Layout-Änderungen mit LaTeX

EDV-Zentrum der TU Wien, Abt. Digitalrechenanlage
Hubert Partl
1988-10-04
*Axel Kielhorn**
1996-01-01
H27.0 — Version 1

Zusammenfassung

Dieses Dokument beschreibt die Möglichkeiten der `refart` und `refrep` Class für LaTeX 2ε. Sie arbeitet nicht mehr mit dem alten LaTeX 2.09 zusammen und enthält einige Verbesserungen gegenüber dem Original. Alte `REFMAN` Quellen sollten sich aber problemlos an die Class anpassen lassen.

Dieses Heft ist als Ergänzung zu Kapitel 5 ("Designing It Yourself") des LaTeX-Manuals von Leslie Lamport gedacht.

Inhaltsverzeichnis

1 Einleitung **3**

1.1 Autor, Layout-Designer und Setzer 3

1.2 Layout-Änderungen 3

2 Die Kunst des Layout-Design **4**

2.1 Allgemeine Regeln 4

2.2 Spezielle Hinweise für technische Beschreibungen 5

3 Wie kann man das LaTeX-Layout ändern? **6**

3.1 Vor und Nachteile des Textsatzsystems LaTeX 6

3.2 Eingabe-Dateien und Class-Dateien 6

3.3 Class-Files und Package-Files 7

3.4 Vorgangsweise bei Layout-Änderungen 7

3.5 Ein einfaches Beispiel (Gleichungsnummern) 9

3.6 Ein etwas komplexeres Beispiel (Reference Manual) 10

Anhang **13**

* A.Kielhorn@web.de

A Die Seitenaufteilung in LaTeX **13**

B Beschreibung der Dokumentklasse Refman **16**

B.1 Aufruf 16

B.2 Optionen 16

B.3 Layout-Änderungen 17

B.4 Fußnoten 18

B.5 Zusätzliche Befehle 19

1 Einleitung

1.1 Autor, Layout-Designer und Setzer

Jedes Schriftstück – von einer kurzen handgeschriebenen Notiz bis zu einem gedruckten Buch –besteht aus zwei Komponenten: **Inhalt** und **Form.** In den meisten Fällen ist die Form nicht Selbstzweck sondern dient nur dazu, den Inhalt für den Leser verständlich darzustellen.

Autor: Der **Inhalt** wird vom jeweiligen Autor geliefert. Bei Druckwerken wird der Autor meist durch den Lektor des Verlages oder den Herausgeber der Zeitschrift unterstützt.

Bei der Verwendung des Textsatzsystems LaTeX wird der Inhalt im Hauptteil des LaTeX-Eingabefiles angegeben – also zwischen `\begin{document}` und `\end{document}` – ergänzt durch LaTeX-Befehle, die angeben, welche Bedeutung die einzelnen TextTeile haben.

Designer: Die **Form** eines Druckwerks wird üblicherweise vom Verleger festgelegt, der das Layout von den im Verlag angestellten Designern entwerfen läßt.

Die mit LaTeX mitgelieferten sogenannten „Document Classes“ wurden von professionellen amerikanischen Layout-Designern für die von Leslie Lamport vorgesehenen Arten von Schriftstücken entworfen, also für wissenschaftliche Zeitschriften, Berichte und Bücher – auf der Grundlage der in den USA üblichen Konventionen. Die „Koma-Script- Classes“ sind an den europäischen Geschmack angepaßt und in vielen Kleinigkeiten optimiert worden. Diese Layouts sind in den sogenannten „Class-Files“ festgelegt. Am Beginn des LaTeX-Eingabefiles – vor dem Befehl `\begin{document}` – wird angegeben, in welcher Form der Inhalt des Schriftstücks gesetzt und ausgedruckt werden soll.

Setzer: Der Setzer in der Druckerei bringt den vom Autor bzw. Lektor gelieferten Inhalt in die vom Layout-Designer festgelegte Form, in der das Schriftstück dann schließlich ausgedruckt wird.

Bei der Verwendung von LaTeX erfolgt das Setzen durch das Computer-Programm TeX und das Ausdrucken durch ein weiteres Computer-Programm, den sogenannten Device-Driver.

1.2 Layout-Änderungen

Wenn das Schriftstück nicht in einem der vorgefertigten LaTeX-Layouts gesetzt werden soll, sind die folgenden zwei Voraussetzungen notwendig:

1. Das Layout muß entworfen werden. Dies sollte im allgemeinen von einem professionellen Designer gemacht werden. Falls kein solcher zur Verfügung steht, muß sich der Autor oder der Herausgeber als „Amateur-Designer“ versuchen – am besten durch Anlehnung an professionell gestaltete Vorbilder.

→ 2 Ausführlichere Hinweise dazu werden in Kapitel 2 angegeben.

2. Das Layout muß in Form eines LaTeX-Class-Files formuliert werden. Dies kann entweder als „Class-Option“ geschehen, oder durch die Definition einer neuen „Class“. Die komplette Neudefinition hat den Vorteil, das sie weitreichendere Änderungen erlaubt. Ich habe mich daher bei der Portierung nach LaTeX 2_ε für eine eigene „Class“ entschieden. Dadurch wird die Übertragung von alten Dateien zwar etwas erschwert, aber meiner Meinung nach überwiegen die Vorteile. Die Angabe dieser „Class“ bzw. dieser „Class-Option“

am Beginn des LaTeX-Eingabefiles bewirkt dann, daß der Inhalt des Schriftstücks in dieser neuen Form gesetzt und gedruckt wird.

→ 3 Ausführlichere Hinweise dazu werden in Kapitel 3 angegeben.

2 Die Kunst des Layout-Design

2.1 Allgemeine Regeln

Es gibt fast keine allgemeingültigen Regeln, sondern man muß für jedes Schriftstück oder jede Reihe von Schriftstücken ein eigenes Layout entwerfen.

Dieses Layout muß sich vor allem daran orientieren, *wer* das Schriftstück *wie* lesen wird.

Ein wichtiges Kriterium ist z. B., ob der Leser es eher von A bis Z durchgehend lesen will wie einen Kriminalroman oder ein „Tutorial“, oder ob er eher bestimmte Stellen rasch auffinden will wie in einem Telefonbuch oder in einem „Reference Manual“.

Außerdem muß das Layout die bestehenden Konventionen berücksichtigen, vor allem die Gewohnheiten des Lesers, aber auch den sogenannten „House Style“, durch den sich verschiedenartige Schriftstücke und die Produkte verschiedener Hersteller oder Institutionen voneinander unterscheiden sollen.[1]

! → Der Hauptzweck des Layouts ist stets, daß der Leser die gesuchten Informationen gut findet und sie gut lesen kann, und daß er den Inhalt und den logischen Aufbau des Schriftstücks richtig versteht. Wichtig sind also Übersichtlichkeit, Lesbarkeit, Konsistenz und dergleichen. Die Schönheit (Ästhetik) steht demgegenüber im allgemeinen nur an zweiter Stelle.[2]

Im folgenden werden ein paar einfache „Faustregeln“ für den Seitenaufbau angeführt, die für die meisten Anwendungen Gültigkeit haben.

Zeilenabstände : Die Zeilenabstände müssen größer als die Wortabstände sein, damit die Augen des Lesers richtig entlang der Zeilen geführt werden.

Zeilenlänge : Die Zeilenlänge – bzw. bei mehrspaltigem Druck die Spaltenbreite – soll möglichst bei etwa 60 Zeichen pro Zeile liegen. Zu lange Zeilen sind schlecht lesbar, weil sie anstrengende Augenbewegungen erfordern und gleichzeitig die Wahrscheinlichkeit erhöhen, beim Zeilenwechsel in die falsche Zeile zu geraten. Zu kurze Zeilen machen Probleme beim Randausgleich (Blocksatz) und beim Abteilen (Silbentrennung). Wenn längere Zeilen notwendig sind, dann müssen die Zeilenabstände vergrößert werden.

Seitenaufbau : Die normalen Textseiten sollen einen möglichst einheitlichen Aufbau haben. Abbildungen, Tabellen und spezielle Seiten wie Inhaltsverzeichnis, Index usw. sollen aber *nicht* in dasselbe Schema gepreßt werden, sondern stets so viel Platz einnehmen, wie für die jeweilige Tabelle etc. am günstigsten ist (also durchaus kürzer oder länger als die normale Zeilenlänge, evtl. mit einem Rahmen darum herum). Wenn möglich sollten bei der Seitenaufteilung auch ästhetische Grundregeln wie z. B. der „Goldene Schnitt“ ($13 : 8 \approx 8 : 5$) angewendet werden.

[1] Man vergleiche z. B. das jeweils typische Layout verschiedener Fachzeitschriften oder Tageszeitungen.

[2] Dies gilt natürlich nicht für Werbeprospekte, Bildbände und dgl.

Randnoten : „Randnoten“ sind für viele Zwecke besser geeignet als Fußnoten, weil sie direkt neben dem Text erscheinen, auf den sie sich beziehen. Dazu zählen auch Symbole wie z. B. „Achtung-Zeichen“, die wichtige Text-Teile markieren bzw. den Leser auf sie hinweisen.

Kopf- und Fußzeilen : Der Inhalt von Kopf- und Fuß-Zeilen soll die Orientierung innerhalb des Schriftstücks erleichtern. Falls zu erwarten ist, daß manche Leser einzelne Seiten aus dem Schriftstück herauskopieren (z. B. einen Artikel aus einer Zeitschrift), sollte in der Kopf- oder Fußzeile auch die Information zu finden sein, aus welchem Schriftstück die Seiten stammen. Falls häufig neue Versionen eines Schriftstücks erstellt werden (z. B. bei Software-Manuals), sollte auch das Datum oder die Versionsnummer in der Kopf- oder Fußzeile erscheinen.

2.2 Spezielle Hinweise für technische Beschreibungen

Wir wollen kurz die Eignung von drei verschiedenen Layouts für den an unserer Universität häufigen Spezialfall von technischen Beschreibungen, Users' Guides, Reference Manuals, Mitteilungsblättern und ähnlichen technischen Druckwerken vergleichen:[3]

Plain TEX : Das Standardformat von Plain TEX hat vor allem den großen Nachteil, daß die Zeilen viel zu lang sind, was sich schlecht auf die Lesbarkeit und Übersichtlichkeit auswirkt. Dieses von Schreibmaschinenschrift gewohnte Schriftbild eignet sich nicht für die von TEX unterstützte „Druck“-Schrift. Die an der TU Wien vor der Einführung von LATEX entwickelten Macros `EASY.TEX` und `ARTii.TEX` bzw. `REPii.TEX` haben außerdem einige weitere Design-Mängel, z. B. sind die Abstände zwischen den Kapitel-Überschriften zu groß.

Standard-LATEX : Auch das Standardformat von LATEX ist für diese Zwecke nicht ideal. Die Zeilenlänge ist zwar richtig gewählt, aber die Seiten enthalten einen zu großen ungenützten Rand, insbesondere oben und unten. Außerdem werden für die Kapitel-Überschriften etwas zu große Schriften verwendet.

Reference-Manual-Style : Ein für solche Zwecke günstigeres Seiten-Layout findet man bei einigen in den letzten Jahren neu erschienenen Reference Manuals:[4]

- Der fortlaufenden Text steht mit einer relativ kurzen, gut lesbaren Zeilenlänge im rechten Teil der Seite. Dieser Teil dient also zum fortlaufenden Lesen einzelner Abschnitte.
- Im breiten linken Rand jeder Seite stehen die Überschriften sowie Stichworte und andere „Randnoten“. Dieser Teil dient also zum raschen Auffinden von bestimmten Informationen innerhalb des gesamten Schriftstücks.
- Abbildungen und Tabellen stehen je nach ihrem Platzbedarf in der einen oder anderen Spalte oder gehen über die gesamte Seitenbreite.

→ 3.6 Eine Anleitung, wie man ein solches Layout mit LATEX realisieren kann, wird in Abschnitt 3.6 angegeben.

[3] Diese Hinweise stammen von Paul Stiff, der Layout-Design an der University of Reading unterrichtet.

[4] Ein häufig zitiertes – allerdings in den konkreten Details auch nicht ganz ideales – Beispiel ist das „PostScript Reference Manual“.

3 Wie kann man das LaTeX-Layout ändern?

3.1 Vor und Nachteile des Textsatzsystems LaTeX

Vorteile: Der große Vorteil von LaTeX besteht darin, daß es ein „generisches“ oder „logisches“ Design unterstützt. Darunter versteht man, daß der Autor nur mit einigen wenigen Befehlen angeben muß, welche *Bedeutung* die einzelnen Text-Teile haben: Überschriften, Zitate, mathematische Gleichungen, Listen, Aufzählungen, Literaturangaben usw. Diese logischen Begriffe werden dann vom System automatisch in der richtigen Art und Weise gesetzt und ausgedruckt. Was „richtig“ ist, muß dem System vorher in geeigneter Weise mitgeteilt werden, bei LaTeX durch die Angabe der „Document Class“ und eventuell weiterer „Packages“.

Im Gegensatz dazu steht das von älteren Textverarbeitungsprogrammen und auch von Plain TeX unterstützte „visuelle“ Design, bei dem der Autor die sichtbaren Eigenschaften wie Schriftart, Schriftgröße, vertikale Abstände, horizontale Einrückungen usw. angeben muß.

Das Prinzip des logischen Design macht einerseits die Anwendung durch den Autor wesentlich einfacher und übersichtlicher und stellt andererseits die logische Konsistenz des Schriftstücks sicher (z. B. einheitliche Schriftart für alle Überschriften des gleichen Levels, einheitliche Numerierung von Kapiteln und von Gleichungen, einheitliche Darstellung von Listen und Aufzählungen, usw.).

Nachteile: Als Nachteil von LaTeX wird immer wieder angeführt, daß es den Autor zu sehr einschränkt und daß man das Layout zu wenig ändern kann. Dies ist nur teilweise richtig, nämlich nur dann, wenn man sich auf die vier mit LaTeX mitgelieferten Standard-Layouts beschränkt, die sich selbstverständlich nicht für alle Anwendungen eignen und die außerdem sehr stark an amerikanische Konventionen angepaßt sind.

LaTeX ist aber wesentlich mächtiger und flexibler: Man kann praktisch beliebige Änderungen am Layout dadurch erreichen, daß man die in den Style-Files enthaltenen Definitionen ändert, die geänderten Definitionen in privaten Style-Option-Files speichert und diese dann als zusätzliche Style-Options aufruft. Der Aufwand dafür ist keineswegs größer als bei der Verwendung von Plain TeX, in manchen Fällen sogar wesentlich geringer.

3.2 Eingabe-Dateien und Class-Dateien

In Einklang mit dem Prinzip des logischen Design und der Trennung von Inhalt und Form verwendet LaTeX grundsätzlich zwei Arten von Dateien:

- Der Inhalt einschließlich der logischen Struktur des Schriftstücks wird in den LaTeX-Eingabefiles festgelegt.
- Die Form (das Layout) wird in sogenannten Class-Files und eventuell auch Packages festgelegt.

Welche Class- und Package-Files ein Schriftstück verwendet – d. h. in welchem Layout es gesetzt werden soll – wird am Beginn des Eingabefiles mit den Befehlen `\documentclass` bzw. `\usepackage` angegeben.

Für die Erzeugung eines Schriftstücks mit LaTeX sind also zumindest zwei Files notwendig: ein Eingabefile und ein Class-File.

Diese Trennung entspricht den in der Einleitung erläuterten unterschiedlichen Funktionen von Autor und Layout-Designer. Sie hat aber auch für den Fall, daß der Autor selbst das Layout festlegt, wesentliche Vorteile:

- Einerseits wird damit sichergestellt, daß zusammengehörige Schriftstücke automatisch im selben Layout gesetzt werden, auch wenn dieses Layout nach dem Erstellen des Textes noch geändert wurde. Dies wäre nur schwer zu erreichen, wenn die Layout-Definitionen direkt in jedem einzelnen Eingabefile enthalten wären.
- Andererseits ist es auf diese Weise ohne großen Aufwand möglich, denselben Inhalt in verschiedenen Layouts auszudrucken, z. B. als Artikel für die eine oder andere wissenschaftliche Zeitschrift oder als Kapitel einer Dissertation und eines Forschungsberichts.

3.3 Class-Files und Package-Files

LaTeX unterstützt innerhalb der Layout-Definition die folgende Hierarchie von „Class“ und „Option“, die sich in den verschiedenen Parametern des Befehls `\documentclass` widerspiegelt:

- Als erstes wird der Inhalt desjenigen „Class-Files“ verarbeitet, dessen Name im `\documentclass`-Befehl zwischen den geschwungenen Klammern angegeben wurde. Dies gibt die grundsätzliche Art des Schriftstücks an.
- Dann werden der Reihe nach diejenigen „Class-Option-Files“ verarbeitet, deren Namen zwischen den eckigen Klammern angegeben wurden. Damit kann man verschiedene Layout-Varianten auswählen.
- Als dritter Schritt werden dann die im `\usepackage`-Befehl angegebenen Packages geladen, die ihrerseits wieder Optionen laden können.

Diese mehrstufige Verarbeitung macht es möglich, die Layout-Definition in mehrere kleinere Einheiten zu zerlegen, wobei in einem Class-Option-File sowohl Layout-Definitionen der ursprünglichen Class-Files abgeändert als auch weitere Definitionen hinzugefügt werden können.

Class-Files, Class-Option-Files und Packages unterscheiden sich von „normalen“ Eingabefiles im wesentlichen durch die folgenden Punkte:

- Sie dürfen nur Definitionen enthalten, aber keine Text-Ausgaben bewirken.
- Das Sonderzeichen „Klammeraffe“ (`@`) hat in ihnen die Bedeutung eines Buchstabens („letter“), d. h. es ist innerhalb von Befehlsnamen erlaubt. Die meisten von LaTeX intern verwendeten Befehle enthalten Klammeraffen in ihren Namen, damit Kollisionen mit den vom Benutzer verwendeten Befehlen ausgeschlossen werden.
- Die Filenamen haben die „Extension“ `cls` bzw. `clo` oder `sty` für Packages (nicht `tex`).

3.4 Vorgangsweise bei Layout-Änderungen

3.4.1 Festlegen der Änderungen gegenüber dem Original-Layout

Im allgemeinen ist es günstiger, die Class-Files für das neue Layout nicht von Null an neu zu erstellen, sondern die bereits bestehenden LaTeX-Class-Files als Vorbilder zu verwenden und so viel wie möglich von den darin enthaltenen Definitionen zu übernehmen.

In den meisten Fällen genügt es sogar, ein mehr oder weniger kurzes Package-File zu erstellen, das nur die Unterschiede des neuen Layouts gegenüber einem der originalen LaTeX-Layouts enthält, und diese „Option“ dann in Verbindung mit der originalen „Class“ zu verwenden.

Im ersten Schritt legt man daher fest, in welchen Bereichen sich das neue Layout vom originalen LaTeX-Layout unterscheidet und worin diese Unterschiede im einzelnen bestehen. Man geht dabei am besten so vor, daß man sich für alle als LaTeX-Befehle oder -Environments definierten logischen Einheiten überlegt, in welchem Layout sie gesetzt werden sollen. Neue zusätzliche LaTeX-Befehle sollte man nach Möglichkeit *nicht* erfinden – und wenn, dann nur im Einklang mit dem Prinzip des logischen Design und mit der Struktur der anderen LaTeX-Befehle (z. B. neue Environments für spezielle Tabellen, oder eigene Befehle für spezielle Hervorhebungen).

3.4.2 Finden der Original-Definitionen

Im zweiten Schritt muß man finden, wo und wie alle die Layout-Eigenschaften, die man ändern will, im Original definiert sind. Dazu soll man die folgenden Quellen in der folgenden Reihenfolge durchsuchen:[5]

1. das LaTeX-Manual von Leslie Lamport,
2. die Dokumentations-Files `*.dtx`
3. die Kernal-Files `*.ltx`,
4. das TeXbook von Donald E. Knuth.

Die beiden Bücher sind im Buchhandel erhältlich (Verlag Addison Wesley). Ob und wo die Files an Ihrem Computer angelegt sind, ist im „LaTeX Local Guide“ Ihrer Installation angegeben. Die Files sind sehr gut kommentiert, sodaß man sich im allgemeinen auch dann gut in ihnen zurechtfindet, wenn man nicht alle Details der darin enthaltenen Befehle versteht.

3.4.3 Schreiben eines neuen Package-Files

Im dritten Schritt erzeugt man das neue Package-File. Man wählt einen Namen für die neue Package (z. B. `refman`) und bildet den Filenamen aus diesem Namen und der Extension `sty` (z. B. `refman.sty`). Ausgerechnet für `refman` trift diese Vorgehensweise nicht zu, da es sich inzwischen um eine eigenständige Class handelt.

Dieses File muß nur diejenigen Definitionen (`\def`-Befehle, Wertzuweisungen und dergleichen) enthalten, die sich gegenüber der Originalversion unterscheiden, und/oder diejenigen, die man zusätzlich definieren will.

Bei den Änderungen geht man am besten so vor, daß man die betreffenden `\def`-Befehle, Wertzuweisungen usw. aus den originalen Files in das neue File kopiert und dann dort entsprechend modifizert.

Auch bei neuen Definitionen lehnt man sich am besten an Vorbilder in den originalen Files an.

Außerdem muß man den Zweck des Files, seine Verwendung, den Autor und das letzte Änderungsdatum sowie alle im File enthaltenen Definitionen auf Kommentarzeilen innerhalb des Files dokumentieren. Bei sehr

[5] Diese Hinweise stammen von Sue Brooks, die im Rahmen der TeX-Konferenz 1988 in Exeter ein Workshop für „LaTeX-Hacker“ leitete.

umfangreichen Style-Files kann man dazu das `docstrip`-Programm benutzen, mit dem auch alle LaTeX 2ε Dateien dokumentiert wurden. Das Script (`*.ins`) erzeugt aus dem Dokument (`*.dtx`) dann eine Package (`*.sty`) oder Class (`*.cls`). Die `dtx`-Datei dient gleichzeitig als Dokumentation. Dadurch bleibt gewährleistet das Package und Dokumentation übereinstimmen. In den meisten Fällen genügt es aber, nur ein `sty`-File mit allen Kommentaren anzulegen.

3.4.4 Verwendung der neuen Style-Option

Durch die Angabe des neuen Package-Namens im `\usepackage`-Befehl werden die Änderungen für das betreffende Schriftstück wirksam, d. h. dieses wird im neuen Layout gesetzt. Beispiel:

```
%Alte Version:
\documentstyle[11pt,twoside,german,refman]{article}
```

wird zu:

```
%Neue Version: (LaTeX2e)
\documentclass[11pt,twoside,a4paper]{refart}
\usepackage{german}
%\usepackage{mysty} %<- Hier wird meine Package geladen
```

Davon abgesehen sollten keine Änderungen im Eingabefile notwendig sein – es sei denn, das Schriftstück enthält spezielle logische Text-Elemente, die im neuen Layout vorgesehen sind, aber in Original-LaTeX nicht unterstützt werden.

! → Wenn man das Eingabefile an einen anderen Computer übertragen will, muß man *alle* darin verwendete Package-Files mitübertragen, damit es auch dort verarbeitet werden kann.

3.5 Ein einfaches Beispiel (Gleichungsnummern)

Angenommen, Sie wollen, daß in einem „Article“ die Gleichungen innerhalb jeder „Section“ separat numeriert werden.

Im LaTeX-Manual finden Sie den Hinweis, daß der „Report“-Style etwas Ähnliches pro „Chapter“ macht.

Sie schauen das File `report.cls` an und finden darin die folgenden zwei Befehle, die sich – wie in den Kommentarzeilen angegeben ist – auf die Gleichungsnummern beziehen:

```
\@addtoreset{equation}{chapter}
\def\theequation{\thechapter .\arabic{equation}}
% In LaTeX2e ab 1995/06/01:
\renewcommand\theequation{\thechapter.\@arabic\c@equation}
```

Es ist nicht notwendig, daß sie diese beiden Befehle in allen Details verstehen, es genügt, wenn Sie erkennen, daß sich `chapter` in beiden Fällen auf die Numerierung pro Kapitel bezieht.

Sie legen ein File mit dem Namen `eqpersec.sty` an,[6] kopieren die Befehle in dieses File, ersetzen darin konsequent `chapter` durch `section` und erhalten also die neuen Definitionen

[6] Je nach dem Computer, an dem Sie arbeiten, kann der Filename eine andere Syntax haben, z. B. `EQPERSEC_STY` an einer CYBER unter NOS/VE.

```
% Dies ist equation_per_section.sty
% Kurzname: eqpersec.sty
%
% Gleichungszaehler am Anfang einer section
% zuruecksetzen.
\@addtoreset{equation}{section}
% Gleichungsnummer = sectionnummer.equationnummer
\def\theequation{\thesection .\arabic{equation}}
% oder:
\renewcommand\theequation{\thesection.\@arabic\c@equation}
```

Natürlich fügen Sie auch noch Kommentarzeilen hinzu, die den Zweck des Files erklären und Ihren Namen als den Autor dieses neuen Style-Option-Files sowie das Änderungsdatum angeben.

Wann immer Sie nun in Ihrem LaTeX-Eingabefile die Package `eqpersec` angeben, z. B. mit

```
\documentclass[11pt]{article}
\usepackage{eqpersec}
```

dann werden die Gleichungen in Ihrem Artikel nach dieser neuen Konvention numeriert.

3.6 Ein etwas komplexeres Beispiel (Reference Manual)

Wir wollen ein ähnliches Seiten-Layout wie beim PostScript Reference Manual erreichen: mit einem breiten linken Rand, der für Überschriften und Randnoten benützt wird, und mit nur einem eher knappen freien Rand oben, rechts und unten.

Dies scheint eine sehr grundlegene Änderung zu sein. Durch die gute Modularität von LaTeX sind aber auch für eine solche Aufgabe nur relativ wenige Änderungen notwendig, die im folgenden komplett beschrieben werden.

3.6.1 Seitenaufteilung

Die neue Seitenaufteilung wird im wesentlichen mit den im LaTeX-Manual beschriebenen Parametern festgelegt. In diesem Beispiel verwenden wir dazu die Plain-TeX-Befehle `\newdimen`, Wertzuweisung und `\advance`, wir könnten aber genausogut auch die entsprechenden LaTeX-Befehle `\newlength`, `\setlength` und `\addtolength` verwenden.

Horizontal: Zunächst definieren wir zwei neue Bezeichnungen für Größen, die im folgenden öfter verwendet werden:

`\fullwidth` gibt die komplette Breite (Textbreite + genützter linker Rand) an und erhält denselben Wert wie in Plain TeX, nämlich 6.5 Inch, sodaß am Papier rechts und links noch jeweils 1 Inch freier Rand bleiben. (Dies ist jetzt natürlich von der Papiergröße abhängig)

`\leftmarginwidth` gibt die Breite des Randes an, der links von der normalen Textbreite für Überschriften und Randnoten verwendet wird:

$$\texttt{leftmarginwidth} = \texttt{fullwidth} - \texttt{textwidth}$$

Da die entsprechenden Maße in Abhängigkeit von der Papierbreite berechnet werden, möchte ich hier auf eine genaue Beschreibung verzichten.

Diese Class wurde mit dem `docstrip`-Programm erstellt und erlaubt es eine ausführliche Dokumentation zu erzeugen. (Allerdings in Englisch)

Vertikal: Das vertikale Layout wird abhängig von der Papierhöhe berechnet. Auch hier möchte ich auf die Dokumentation zur Class verweisen.

3.6.2 Kapitel-Überschriften

Die Überschriften von Sections, Sub- und Subsubsections sollen so abgeändert werden, daß sie in den linken Rand hinausragen.

Im File `classes.dtx` finden wir, daß diese Überschriften durch Aufrufe des Befehls `\@startsection` definiert sind, dessen Funktion auf den Kommentarzeilen beschrieben ist. Für uns relevant sind nur folgende Parameter: Der Absolutbetrag des vierten und des fünften Parameters gibt den Abstand vor bzw. nach der Überschrift an. Der sechste Parameter gibt an, wie der Text der Überschrift gesetzt werden soll. Der Befehl für Sections ist z. B. im Original so definiert:

```
\newcommand\section{\@startsection
             {section}{1}{\z@}%
             {-3.5ex plus -1ex minus -.2ex}%
             {2.3ex plus .2ex}%
             {\normalfont\Large\bfseries}}
```

und analog die Befehle für Sub- und Subsubsections. Die Anpassung an die jeweilige Schriftgröße erfolgt automatisch durch die Verwendung der relativen Einheit `ex`.

Wir definieren zunächst eine Abkürzung `\secshape` für das neue Layout aller dieser Überschriften. Die wesentliche Änderung besteht darin, daß die Überschriften um die Breite des linken Randes weiter links als der normale Text beginnen sollen. Zu diesem Zweck setzen wir den linken Rand auf die entsprechende negative Größe. Außerdem wollen wir, daß die Wörter in der Überschrift nicht abgeteilt werden und daß deshalb rechts kein Randausgleich gemacht wird. Dies erreichen wir mit den folgenden Befehlen, die im TEXbook beschrieben sind:

```
\def\secshape{\leftskip=-\leftmarginwidth
              \rightskip=0pt plus 1fil
              \hyphenpenalty=2000}
```

Ein Aufruf dieses Befehls soll nun jeweils in den sechsten Parameter von `\@startsection` eingefügt werden.

Da die Überschriften durch das Hinausragen in den linken Rand deutlich genug vom normalen Text abgesetzt sind, müssen sie nicht durch so große Schrift und Abstände wie im Original hervorgehoben werden. Wir verkleinern daher auch noch die entsprechenden Angaben im vierten, fünften und sechsten Parameter und erhalten schließlich die neue Definition

```
\newcommand\section{\@startsection
             {section}{1}{\z@}%
             {-2ex plus -1ex minus -.2ex}%
             {0.5ex plus .2ex}%
             {\secshape\normalfont\large\bfseries}}
```

und analoge neue Definitionen für Sub- und Subsubsections.

3.6.3 Positionierung der Randnoten

Die Positionierung der Randnoten muß so abgeändert werden, daß *alle* Randnoten in den linken Rand gesetzt werden – unabhängig davon, ob es sich um eine rechte oder linke Seite und um eine normale oder verkehrte Randnote handelt.

Im File `latex.dtx` finden wir die viele Zeilen lange Definition des Befehls `\@addmarginpar`, der das Setzen der Randnoten durchführt. Wir brauchen uns nicht um alle Details dieses komplizierten Befehls zu kümmern, wir müssen nur eines herausfinden: Im ersten Teil des Befehls wird eine interne Größe `\@tempcnta` entweder auf `\@ne` oder auf `\m@ne` gesetzt, je nachdem, ob die Randnote rechts oder links vom Text gesetzt werden soll. Dazu dienen die folgenden Zeilen.

```
\@tempcnta\@ne
\if@twocolumn
    \if@firstcolumn \@tempcnta\m@ne \fi
\else
  \if@mparswitch
     \ifodd\c@page \else\@tempcnta\m@ne \fi
  \fi
  \if@reversemargin \@tempcnta -\@tempcnta \fi
\fi
```

Diese Zeilen ersetzen wir also einfach durch die Zuweisung

```
\@tempcnta\m@ne
```

Die restlichen Zeilen, die das eigentliche Setzen der Randnote je nach dem Wert von `\@tempcnta` an die richtige Stelle der Seite bewirken, lassen wir unverändert.

3.6.4 Erweiterungen

Die hier angeführten Definitionen genügen durchaus für einfache Anwendungen. In der Praxis können allerdings noch einige Erweiterungen nützlich sein. Ein Beispiel für eine solche „komplette“ Style-Option ist in Anhang B beschrieben.

→ B

Anhang

A Die Seitenaufteilung in LaTeX

In diesem Anhang wird angegeben, aus welchen Komponenten der Seitenaufbau besteht und wie die einzelnen Parameter in LaTeX definiert sind (siehe auch Abbildung 1, die von Nelson Beebe an der University of Utah erzeugt wurde).

→ Abb. 1

Textbereich : Der normale Textbereich („Body“) umfaßt den laufenden Text einschließlich Fußnoten, Tabellen und Abbildungen. *Nicht* zu diesem Bereich gehören Kopfzeilen, Fußzeilen und Randnoten.

Der Textbereich hat die Breite `\textwidth` und die Höhe `\textheight`.

Bei zweispaltiger Ausgabe enthält der Textbereich zwei Spalten, die jeweils `\columnwidth` breit sind, und zwischen ihnen einen Abstand der Breite `\columnsep`. Die Spaltenbreite `\columnwidth` ist also etwas geringer als die Hälfte von `\textwidth`.

`\textwidth` bzw. `\columnwidth` soll ein Vielfaches der Buchstabenbreite des `tt`-Fonts sein.

`\textheight` soll ein Vielfaches der Zeilenhöhe `\baselineskip`, vermehrt um die Konstante `\topskip`, sein.

Innerhalb des Textbereichs werden Einrückungen mit `\leftskip` und `\rightskip` festgelegt. Diese beiden Größen sollen nicht explizit verändert werden sondern nur implizit durch die Environments.

linker Rand : Der Rand links vom Textbereich ist um 1 Inch mehr als `\odd-` bzw. `\evensidemargin`, je nachdem ob die Seitennummer ungerade oder gerade ist. Normalerweise sind beide Größen gleich, nur bei der Option `twoside` werden rechte (=ungerade) und linke (=gerade) Seiten verschieden gesetzt.

oberer Rand : Der Rand oberhalb des Textbereichs ist um 1 Inch mehr als die Summe von `\topmargin`, `\headheight` und `\headsep`.

rechter Rand : Der Rand rechts vom Textbereich ergibt sich aus der Papierbreite minus der Breite von linkem Rand und Textbereich.

unterer Rand : Der Rand unterhalb des Textbereichs ergibt sich aus der Papierhöhe minus der Höhe von oberem Rand und Textbereich.

Kopfzeilen : Kopfzeilen („Header“) stehen innerhalb des oberen Randes, und zwar so, daß zwischen dem unteren Rand der Kopfzeile und dem oberen Rand des Textbereichs der Abstand `\headsep` bleibt. Oberhalb der Kopfzeile bleibt ein freier Rand von 1 Inch mehr als `\topmargin`.

Fußzeilen : Fußzeilen („Footer“) stehen innerhalb des unteren Randes, und zwar so, daß der Abstand vom unteren Rand des Textbereichs zum unteren Rand der Fußzeile `\footskip` ist.

Randnoten : Randnoten („Margin Notes“) stehen innerhalb des rechten oder linken Randes. Sie haben eine Breite von `\marginparwidth`, und zwischen den Randnoten und dem Textbereich bleibt ein horizontaler Abstand von `\marginparsep`. Der vertikale Abstand zwischen aufeinanderfolgenden Randnoten ist mindestens `\marginparpush`.

Die Papierhöhe setzt sich also von oben nach unten aus folgenden Elementen zusammen:

1 Inch
`\topmargin`
`\headheight`
`\headsep`
`\textheight`
`\footskip`
Rest der Seite

Bei Seiten mit rechten Randnoten – was der Standardfall ist – setzt sich die Papierbreite von links nach rechts aus folgenden Elementen zusammen:

1 Inch
`\oddsidemargin` bzw. `\evensidemargin`
`\textwidth`
`\marginparsep`
`\marginparwidth`
Rest der Seite

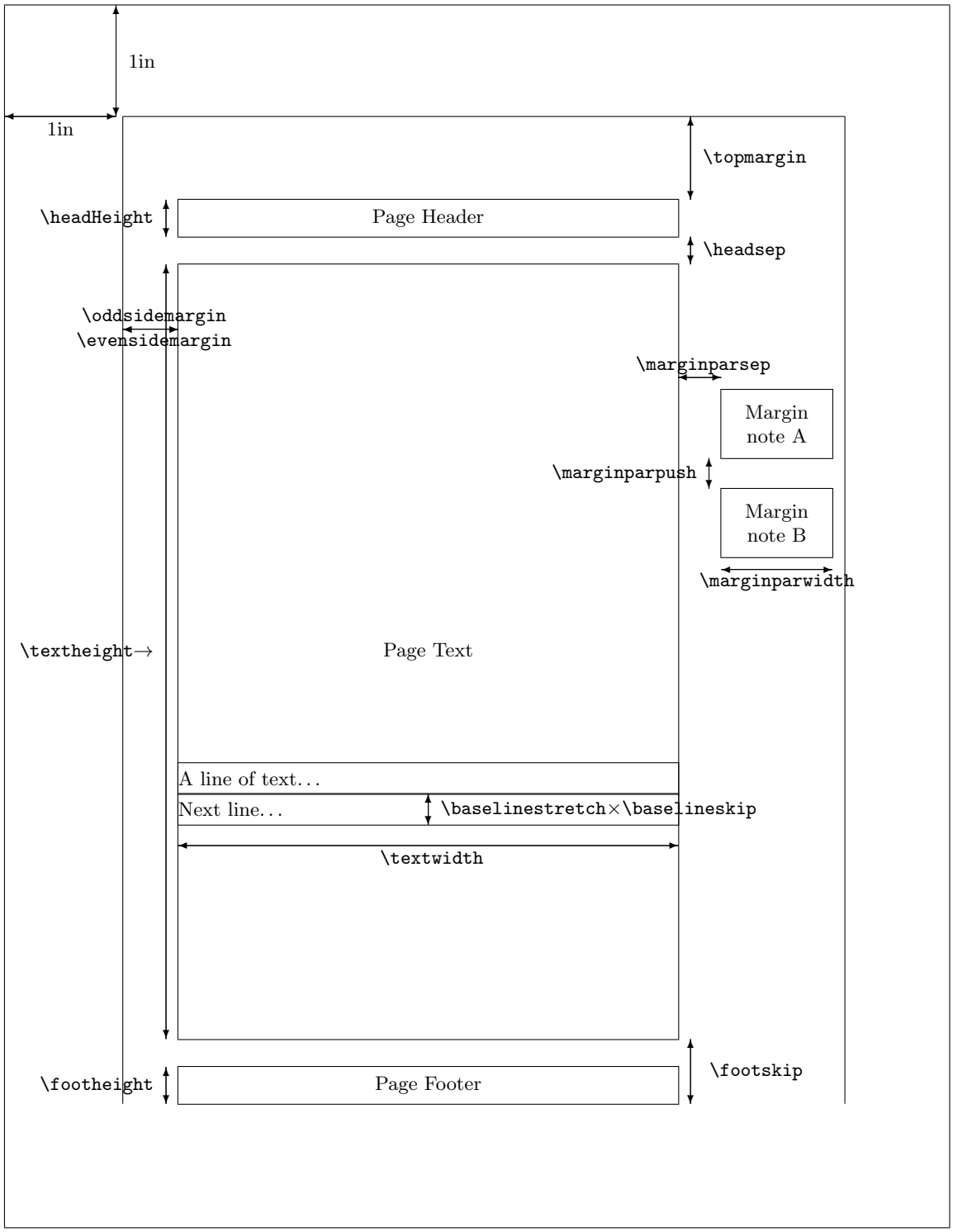
Bei der Option `twoside` besteht die Seitenbreite von linken Seiten mit linken Randnoten aus folgenden Elementen:

1 Inch
`\evensidemargin`
`\textwidth`
Rest der Seite

Anmerkungen: Die Größen `\topmargin`, `\oddsidemargin` und `\evensidemargin` können auch negative Werte haben, dann ergibt sich an der betreffenden Stelle ein Rand von weniger als 1 Inch. Auch `\leftskip` und `\rightskip` können negative Werte haben, dann ragt der Text entsprechend aus dem Textbereich heraus. Ausführlichere Hinweise und Skizzen zu diesem Thema findet man unter anderem in der Zeitschrift TUGboat Vol.9, No.1 (April 1988).

Die Größe `\footheight` ist in LaTeX2e *nicht* mehr definiert, da sie anscheinend von niemandem benutzt wurde.

Abbildung 1: Seiten-Layout



B Beschreibung der Dokumentklasse Refman

Am EDV-Zentrum der TU Wien wurde – aufbauend auf den in den vorangegangenen Kapiteln angeführten Grundsätzen – eine Style-Option `refman` zu den LaTeX-Styles `article` und `report` entwickelt, die ein Layout für Reference Manuals, technische Beschreibungen und dergleichen realisiert. Das Layout entspricht im wesentlichen den Angaben in Abschnitt 3.6: Die normalen Textseiten haben einen breiten linken Rand, der für Überschriften und Randnoten benützt wird, und nur einen eher knappen freien Rand oben, rechts und unten.

Für LaTeX2e wurde diese Option in eine echte Class umgewandelt. Dadurch ist es leichter möglich, verschiedene Papierformate zu unterstüzen. Außerdem gibt es neben der `report`-ähnliche `refrep`-Class eine `article`-ähnliche `refart`-Class, die sich durch andere Einträge in Kopf- und Fuß-Zeile sowie das Fehlen der Gliederungsebene `chapter` von `refrep` unterscheidet.

Die derzeitige Version dieser Class wird im folgenden kurz in Hinblick auf ihre Anwendeung beschrieben. Dies soll gleichzeitig als Beispiel dafür dienen, welche Änderungen und zusätzliche Komponenten ein typischer neuer LaTeX-Style enthalten kann.

Das vorliegende Handbuch wurde mit der Documentclass `refart` gesetzt. Es ist also gleichzeitig ein Beispiel für das von dieser Class bewirkte Layout.

B.1 Aufruf

Im LaTeX Local Guide ist jeweils angegeben, ob und wo dieses oder ein ähnliches Class-File in einer bestimmten LaTeX-Installation verfügbar ist. Der Aufruf erfolgt durch die Angabe der Class `refart` im Befehl `\documentclass`, also z. B. mit

```
\documentclass[11pt,a4paper]{refart}
\usepackage{german}
```

B.2 Optionen

Die Class `refart` ersetzt die Class `article` und die Class `refrep` die Class `report`. Sie kann *nicht* mit der folgenden Option verwendet werden: `twocolumn`.

Da es sich jetzt um eine selbstständige Class handelt, sind folgende Optionen möglich: `landscape`, `square` sowie die aus den Standard–Classes bekannten Papiergrößen-Optionen `a4paper`, `a5paper` oder auch die Optionen für amerikanische Formate. Die `square`-Option erzeugt ein quadratisches Layout, bei dem die Höhe auf die Papierbreite begrenzt wird.

Sie *kann* auch mit zahlreichen anderen Optionen kombiniert werden, auch z. B. mit `twoside` oder `fleqn`. Die Option `titelpage` erzeugt bei der `refart`-Class eine eigene Titelseite. Bei der Class `refrep` ist dies die Standardeinstellung.

Die Classes `refart` und `refrep` unterstützen nur die einspaltige Ausgabe, die Befehle `\twocolumn` und `\onecolumn` können in diesem Fall *nicht* verwendet werden.

Der Index wird automatisch zweispaltig gesetzt. Das Layout entspricht hier dem der Standard–Classes, allerdings mit kleinen Überschriften wie im Rest des Dokumentes.

B.3 Layout-Änderungen

B.3.1 Seitenaufteilung

Horizontal: Die komplette genützte Breite (`\fullwidth`) berechnet sich so, daß am Papier rechts und links noch jeweils 1 Inch freier Rand bleiben. Die Zeilenlänge des normalen Textes (`\textwidth`) und alle davon abgeleiteten Größen betragen 70% der `\fullwidth`. Die Breite des Randes (`\leftmarginwidth`), der links vom normalen Textbereich für Überschriften und Randnoten verwendet wird, ergibt sich aus der Differenz von `\fullwidth` und `\textwidth`.

Vertikal: Die Texthöhe berechnet sich so, daß ein oberer und unterer Rand von 1 Inch bleibt. Manche Pagestyles verwenden einen anderen Wert für den oberen Rand. (siehe B.4.3).

Mit der Option `smallborder` wird der Rand auf 0,25 Inch verringert.

Der Rand kann durch den Befehl `\setlength\papermarginwidth{0.25in}` umdefiniert werden, danach ist der Befehl `\setpagefraction{0.7}` aufzurufen, um das neue Layout zu berechnen. Dies funktioniert nur in der Präambel. Die Blattgröße wird dem DVI-Treiber mitgeteilt, wenn die Option `pagesize` angegeben wird.

Es wird kein vertikaler Randausgleich gemacht (`\raggedbottom`).

B.3.2 Kapitel-Überschriften

Die Überschriften von `\section`, `\subsection` und `\subsubsection` verwenden die gesamte Seitenbreite, d. h. sie ragen in den linken Rand hinaus. Dafür werden kleinere vertikale Abstände vor und nach den Überschriften und eine weniger große Schrift verwendet. In den Überschriften werden keine Wortabteilungen und kein rechter Randausgleich gemacht.

Beim Style `refart` werden die von `\maketitle` und `\part` erzeugten Überschriften im folgenden Layout gesetzt: Sie stehen am Beginn einer neuen Seite (`\clearpage`), verwenden die gesamte Breite und sind von zwei waagrechten Linien eingeschlossen. Der Befehl `\maketitle` setzt die Überschrift (`\title`) und den Autor (`\author`). Der Befehl `\makeauthor`, wird nicht mehr unterstützt. Beim Befehl `\part` wird nur die römische Nummer des Abschnitts vor den Text gesetzt (ohne das Wort „Part“ oder „Teil“).

Beim Style `refrep` werden Kapitel-Überschriften (`\chapter`) im folgenden Layout gesetzt: Sie stehen am Beginn einer neuen Seite (`\clearpage`), verwenden die gesamte Breite und sind von zwei waagrechten Linien eingeschlossen. Die Befehle `\maketitle` und `\part` bleiben in diesem Style unverändert, es wird aber empfohlen, sie in ein Fullpage-Environment (siehe B.5.5) einzuschließen.

B.3.3 Absätze

Absätze werden durch einen vertikalen Abstand (`\parskip`) von einer halben Zeile getrennt, ohne horizontale Einrückung (`\parindent` ist Null). Der vertikale Abstand enthält einen „stretchable“ Anteil von „`plus 2pt`“; dies vergrößert die Wahrscheinlichkeit, daß LaTeX die am besten geeigneten Stellen für alle Seitenwechsel findet.

Die vertikalen Abstände in den List-Environments sind so gesetzt, daß sich vor, nach und innerhalb der Environments jeweils derselbe Abstand

wie zwischen den normalen Absätzen ergibt.

B.4 Fußnoten

Das Layout der Fußnoten wurde an das Absatzlayout angepast, d. h. auch die Fußnoten werden ohne Einzug in der ersten Zeile gesetzt.

B.4.1 Description-Environment

Das `description`-Environment ist so abgeändert, daß die Item-Labels den gesamten linken Rand verwenden.

→ Kapitel 3

Beispiele dafür finden sich unter anderem in Kapitel 3.

B.4.2 Positionierung der Randnoten

Randnoten (`\marginpar`) werden stets in den linken Rand gesetzt und verwenden dessen gesamte Breite.

Der minimale vertikale Abstand zwischen Randnoten (`\marginparpush`) ist auf Null gesetzt, damit möglichst keine Verschiebung von Randnoten erfolgt.

B.4.3 Kopf- und Fußzeilen, Pagestyles

Beim Pagestyle `plain` werden die Seitennummern in der Fußzeile jeweils ganz rechts bzw. (auf linken Seiten bei der Option `twoside`) ganz links außen gesetzt.

Bei den Pagestyles `headings` und `myheadings` wird ebenfalls die gesamte Breite verwendet, und unter die Kopfzeile wird eine waagrechte Linie gesetzt. Beim Pagestyle `headings` enthält die Kopfzeile beim `refart` die `\section`-Überschrift auf der linke und die `\subsection`-Überschrift auf der linken Seite, beim `refrep` die `\chapter`-Überschrift auf der linken und die `\section`-Überschrift auf der rechten Seite. Die Schrift wird in schräger (slanted) Schrift gesetzt.

Seit Version 2.0 e wird für die erste Seite eines Kapitels der Pagestyle `empty` anstelle von `plain` verwendet. Bei Verwendung des Pagestyles `headings` kamm es zu seltsamen Effekten, wenn die erste Seite mit den Pagestyle `plain` gesätzt wurde.

Zusätzlich sind auch Pagestyles `footings` und `myfootings` definiert, die entsprechende Fußzeilen (mit einer waagrechten Linie darüber) setzen. Beispiel: Die folgenden Befehle setzen den Kurztitel des Schriftstücks in die Fußzeile:

```
\pagestyle{myfootings}
\markboth{Kurztitel}{Kurztitel}
```

Die Pagestyles `headings` und `myheadings` setzen den oberen Rand auf einen etwas größeren Wert, um die leere Fußzeile auszugleichen. Die Pagestyles `footings` und `myfootings` setzen den oberen Rand auf einen etwas kleineren Wert, um die leere Kopfzeile auszugleichen. Die Pagestyles `empty` und `plain` lassen den Rand unverändert. Daraus ergibt sich: Innerhalb eines Schriftstücks kann manentweder zwischen den Pagestyles `headings`, `myheadings` und `empty`
oder zwischen den Pagestyles `footings`, `myfootings` und `empty` wechseln, aber man sollte nicht zwischen `footings`- `headings`- und `plain` Seiten wechseln.

B.5 Zusätzliche Befehle

B.5.1 Marginlabel

Der Befehl `\marginlabel{xxx}` setzt das Wort oder den Text `xxx` rechtsbündig in den linken Rand. Dies kann für Stichworte, kurze Überschriften von Absätzen oder ähnliches verwendet werden.

Beispiel: Das Wort „Beispiel:“ links von diesem Absatz ist ein Beispiel für einen Marginlabel.

B.5.2 Attention

Der Befehl `\attention` setzt ein Ausrufungszeichen in den linken Rand. Dies dient zum Markieren wichtiger Absätze oder Sätze.

! → Das ist ein Beispiel für `\attention`.

`:-)` Seit Version 2.0c ist es möglich das Symbol für den `\attention` Befehl mit Hilfe von `\renewcommand{\attentionsymbol}{\texttt\{:-)\}}` zu verändern. Mit `\renewcommand{\attentionsymbol}{\large \bfseries ! $\rightarrow$}` wird wieder auf das Originalverhalten zurückgeschaltet.

Zusätzlich unterstützt der `\attention` Befehl ein optionales Argument um das Symbol im Rand einmalig durch ein anderes zu ersätzen. In diesem Fall ist es nicht notwendig, das `\attentionsymbol` zurückzusätzen. Diese Erweiterung dient vor allem zur Unterstützung für das `manfnt` Paket und

`:-(` sollte nicht zur Erzeugung eines bunten Layouts mißbraucht werden.

Mit dem `manfnt` Paket existiert eine einfache Möglichkeit, um auf die Symbole des `manfnt` Zeichensatzes zuzugreifen. Dieser Zeichensatz enthält u.a. das „Dangerous Bend“ und „Double Dangerous Bend“ Zeichen.

B.5.3 Seealso

Der Befehl `\seealso{n}` markiert einen Verweis auf ein Kapitel oder eine Abbildung.

Beispiele dafür finden sich nicht nur hier (am linken Rand) sondern unter anderem auch in Kapitel 1.

→ Kapitel 1

B.5.4 Maxipage-Environment

Das `maxipage`-Environment ist eine Art von „Minipage“, die sich über die gesamte Seitenbreite erstreckt. Es eignet sich z. B. für überlange mathematische Gleichungen oder breite Tabellen innerhalb des Textes oder in „Floats“ (`table` oder `figure`). Innerhalb der „Maxipage“ sind keine Seitenwechsel und keine Randnoten möglich. Beginn und Ende des `maxipage`-Envrionment bewirken jeweils einen neuen Absatz und eine waagrechte Linie.

Hier folgt nun ein Beispiel für eine Maxipage.

Diese überlangen Zeilen sind der Inhalt der Maxipage. Sie erstrecken sich über die gesamte Seitenbreite, ohne linken Rand.

Dies ist der normale Text nach dem Ende der Maxipage.

B.5.5 Fullpage-Environment

Das `fullpage`-Environment besteht aus einer oder mehreren Seiten, bei denen sich der Text über die gesamte Seitenbreite erstreckt und kein linker Rand bleibt. Randnoten sind darin nicht möglich. Beginn und Ende des `fullpage`-Envrionment bewirken jeweils einen Seitenwechsel (`\clearpage`).

Dies eignet sich für Spezialseiten wie mehrseitige große Tabellen, Programmlistings, Titelseiten, Inhaltsverzeichnisse und ähnliches.

→ Seite 15

Seite 15 ist ein Beispiel für eine Fullpage.

B.5.6 Noparskip

Der Befehl `\noparskip` entfernt den vertikalen Abstand vor einem Absatz – analog zum Befehl `\noindent`, der die horizontale Einrückung am Beginn eines Absatzes entfernt.

B.5.7 Setleftmarginwidth

Der Befehl `\setleftmarginwidth` wird nicht mehr unterstützt. Stattdessen kann mit dem Befehl `\settextfraction` der Anteil des Textes (normalerweise 70 %) geändert werden. Das Argument muß zwischen 0 und 1 liegen.

B.5.8 Descriptioncolon

Der Doppelpunkt in einer `description` kann durch den Befehl `\descriptioncolonfalse` ab und durch `\descriptioncolontrue` wieder eingeschaltet werden. Im Normalfall ist der Doppelpunkt eingeschaltet.

B.5.9 Descriptionleft

Mit `\descriptionlefttrue` wird der Label für eine `description` linksbündig in den Rand gesetzt. Der Normalfall ist rechtsbündig, dies wird durch `\descriptionleftfalse` erreicht.

B.5.10 Maxipagerule

Mit `\maxipagerulefalse` lassen sich die Striche vor und nach einer `maxipage` abschalten. Sie sind normalerweise aktiv.

B.5.11 Condbreak

Mit `\condbreak{2cm}` wird sichergestellt, daß die nächsten 2 cm nicht getrennt werden. Sie erscheinen entweder auf der aktuellen Seite oder werden komplett auf die neue Seite umgebrochen.

B.5.12 Example

Die `example`-Umgebung erzeugt eine `verse`-Umgebung, schaltet allerdings auf `tt` und versucht einen Seitenumbruch vor der Umgebung zu vermeiden.

B.5.13 Pageperchapter

Der Befehl `\pageperchapter` erzeugt eine kapitelweise Seitennumerierung. Dies ist besonders für umfangreiches Reference Manuals sinnvoll. Da nur `refrep` über die Gliederungsebene `chapter` verfügt, ist dieser Befehl auch nur hier möglich.

B.5.14 Smallborder

Der normale Seitenrand beträgt 1 inch (25,4 mm). Das ist für ein gedrucktes Dokument in Ordnung, für die Betrachtung auf dem Monitor aber verschwendete Bildschirmfläche. Die Option `smallborder` reduziert den Rand auf 0,25 inch (6,35 mm). Damit klebt der Text noch nicht am Rand, es wird aber auch nicht zu viel Bildschirmfläche verschenkt.

Der Rand kann durch den Befehl `\setlength\papermarginwidth{0.25in}` umdefiniert werden, danach ist der Befehl `\setpagefraction{0.7}` aufzurufen, um das neue Layout zu berechnen.

B.5.15 Dvips

Die Option `dvips` teilt DVIPS die aktuelle Blattgröße mit.

B.5.16 Pdftex

Die Option `pdftex` teilt PDFTEX die aktuelle Blattgröße mit.

B.5.17 Pagesize

Pagesize wählt automatisch den passenden Befehl für den jeweiligen DVI-Treiber. Dies funktioniert bei DVIPS und DVIPDFMX für DVI-Ausgabe und PDFTEXfür PDF-Ausgabe.

B.5.18 Ifpdfoutput

Mit dem Befehl `\ifpdfoutput{pdftext}{dvitext}` können Texte abhängig vom Ausgabeformat gesätzt werden. Die Option `pagesize` nutzt diesen Befehl.

Die letzten vier Befehle wurden aus KOMA-Script übernommen, vielen Dank Markus!

Index

a4paper, 16
Attention, 19
Attentionsymbol, 19
Autor, 3

Condbreak, 20

Description-Environment, 18
Descriptioncolon, 20
Descriptionleft, 20
Design, generisches, 6
Design, logisches, 6
Design, visuelles, 6
Designer, 3
Dvips, 21

Example, 20

Faustregeln, 4
Footings, 18
Fullpage, 20
Fußzeilen, 5, 13

Headings, 18
House-Style, 4

Ifpdfoutput, 21

Kopfzeilen, 5, 13

Landscape, 16
Layout-Änderungen, Vorgangsweise, 7
Layout-Design, 4

manfnt, 19
Manual, 4
Marginlabel, 19
Maxipage, 19
Maxipagerule, 20
Myfootings, 18
Myheadings, 18

Optionen, 16

Pageperchapter, 21
Pagesize, 21
Pagestyles, 18
Pdftex, 21

Randnoten, 5, 13
Refart,Aufruf, 16
Refart.cls, 16
Refrep,Aufruf, 16
Refrep.cls, 16
Regeln, 4

Seealso, 19
Seitenaufbau, 4
Seitenaufteilung, horizontal, 17
Seitenaufteilung, vertikal, 17
Setleftmarginwidth, 20
Setzer, 3
Smallborder, 21
Square, 16

Telefonbuch, 4

Zeilenabstände, 4
Zeilenlänge, 4