
Das L^AT_EX 2_ε-Sündenregister

oder Veraltete Befehle, Pakete und andere Fehler

Marc Ensenbach, Mark Trettin

E-Mail-Kontakt: Marc.Ensenbach@post.rwth-aachen.de

Version 2.3 vom 20. September 2011

Zusammenfassung

Die Erstellung dieser kleinen Übersicht wurde angeregt durch eine Diskussion in der deutschsprachigen T_EX-Newsgroup^a über das wiederholte Auftauchen von veralteten und »schlechten« Paketen und Befehlen.

Wir versuchen in diesem Artikel die gängigsten Fehler zu zeigen und Alternativen anzubieten. Diese Übersicht soll weder Einführungen wie l2kurz [10] noch die De-TeX-FAQ^b ersetzen, sondern lediglich einen kleinen Überblick bieten.

Für Vorschläge, Verbesserungen und Kommentare sind wir dankbar.

^a de.comp.text.tex

^b <http://projekte.dante.de/DanteFAQ/WebHome>

Copyright © 2003–2007 by Mark Trettin, 2009–2011 by Marc Ensenbach.

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/>).

Die in dieser Publikation erwähnten Software- und Hardwarebezeichnungen sind in den meisten Fällen auch eingetragene Warenzeichen und unterliegen als solche den gesetzlichen Bestimmungen.

Wir bedanken uns bei Ralf Angeli, Christoph Bier, Christian Faulhammer, Jürgen Fenn¹, Ulrike Fischer, Ralf Heckmann, Yvon Henel², Yvonne Hoffmüller, David Kastrup, Markus Kohm, Thomas Lotze, Gonzalo Medina Arellano³, Frank Mittelbach, Heiko Oberdiek, Walter Schmidt, Uwe Siart, Stefan Stoll, Knut Wenzig, Emanuele Zannarini⁴ und Reinhard Zierke für Tipps, Anmerkungen und Korrekturen. Falls wir jemanden vergessen haben sollten, bitten wir um eine Mail.

¹ Englische Übersetzung: <CTAN:info/l2tabu/english/l2tabuen.pdf>

² Französische Übersetzung: <CTAN:info/l2tabu/french/l2tabufr.pdf>

³ Spanische Übersetzung: <CTAN:info/l2tabu/spanish/l2tabues.pdf>

⁴ Italienische Übersetzung: <CTAN:info/l2tabu/italian/l2tabuit.pdf>

Inhaltsverzeichnis

1	»Todsünden«	5
1.1	a4.sty, a4wide.sty	5
1.2	Layoutänderungen	5
1.3	Änderungen von Paketen und Klassen	5
1.4	Änderung des Zeilenabstandes mittels \baselinestretch	6
1.5	Absatz einzug und -abstand (\parindent, \parskip)	6
1.6	Abgesetzte Formeln mit \$\$...\$\$	7
1.7	\def vs. \newcommand	8
1.8	Verwendung von \sloppy	8
2	Veraltete Befehle, Klassen und Pakete	10
2.1	Befehle	10
2.1.1	Änderung des Schriftstils	10
2.1.2	Mathematische Brüche (\over vs. \frac)	11
2.1.3	Zentrierung mit \centerline	11
2.2	Klassen und Pakete	11
2.2.1	scrlettr.cls vs. scrlettr2.cls	11
2.2.2	epsf.sty, psfig.sty, epsfig.sty vs. graphics.sty, graphicx.sty	12
2.2.3	doubleSPACE.sty vs. setSPACE.sty	12
2.2.4	fancyheadings.sty, scrpage.sty vs. fancyhdr.sty, scrpage2.sty	12
2.2.5	Die caption.sty-Familie	12
2.2.6	isolatin.sty, umlaut.sty vs. inputenc.sty	13
2.2.7	t1enc.sty vs. fontenc.sty	15
2.2.8	natdin.bst vs. dinat.bst	15
2.2.9	glossary.sty vs. glossaries.sty	15
2.2.10	SIstyle.sty und SIunits.sty vs. siunitx.sty	15
2.2.11	subfigure.sty und subfig.sty vs. subcaption.sty	15
2.3	Schriften	16
2.3.1	times.sty	16
2.3.2	mathptm.sty	16
2.3.3	pslatex.sty	16
2.3.4	palatino.sty	17
2.3.5	mathpple.sty	17
2.3.6	utopia.sty	17
2.3.7	Aufrechte griechische Buchstaben	18
2.3.8	euler.sty vs. eulervm.sty	18
2.3.9	ae.sty, aecompl.sty und zefonts.sty	18
3	Verschiedenes	20
3.1	Gleitumgebungen – »figure«, »table«	20
3.2	Der Anhang	20

Inhaltsverzeichnis

3.3	Mathematiksatz	20
3.4	Die Verwendung von \graphicspath	21
3.5	Die \dotsname-Makros	22
A	Beispiel zu \sloppy	24

1 »Todsünden«

In diesem Abschnitt haben wir 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^AT_EX-Quelltext gestrichen und durch die Klassenoption a4paper beziehungsweise paper=a4 für KOMA-Script ab Version 3 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 zum Beispiel typearea.sty erzeugten benötigt, dann sollte man bitte das Paket geometry.sty verwenden und nicht versuchen, »zu Fuß« \oddsidemargin & Co. zu ändern. Das Paket anysize.sty ist veraltet, und das hier vormals empfohlene Paket vmargin.sty sollte ebenfalls vermieden werden, da es unerwünschte Nebeneffekte erzeugen kann.⁵

Unter gar keinen Umständen sollte man an \hoffset beziehungsweise \voffset herumfummeln, außer man kennt sich sehr gut mit den T_EX-Internia aus.

1.3 Änderungen von Paketen und Klassen

Niemals Dokumentklassen (article.cls, scrbook.cls usw.) oder Styledateien (varioref.sty, color.sty) direkt ändern! Entweder man bastelt sich Containerklassen beziehungsweise -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 zur Anpassung von Dokumentenklassen⁶.

⁵ Vergleiche auch [Message-ID: <uirrzdr16.fsf@uwe-siart.de>](mailto:uirrzdr16.fsf@uwe-siart.de)

⁶ <http://projekte.dante.de/DanteFAQ/Dokumentenklasse>

Hinweis Generell sollte man solche zusätzlich installierten Klassen und Pakete entweder in den lokalen oder den $\$HOME$ -T_EX-Baum speichern, damit bei einem Upgrade der T_EX-Distribution diese Änderungen nicht überschrieben werden. Braucht man diese Änderungen nur in dem speziellen Projekt und möchte es weitergeben, dann kann man die angepasste Klasse auch im aktuellen Projektverzeichnis speichern.

1.4 Änderung des Zeilenabstandes mittels `\baselinestretch`

Anpassungen von Parametern sollten generell auf der obersten dafür vorgesehenen Ebene der Benutzungsschnittstelle erfolgen. Eine Änderung des Durchschusses kann auf drei Ebenen durchgeführt werden:

1. Verwendung des Paketes `setspace.sty`
2. Verwendung der L^AT_EX-Anweisung `\linespread{<Faktor>}`
3. Umdefinition von `\baselinestretch`

Eine Umdefinition von Parametern wie `\baselinestretch` stellt die unterste Ebene dar und sollte deshalb Paketen vorbehalten bleiben. Eine bessere, da extra dafür vorgesehene Methode, ist die Verwendung von `\linespread{<Faktor>}`. Die beste Vorgehensweise stellt aber die Verwendung des Paketes `setspace.sty` dar, das sich bei der Änderung des Durchschusses um die Beibehaltung der Abstände in Fußnoten oder Listenumgebungen kümmert, deren Änderung meist nicht erwünscht ist.

Wenn man also einen größeren Durchschuss benötigt (zum Beispiel eineinhalbzeilig oder zweizeilig), bietet sich folglich zuerst das Paket `setspace.sty` an. Möchte man hingegen nur kleine Anpassungen für andere Schriften als Computer Modern (zum Beispiel Palatino) machen, dann kann beziehungsweise sollte man `\linespread{<Faktor>}` verwenden.

Bei der Schrift Palatino bietet sich beispielsweise `\linespread{1.05}` an.

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. Die »Schriftabhängigkeit« bedeutet hierbei nicht nur, dass sich der Einzug mit der Schriftgröße ändert, sondern beinhaltet darüber hinaus auch eine Abhängigkeit von der Schriftart, da die Geviertgröße 1em vom jeweiligen Entwerfer individuell für jede Schriftart angepasst wird.

- man L^AT_EX-Syntax verwendet, da diese die wenigsten Probleme birgt. Zum Beispiel ist sie für externe Programme/Skripte leichter zu parsen⁷, 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.

Das Paket `parskip.sty` und die KOMA-Script-Klassen treiben einigen Aufwand, um diese Begleiterscheinungen zu vermeiden. Zur Verwendung der verschiedenen Optionen (`parskip=full`, `parskip=half` 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 plainT_EX-Befehl und sollte in L^AT_EX vermieden werden, da dadurch die vertikalen Abstände bei abgesetzten Formeln inkonsistent werden (siehe auch Abschnitt 3.3 auf Seite 20, insbesondere die Warnung bezüglich `displaymath` im Zusammenhang mit `amsmath.sty`). Ferner funktioniert die Klassenoption `fleqn` nicht mehr.

▷ **Ersetze:** `$$...$$` **Durch:** `\[...\]`

oder (nur bei Verwendung von `amsmath.sty`)

```
\begin{equation*}
...
\end{equation*}
```

oder (nur ohne `amsmath.sty`)

```
\begin{displaymath}
...
\end{displaymath}
```

⁷ syntaktisch analysieren, aufspalten

1.7 \def vs. \newcommand

Makros sollte man *immer* mittels `\newcommand{<name>}{...}` definieren und *nicht* mit `\def{<name>}{...}`.⁸

Das Hauptproblem von `\def` ist, dass keine Überprüfung auf die Existenz eines Makros durchgeführt wird. Es wird deshalb gegebenenfalls ohne Fehlermeldung/Warnung überschrieben.

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

Wer genau weiß, *warum* er `\def` benötigt, weiß auch um dessen Nach- beziehungsweise 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. mittels `babel.sty`) und T1-Schriften geladen sind (siehe auch die FAQ zur Silbentrennung⁹).
2. Paket `microtype.sty` verwenden. Dadurch wird die Laufweite der Buchstaben leicht angepasst, um einen besseren Umbruch zu erreichen. Diese automatische Anpassung ist nur mit pdfTeX ab Version 1.20 möglich und funktioniert nur bei der Erzeugung einer PDF-Datei, nicht jedoch bei der Verwendung von pdfTeX im DVI-Modus.
3. 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.
4. Moderate Anpassung der Parameter, die TeX bei der Berechnung der Zeilenumbrüche verwendet. Axel Reichert hat in `de.comp.text.tex` einmal seine persönliche Anpassung¹⁰ gepostet¹¹, mit der sich die meisten Umbruchprobleme bei weiterhin gutem Layout vermeiden lassen. (Man sollte hierbei allerdings beachten, dass nun auftretende Warnungen *wirklich* zu beheerzigen und durch Umformulieren zu beseitigen sind.):

⁸ Siehe auch [4, Abschnitt 2.7.2] und [6, Abschnitt 3.4]

⁹ <http://projekte.dante.de/DanteFAQ/Silbentrennung>

¹⁰ Man kann diese Werte natürlich nach dem persönlichen Geschmack ändern, aber man sollte vor allem bei `\emergencystretch` aufpassen. Sonst erhält man einen löchrigen Blocksatz, wie bei einem sehr bekannten Textverarbeitungsprogramm.

¹¹ Zu finden unter der Message-ID: `<a84us0$plqcm$7@ID-30533.news.dfncis.de>`


```

\tolerance 1414
\hbadness 1414
\emergencystretch 1.5em
\hfuzz 0.3pt
\widowpenalty=10000
\vfuzz \hfuzz
\raggedbottom

```

Erst wenn diese Punkte nicht geholfen haben, kann man versuchen, mit der `sloppypar`-Umgebung den nachfolgenden Absatz »lockerer« zu setzen.

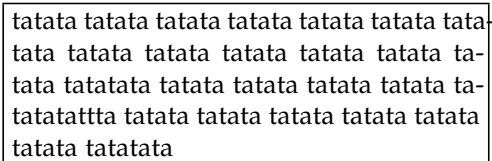


Abbildung 1: Beispiel mit \LaTeX s Standardwerten

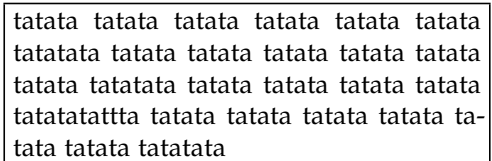


Abbildung 2: Beispiel mit `\sloppy`

In den Abbildungen 1 und 2 wird versucht, den Effekt von `\sloppy` darzustellen. Markus Kohm hat in `comp.text.tex` ein Beispiel veröffentlicht, welches den Effekt sehr deutlich zeigt. Es ist mit seiner Erlaubnis angehängt (siehe Anhang A auf Seite 24).

2 Veraltete Befehle, Klassen und Pakete

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 T_EX-Parser enthält und deshalb nur 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 L ^A T _E X 2 _ε	
	lokal	global/Schalter
<code>{\bf ...}</code>	<code>\textbf{...}</code>	<code>\bfseries</code>
—	<code>\emph{...}</code>	<code>\em^a</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>

^a Nützlich in Makrodefinitionen. Innerhalb des Texts sollte nicht `{\em ...}`, sondern `\emph{...}` verwendet werden.

Warum sollte man die alten Befehle nicht verwenden? Die alten Befehle unterstützen nicht die Schriftverwaltung¹² von L^AT_EX 2_ε. `{\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}}`

¹² NFSS: New Font Selection Scheme. Einen Überblick über das NFSS gibt der fntguide [5].

hingegen verhalten sich wie erwartet und erzeugen: *Test*. Ferner gibt es bei den alten Befehlen keine »Italic-Korrektur«, zum Beispiel *fünfhundert* (`\it fünf`hundert) und *fünfhundert* (`\textit{fünf}`hundert).

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 beziehungsweise 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 – vorallem 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. Einerseits ist `\centerline` inkompatibel zu einigen \LaTeX -Paketen (z. B. `color.sty`), und andererseits kann die Verwendung zu unerwünschten beziehungsweise unerwarteten Effekten führen:

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

Ein Punkt
1.

▷ **Ersetze:** `\centerline{...}`

Durch: `{\centering ...}` oder `\begin{center}`
`...`
`\end{center}`

Anmerkung Einige Hinweise zur Zentrierung von Grafiken und Tabellen sind in Abschnitt 3.1 auf Seite 20 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 die neue `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 wird hier auf den scrguide [3] verwiesen.

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 Wrapper¹³ um graphicx.sty für alte Dokumente, die mit psfig.sty erstellt wurden, verwenden zu können.

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 20.

▷ **Ersetze:** `\usepackage{psfig}`
`\psfig{file=Bild,...}`

Durch: `\usepackage{graphicx}`
`\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. Bitte auch Abschnitt 1.4 auf Seite 6 beachten.

▷ **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 Die caption.sty-Familie

Das Pakete caption2.sty sollte nicht mehr verwendet werden, da es eine neue Version (v3.x) von caption.sty gibt. Man sollte allerdings darauf achten, dass eine nicht zu

¹³ Hier: Ein Stylefile, welches ein oder mehrere andere aufruft und damit Funktionen nachbildet.

alte Version des Paketes benutzt wird. Um dies sicherzustellen, sollte man das Paket folgendermaßen laden:

- ▷ **Ersetze:** `\usepackage{caption}` **Durch:** `\usepackage{caption}[2008/08/24]`

Hatte man vorher `caption2.sty` benutzt, dann sollte man unbedingt die Dokumentation anleitung [11, Abschnitt C.2] beachten.

2.2.6 `isolatin.sty`, `umlaut.sty` vs. `inputenc.sty`

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

1. `H{"u}lle`: Der Vorteil dieser Art der Eingabe ist, dass sie immer und auf jedem System funktioniert.

Die Nachteile hingegen sind, dass das Kerning¹⁴ 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 `H\"u lle` beziehungsweise `H\"{u}lle` hat die oben genannten Kerningprobleme nicht und ist ebenfalls auf jedem System nutzbar.

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` beziehungsweise der Option `(n)german` beim Paket `babel.sty` kann man die Umlaute etwas einfacher (`H"u lle`) eingeben. Der Vorteil ist auch hier wieder, dass es auf allen Systemen funktioniert. Da `babel.sty` beziehungsweise `(n)german.sty` auf allen \TeX -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 \LaTeX mit der verwendeten Eingabekodierung bekannt machen muss und dass es beim Austausch von Dateien zwischen verschiedenen

¹⁴ Einfügen positiver beziehungsweise negativer Abstände zwischen Zeichen in Abhängigkeit der Zeichenkombination

Systemen eventuell zu Problemen kommen kann. Das ist *kein* Problem für \TeX beziehungsweise \LaTeX 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 $\text{\textcircled{C}}$ 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 $\text{H}\text{"ulle}$ oder $\text{H}\text{"{u}lle}$ verwenden sollte und im übrigen Text entweder $\text{H}\text{"ulle}$ oder Hülle .

Eingabekodierung Um die verwendete Kodierung \LaTeX bekannt zu machen, sollte man *nicht* `isolatin1.sty` beziehungsweise `isolatin.sty` oder `umlaut.sty` verwenden! Diese Pakete sind veraltet beziehungsweise nicht auf allen Systemen vorhanden.

Korrekt ist das Paket `inputenc.sty` mit folgenden Optionen zu benutzen:

latin1/latin9 für unixoide Systeme (latin1 ist auch unter Microsoft Windows und Mac OS X verwendbar)

ansinew für Microsoft Windows

applemac für Macs¹⁵

cp850 für OS/2

utf8 für »moderne« unixoide Systeme beziehungsweise für Systeme (und Editoren), die Unicode (UTF-8) unterstützen

- | | |
|---|--|
| ▷ Ersetze: <code>\usepackage{isolatin1}</code> | Durch: <code>\usepackage[latin1]{inputenc}</code> |
| ▷ Ersetze: <code>\usepackage{umlaut}</code> | Durch: <code>\usepackage[latin1]{inputenc}</code> |

Anmerkung Die oben angegebene Methode zur Deklaration der Eingabekodierung findet bei »klassischen« \LaTeX -Übersetzern wie `pdf \LaTeX` Anwendung. Die Neuentwicklungen `X \LaTeX` und `Lua \LaTeX` verarbeiten Unicode-Dateien auch ohne Kodierungsangabe. Beide können bereits produktiv eingesetzt werden, verlangen aber in der Regel ein Umschreiben »alter« \LaTeX -Dateien in Bezug auf die Sprach- und Schriftartenwahl und besitzen auch noch nicht die mikrotypografischen Fähigkeiten in dem Umfang, wie `pdf \LaTeX` sie bietet.

¹⁵ Bei der Verwendung von OS X sollte ebenfalls `latin1/latin9` verwendet werden, da diese Kodierung besser für den Dateiaustausch mit Benutzern anderer Betriebssysteme geeignet ist als `applemac`. Allerdings sollte unbedingt die Eingabekodierung des verwendeten Editors beachtet werden.

2.2.7 `tlenc.sty` vs. `fontenc.sty`

Das Paket `tlenc.sty` ist veraltet und sollte deshalb durch `fontenc.sty` ersetzt werden.

- ▷ **Ersetze:** `\usepackage{tlenc}` **Durch:** `\usepackage[T1]{fontenc}`

2.2.8 `natdin.bst` vs. `dinat.bst`

Der Autor von `dinat.bst` empfiehlt, `dinat.bst` durch `natdin.bst` zu ersetzen, da die Weiterentwicklung von `dinat.bst` eingestellt und `natdin.bst` stark verbessert wurde.

- ▷ **Ersetze:** `dinat.bst` **Durch:** `natdin.bst`

Anmerkung In früheren Versionen dieses Dokuments wurde genau die gegenteilige Empfehlung gegeben, die auf dem damaligen Entwicklungsstand beruhte. Die Verbesserungen an `natdin.bst`, die seither durchgeführt wurden, rechtfertigen nun die Bevorzugung von `natdin.bst`.

2.2.9 `glossary.sty` vs. `glossaries.sty`

Das Paket `glossary.sty` ist veraltet; als Nachfolger wurde von der Autorin `glossaries.sty` bereitgestellt. Details zum Umstieg finden sich in [12].

2.2.10 `SIstyle.sty` und `SIunits.sty` vs. `siunitx.sty`

Die Pakete `SIstyle.sty` und `SIunits.sty` zum Satz von physikalischen Einheiten werden nicht mehr weiterentwickelt. Die Funktionalität beider Pakete wurde im Nachfolger `siunitx.sty` vereint. Aufgrund der großen Flexibilität kann `siunitx.sty` auch Pakete wie `unitsdef.sty`, `units.sty`, `fancyunits.sty` und `fancynum.sty` ersetzen.

- ▷ **Ersetze:** `\usepackage{SIunits}`
`\unit{1}{\metre}` **Durch:** `\usepackage{siunitx}`
`\SI{1}{\metre}`

2.2.11 `subfigure.sty` und `subfig.sty` vs. `subcaption.sty`

Die Pakete `subfigure.sty` und `subfig.sty` werden nicht mehr weiterentwickelt und sind inkompatibel zu einigen aktuelleren Paketen wie `hyperref.sty` und `caption.sty`. Als Ersatz empfiehlt sich `subcaption.sty`.

- ▷ **Ersetze:** `\usepackage{subfig}`
`\begin{figure}`
`\subfloat[][Text]{Bild}`
`\end{figure}` **Durch:** `\usepackage{caption}`
`\usepackage{subcaption}`
`\begin{figure}`
`\begin{subfigure}{Breite}`
`Bild\caption{Text}`
`\end{subfigure}`
`\end{figure}`

2.3 Schriften

Das Thema »Schriften und \LaTeX « ist ein Quell ewiger »Freude« in [de.comp.text.tex](#), meistens ausgelöst durch die Frage, warum denn die Schrift im Acrobat Reader so pixelig sei. Die häufigsten *falschen* Antworten auf diese Frage verweisen auf `times.sty` beziehungsweise `pslatex.sty`. Durch die Nutzung dieser Pakete werden nicht nur gänzlich andere Schriften eingestellt, sondern auch – da diese Pakete veraltet sind – neue Probleme hervorgerufen. Wird wie im Abschnitt 2.3.9 empfohlen Latin Modern als (quasi) identisch aussehender Ersatz für die »Standard- \LaTeX -Schrift« verwendet, kann das Problem der pixeligen Darstellung der Schrift im AR nicht auftreten.

Generelle Informationen zu \LaTeX 2_εs New Font Selection Scheme (NFSS) gibt der `fntguide` [5].

2.3.1 `times.sty`

Das Paket `times.sty` ist veraltet (siehe `psnfss2e` [9]). 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=.92]{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=.92]{helvet}`
`\usepackage{courier}`

Anmerkung zu allen Times/Helvetica-Kombinationen Man kann auch als Schreibmaschinenschrift 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 »Palatino« 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 sollte 0.95 betragen.

Die »Helvetica« ist *nicht* die optimale serifenlose Schrift in Kombination mit der »Palatino«, aber die oben genannte Ersetzung soll sich so weit wie möglich am »alten« Paket orientieren. Ferner war die »Helvetica« lange Zeit die passendste, *freiverfügbare* serifenlose Schrift zur »Palatino«. Mittlerweile stellt die »Vera/Bera Sans« eine – unserer Meinung nach – bessere Alternative dar.

Wer eine (auch ältere¹⁶) CorelDraw®-CD besitzt, kann die »Palatino« auch sehr gut mit den Schriften »Frutiger«¹⁷ oder »Optima«¹⁸ kombinieren. Walter Schmidt hat auf seiner Homepage¹⁹ die entsprechenden T_EX-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` [9].

- ▷ **Ersetze:** `\usepackage{mathpple}` **Durch:** `\usepackage{mathpazo}`

2.3.6 utopia.sty

Das Paket `utopia.sty` sollte nicht mehr verwendet werden, da sein Nachfolger `fourier.sty` ein verbessertes Interface zur Schriftart Utopia bietet. Genaueres siehe `psnfss2e` [9].

- ▷ **Ersetze:** `\usepackage{utopia}` **Durch:** `\usepackage{fourier}`

¹⁶ Die Schriften, die mit älteren Versionen von CorelDraw® ausgeliefert werden, enthalten noch kein Euro-Zeichen!

¹⁷ Bitstream »Humanist 777«, bfr

¹⁸ Bitstream »Zapf Humanist«, bop

¹⁹ Schriften für T_EX: <http://home.vr-web.de/was/fonts>

2.3.7 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` [8] entnehmen.

Die `pifont.sty`-Tricks

▷ **Ersetze:**

```
\usepackage{pifont}
\newcommand{\uppi}{%
  \Pisymbol{psy}{112}}
...
\uppi

beziehungsweise

\newcommand[1]{\upgreek}{%
  \usefont{U}{psy}{m}{n}#1}
...
\upgreek{p}
```

Durch:

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

Der `babel.sty`-Trick

▷ **Ersetze:**

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

Durch:

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

2.3.8 `euler.sty` vs. `eulervm.sty`

Das Paket `euler.sty` sollte durch `eulervm.sty` ersetzt werden, da es Kompatibilitätsprobleme mit anderen Paketen ausräumt und einige Detailverbesserungen enthält:

- `\hbar` (`\hslash` bei dieser Schrift) funktioniert nun sauber
- Fette Mathematikschriften inklusive griechische Symbole sind möglich.

Genauere Informationen siehe `eulervm` [7].

▷ **Ersetze:**

```
\usepackage{euler}
```

Durch:

```
\usepackage{eulervm}
```

2.3.9 `ae.sty`, `aecompl.sty` und `zefonts.sty`

Die Pakete `ae.sty` (zusammen mit `aecompl.sty`) und `zefonts.sty` dienen als Notbehelf für ein Problem im Zusammenhang mit Umlauten und anderen akzentuierten Buchstaben

bei Verwendung der L^AT_EX-Standardschrift Computer Modern (CM). Da diese in der sogenannten OT₁-Kodierung vorliegt und somit akzentuierte Buchstaben durch zwei Zeichen dargestellt werden müssen (etwa »ü« als Kombination von »u« und »¨«), wird bei diesen Buchstaben der Trennalgorithmus von L^AT_EX durcheinandergebracht, was dazu führt, dass beispielsweise Wörter mit Umlauten nicht getrennt werden können. Dieses Problem beheben `ae.sty` und `zefonts.sty` durch die Verwendung sogenannter virtueller Fonts.

Ein weiteres Problem in diesem Zusammenhang wird durch die genannten Pakete jedoch nicht behoben: Wird eine PDF-Datei erzeugt, stehen in dieser immer noch die aus zwei Zeichen zusammengesetzten Buchstaben, was zur Folge hat, dass Wörter mit Umlauten beim Suchen oder Kopieren aus dem Dokument nicht korrekt behandelt werden.

Die hier beschriebenen Probleme treten nicht auf, wenn eine Schrift in T₁-Kodierung und Type-1-Format verwendet wird.²⁰ Dazu wurden in letzter Zeit die Schriftfamilien `cm-super` und `Latin Modern` entworfen. Da Letztere sich stärker an der originalen Computer Modern orientiert, wird empfohlen, die CM durch `Latin Modern` zu ersetzen.

▷ Ersetze:	<code>\usepackage{ae}</code> beziehungsweise <code>\usepackage{zefonts}</code>	Durch:	<code>\usepackage[T1]{fontenc}</code> <code>\usepackage{lmodern}</code>
-------------------	--	---------------	--

²⁰ Man beachte, dass T₁ und Type 1 trotz der ähnlichen Bezeichnung nicht dasselbe bedeuten: T₁ beschreibt die Kodierung der Schrift auf der L^AT_EX-Seite, während Type 1 eine Aussage darüber macht, wie die Buchstaben in der PDF-Datei repräsentiert sind. Somit behebt die Verwendung einer Schrift in T₁-Kodierung das Problem mit der Trennung von Wörtern mit Umlauten, aber nicht notwendigerweise das Problem mit dem Suchen von Umlauten in einer PDF-Datei.

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 `center`-Umgebung verwenden, da diese zusätzlichen vertikalen Abstand einfügt, der meistens nicht erwünscht ist.

▷ **Ersetze:**

```
\begin{figure}
  \begin{center}
    \includegraphics{bild}
  \end{center}
\end{figure}
```

Durch:

```
\begin{figure}
  \centering
  \includegraphics{bild}
\end{figure}
```

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:**

```
\begin{appendix}
  \section{Blub}
\end{appendix}
```

Durch:

```
\appendix
\section{Blub}
```

3.3 Mathematiksatz

Generell sollte man für komplizierteren Mathematiksatz `amsmath.sty`, gegebenenfalls ergänzt durch `mathtools.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 (zum Beispiel `split`) ermöglichen es, lange Gleichungen einfacher zu umbrechen.
- Einfache Möglichkeit, neue Operatoren (ähnlich wie `\sin` usw.) mit sauberen Zeichenabständen zu definieren.

$$\begin{array}{l} a = b \\ b = c \\ a = c \end{array}$$

Abbildung 3: Beispiel für `eqnarray*`

$$\begin{array}{l} a = b \\ b = c \\ a = c \end{array}$$

Abbildung 4: Beispiel für `align*`

Warnung Bei der Verwendung des Paketes `amsmath.sty` sollte man die Umgebungen `displaymath`, `eqnarray` und `eqnarray*` *keinesfalls* weiterverwenden, da diese von `amsmath.sty` nicht unterstützt werden und deshalb wieder inkonsistente Abstände ergäben (siehe die Abbildungen 3 bis 4 auf dieser Seite).

Die `\[...]`-Umgebung wird von `amsmath.sty` korrekt angepasst und kann an Stelle von `displaymath` genutzt werden. `eqnarray` und `eqnarray*` kann in erster Näherung durch `align` beziehungsweise `align*` ersetzt werden. Für eine vollständige Übersicht der Möglichkeiten von `amsmath.sty` sei auf die Dokumentation `amslatex` [1] verwiesen.

▷ **Ersetze:**

```
\begin{eqnarray}
a &=& b\\
b &=& c\\
a &=& c
\end{eqnarray}
```

Durch:

```
\begin{align}
a &= b\\
b &= c\\
a &= c
\end{align}
```

3.4 Die Verwendung von `\graphicspath`

Das beliebte Makro `\graphicspath` sollte aus folgenden Gründen vermieden und durch die Umgebungsvariable `TEXINPUTS` ersetzt werden²¹:

1. Die \TeX -Suche dauert länger, als diese Aufgabe durch die `kpathsea`-Bibliothek lösen zu lassen. (Bei heutigen Prozessoren fällt das nicht mehr so ins Gewicht.)
2. Jedes Bild verbraucht einen Teil des begrenzten \TeX -Speichers *und* gibt ihn während des gesamten Kompilierens nicht mehr frei.

Bei einer Bourne-Shell kann man zum Beispiel folgenden Aufruf

```
[luser@host]> TEXINPUTS=Bildverz: latex datei.tex
```

verwenden, oder in der `$HOME/.profile` zusätzlich

```
export TEXINPUTS=./Bildverz:
```

²¹ Vgl. David Carlisle's Antwort auf Markus Kohms »Bug-Report«: <http://www.latex-project.org/cgi-bin/ltxbugs2html?pr=latex/2618>

eintragen. Bei letzterem werden die Dateien im Bildverz unterhalb des aktuellen Arbeitsverzeichnisses gefunden.

Unter Microsoft Windows (≤ 98) setzt man die Variable mittels

```
set TEXINPUTS=.\Bildverz;
```

beziehungsweise

```
set TEXINPUTS=./Bildverz;
```

für MiKTeX in der `autoexec.bat`. Bei Windows NT4 und Windows 2000 kann man Umgebungsvariablen über Systemsteuerung → System → Umgebung und seit Windows XP über Systemsteuerung → System → Erweitert → Umgebungsvariablen setzen.

Die hier aufgeführten Vorgehensweisen sind Beispiele. Uns ist durchaus bewusst, dass man TEXINPUTS auch auf andere Art und Weise beziehungsweise in anderen Dateien anpassen kann. Genauer ist der Dokumentation des verwendeten Betriebssystems beziehungsweise der TeX-Distribution zu entnehmen.

3.5 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, sind 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 zu diesem Thema²².

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

Bei der Verwendung von KOMA-Script empfiehlt es sich, den dort bereitgestellten Befehl `\renewcaptionname` zu benutzen:

```
\renewcaptionname{ngerman}{\listfigurename}{Abbildungen}
```

(Dieser Befehl muss nach `\begin{document}` ausgeführt werden.)

²² »Meine Versuche, `\figurename`, `\tablename` o.ä. bei Verwendung des Pakets `babel.sty` umzudefinieren, mißlingen. Was kann ich tun?« in <http://projekte.dante.de/DanteFAQ/Verschiedenes>

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

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

^a Nur in den article-Klassen^b Nur in den report- und book-Klassen

A Beispiel zu \sloppy

Hier ist der Beispiel-Code, welchen Markus Kohm veröffentlicht hat:

```
\documentclass{article}

\setlength{\textwidth}{20em}
\setlength{\parindent}{0pt}
\begin{document}
\typeout{First without \string\sloppy\space and underfull \string\hbox}

tatata tatata tatata tatata tatata tatata ta\ -ta\ -tata
tatata tatata tatata tatata tatata tatata tata\ -tata
tatata tatata tatata tatata ta\ -tatatat\ -ta
tatata tatata tatata tatata tatata tatata ta\ -ta\ -ta\ -ta

\typeout{done.}

\sloppy
\typeout{Second with \string\sloppy\space and underfull \string\hbox}

tatata tatata tatata tatata tatata tatata ta\ -ta\ -tata
tatata tatata tatata tatata tatata tatata tata\ -tata
tatata tatata tatata tatata ta\ -tatatat\ -ta
tatata tatata tatata tatata tatata tatata ta\ -ta\ -ta\ -ta

\typeout{done.}
\end{document}
```

Quelle: [Message-ID: <8557097.gEimXdBtjU@ID-107054.user.dfncis.de>](#)

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*. November 2005.
URL: CTAN:macros/latex/required/graphics/.
- [3] MARKUS KOHM und JENS-UWE MORAWSKI: *KOMA-Script*. Juni 2010, Version 3.06.
URL: CTAN:macros/latex/supported/koma-script/.
- [4] THE L^AT_EX₃ PROJECT TEAM: *L^AT_EX 2_ε for class and package writers*. Februar 2006.
URL: CTAN:macros/latex/doc/clsguide.pdf
- [5] THE L^AT_EX₃ PROJECT TEAM: *L^AT_EX 2_ε font selection*. November 2005.
URL: CTAN:macros/latex/doc/fntguide.pdf
- [6] THE L^AT_EX₃ PROJECT TEAM: *L^AT_EX 2_ε for authors*. November 2005.
URL: CTAN:macros/latex/doc/usrguide.pdf
- [7] WALTER SCHMIDT: *The Euler Virtual Math Fonts for use with L^AT_EX*. Januar 2005, Version 4.0.
URL: CTAN:fonts/eulervm/
- [8] WALTER SCHMIDT: *The upgreek package for L^AT_EX 2_ε*. Februar 2003, Version 2.0.
URL: CTAN:macros/latex/contrib/was/.
- [9] WALTER SCHMIDT: *Using common PostScript fonts with L^AT_EX*. September 2004, PSNFSS Version 9.2.
URL: CTAN:macros/latex/required/psnfss/psnfss2e.pdf
- [10] WALTER SCHMIDT, JÖRG KNAPPEN, HUBERT PARTL und IRENE HYNÄ: *L^AT_EX 2_ε-Kurzbeschreibung*. April 2003, Version 2.3.
URL: CTAN:info/lshort/german/.
- [11] AXEL SOMMERFELD: *Anpassen der Abbildungs- und Tabellenbeschriftungen mit Hilfe des caption-Pakets*. April 2008.
URL: CTAN:macros/latex/contrib/caption/.
- [12] NICOLA L. C. TALBOT, *Upgrading from the glossary package to the glossaries package*. April 2011.
URL: CTAN:macros/latex/contrib/glossaries/glossary2glossaries.pdf.