu allen Times/Helvetica-Kombinationen** Man kann auch als Schreibmaschi-  
 nenschrift bei der `cmtt` bleiben, also auf das Laden von `courier.sty` verzichten.

### 2.3.4 *palatino.sty*

Das Paket *palatino.sty* verhält sich wie *times.sty* (außer das natürlich `\rmdefault` auf „Palati-  
 no“ gesetzt wird) und sollte deshalb nicht mehr benutzt werden.

Ersetze: `\usepackage{palatino}` durch `\usepackage{mathpazo}`  
`\usepackage[scaled=.95]{helvet}`  
`\usepackage{courier}`

**Anmerkung** Der Skalierungsfaktor für *helvet.sty* in Kombination mit der Schrift Palatino soll-  
 te 0.95 betragen.

Die „Helvetica“ ist *nicht* die optimale serifenlose Schrift in Kombination mit der „Palatino“,  
 aber die beste *freiverfügbare*. Wer eine (auch ältere) CorelDraw<sup>®</sup>-CD besitzt, kann die „Palatino“  
 auch sehr gut mit den Schriften „Frutiger“<sup>8</sup> oder „Optima“<sup>9</sup> kombinieren. Walter Schmidt hat  
 auf seiner Homepage<sup>10</sup> die entsprechenden T<sub>E</sub>X-Anpassungen veröffentlicht.

### 2.3.5 *mathpple.sty*

Dieses Paket ist der Vorläufer von *mathpazo.sty*. Ihm fehlen einzelne Zeichen, die Schriften  
 werden aus den Euler-Fonts genommen, andere Zeichen passen nicht gut zu Palatino und die  
 Zeichenabstände sind zum Teil falsch. Genaueres siehe `psnfss2e` [6].

### 2.3.6 Aufrechte griechische Buchstaben

Die im folgenden rot markierten Passagen sind nicht veraltet im Sinne von „man soll sie nicht  
 mehr benutzen“, aber es gibt nun mit dem Paket *upgreek.sty* eine Vereinfachung der Eingabe.  
 Hinweise zur Benutzung bitte wie immer der Dokumentation `upgreek` [5] entnehmen.

#### Die *pifont.sty*-Tricks

Ersetze: `\usepackage{pifont}` durch `\usepackage{upgreek}`  
`\newcommand{\uppi}{\Pisymbol{psy}{112}}`  `$\uppi$`  
`\uppi`  
 oder  
`\newcommand[1]{\upgreek}{%`  
 `\usefont{U}{psy}{m}{n}#1}`  
`\upgreek{p}`

<sup>8</sup>Bitstream „Humanist 777“, bfr

<sup>9</sup>Bitstream „Zapf Humanist“, bop

<sup>10</sup>Schriften für T<sub>E</sub>X: <http://home.vr-web.de/was/fonts>

## 2 Obsoletes

### Der *babel.sty*-Trick

Ersetze:

```
\usepackage[greek,...]{babel}  
\newcommand[1]{\upgreek}{%  
  \foreignlanguage{greek}{#1}}  
\upgreek{p}
```

durch

```
\usepackage{upgreek}  
$\uppi$
```

### 3 Verschiedenes

Dieser Abschnitt enthält – mit Ausnahme von 3.2 – eher allgemeine Tipps und Hinweise als „Sünden“.

#### 3.1 Gleitumgebungen – „figure“, „table“

Um den Inhalt einer Gleitumgebung zu zentrieren, sollte man `\centering` an Stelle der `\begin{center}`-`\end{center}`-Umgebung verwenden, da diese zusätzlichen vertikalen Abstand einfügt, der meistens nicht erwünscht ist.

Ersetze:

<code>\begin{figure}</code>	durch	<code>\begin{figure}</code>
<code>\begin{center}</code>		<code>\centering</code>
<code>\includegraphics{bild}</code>		<code>\includegraphics{bild}</code>
<code>\end{center}</code>		<code>\end{figure}</code>
<code>\end{figure}</code>		

**Anmerkung** Wenn man innerhalb des Fließtextes oder der `titlepage`-Umgebung einen Bereich zentrieren möchte, kann dieser zusätzliche Abstand natürlich durchaus erwünscht sein.

#### 3.2 Der Anhang

Der Anhang wird mit dem *Schalter* `\appendix` eingeleitet. Er ist *keine* Umgebung.

Ersetze:

<code>\begin{appendix}</code>	durch	<code>\appendix</code>
<code>\section{Blub}</code>		<code>\section{Blub}</code>
<code>\end{appendix}</code>		

#### 3.3 Mathematiksatz

Generell sollte man für komplizierteren Mathematiksatz `amsmath.sty` benutzen. Es bietet neue Umgebungen, die vor allem `eqnarray` ersetzen sollen. Die Vorteile des Paketes:

- Abstände innerhalb und außerhalb von Umgebungen sind konsistenter.
- Gleichungsnummern werden so positioniert, dass sie nicht mehr überdrückt werden.
- Die neuen Umgebungen (z. B. `split`) ermöglichen es, lange Gleichungen einfacher zu umbrechen.
- Einfache Möglichkeit, neue Operatoren (ähnlich wie `\sin` usw.) mit sauberen Zeichenabständen zu definieren.

**Warnung** Bei der Verwendung von `amsmath.sty` sollte man die Umgebungen `displaymath`, `eqnarray` und `eqnarray*` *keinesfalls* weiterverwenden, da diese von `amsmath.sty` nicht unterstützt werden. Die Folge wäre wieder inkonsistente Abstände.

`\[...]` wird von `amsmath.sty` korrekt angepasst und kann an Stelle von `displaymath` genutzt werden. `eqnarray` und `eqnarray*` kann in erster Näherung durch `align` bzw.

### 3 Verschiedenes

`align*` ersetzt werden. Für eine vollständige Übersicht der Möglichkeiten von *amsmath.sty* verweise ich auf die Dokumentation `amslatex` [1].

Ersetze:	<code>\begin{eqnarray}</code>	durch	<code>\begin{align}</code>
	<code>a &amp;= &amp; b \\\</code>		<code>a &amp;= b \\\</code>
	<code>b &amp;= &amp; c \\\</code>		<code>b &amp;= c \\\</code>
	<code>a &amp;= &amp; c</code>		<code>a &amp;= c</code>
	<code>\end{eqnarray}</code>		<code>\end{align}</code>

#### 3.4 Die `\*name`-Makros

Da in `de.comp.text.tex` von Zeit zu Zeit danach gefragt wird, wie man zum Beispiel „Literatur“ in „Quellenverzeichnis“ ändern kann, habe ich in Tabelle 2 auf der nächsten Seite die entsprechenden Makros zusammengestellt. Sie sind aus *german.sty* entnommen.

Um beispielsweise das „Abbildungsverzeichnis“ in „Abbildungen“ umzubennen, benutzt man folgenden Befehl:

```
\renewcommand*\listfigurename{Abbildungen}
```

Die anderen Makros lassen sich analog umbenennen. Bei der Verwendung von *babel.sty* muss man mit `\addto` arbeiten. Siehe auch die De-TeX-FAQ [4, Punkt 8.5.9].

```
\addto{\captionsngerman}{%  
  \renewcommand*\listfigurename{Abbildungen}}
```

### 3 Verschiedenes

**Tabelle 2:** Von *(n)german.sty* bzw. *babel.sty* mit der Option *(n)german* definierte Makros

Makroname	Original Definition	deutsche Ausgabe
<code>\prefacename</code>	Preface	Vorwort
<code>\refname<sup>a</sup></code>	References	Literatur
<code>\abstractname</code>	Abstract	Zusammenfassung
<code>\bibname<sup>b</sup></code>	Bibliography	Literaturverzeichnis
<code>\chaptername</code>	Chapter	Kapitel
<code>\appendixname</code>	Appendix	Anhang
<code>\contentsname</code>	Contents	Inhaltsverzeichnis
<code>\listfigurename</code>	List of Figures	Abbildungsverzeichnis
<code>\listtablename</code>	List of Tables	Tabellenverzeichnis
<code>\indexname</code>	Index	Index
<code>\figurename</code>	Figure	Abbildung
<code>\tablename</code>	Table	Tabelle
<code>\partname</code>	Part	Teil
<code>\enclname</code>	encl	Anlage(n)
<code>\ccname</code>	cc	Verteiler
<code>\headtoname</code>	To	An
<code>\pagename</code>	Page	Seite
<code>\seename</code>	see	siehe
<code>\alsoname</code>	see also	siehe auch

<sup>a</sup>Nur in den `article`-Klassen

<sup>b</sup>Nur in den `report`- und `book`-Klassen

## Literatur

- [1] AMERICAN MATHEMATICAL SOCIETY: *User's Guide for the amsmath Package*. Dezember 1999, Version 2.0.  
URL: CTAN:macros/latex/required/amslatex/.
- [2] DAVID P. CARLISLE: *Packages in the 'graphics' bundle*. Januar 1999.  
URL: CTAN:macros/latex/required/graphics/.
- [3] MARKUS KOHM, FRANK NEUKAM und AXEL KIELHORN: *Das KOMA-Script Paket*. April 2003, Version 2.9o.  
URL: CTAN:macros/latex/supported/koma-script/.
- [4] BERND RAICHLE, ROLF NIEPRASCHK und THOMAS HAFNER: *Fragen und Antworten (FAQ) über das Textsatzsystem T<sub>E</sub>X und DANTE, Deutschsprachige Anwendervereinigung T<sub>E</sub>X e.V. WWW*, September 2003, Version 72.  
URL: <http://www.dante.de/faq/de-tex-faq/>.
- [5] WALTER SCHMIDT: *The upgreek package for L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>*. Mai 2001, Version 1.0.  
URL: CTAN:macros/latex/contrib/supported/was/.
- [6] WALTER SCHMIDT: *Using common PostScript fonts with L<sup>A</sup>T<sub>E</sub>X*. April 2002, PSNFSS version 9.0.  
URL: CTAN:macros/latex/required/psnfss/psnfss2e.pdf
- [7] WALTER SCHMIDT, JÖRG KNAPPEN, HUBERT PARTL und IRENE HYNÄ: *L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>-Kurzbeschreibung*. April 1999, Version 2.1.  
URL: CTAN:info/lshort/german/.