

Das pauldoc-Package – Anpassungen für doc für Pauls Package-Dokus*

Paul Ebermann[†]

6. November 2009

Zusammenfassung

Dieses Package enthält einige Befehle, welche die Funktion von `doc` an meine Wünsche anpassen.

Außerdem werden noch einige gewünschte Pakete geladen sowie einige neue Befehle definiert.

Inhaltsverzeichnis

1 Benutzerdoku	2
1.1 Package-Optionen	2
1.2 Neue Makros	2
2 Implementation	3
2.1 Package-Optionen	3
2.2 geladene Pakete	3
2.3 Diverse Einstellungen	4
2.4 Neue oder geänderte Makros	4
2.4.1 Bedingte Anweisungen	4
2.4.2 Packagenamen	5
2.4.3 Logos	5
2.4.4 Lizenz	5
2.4.5 Kopiert aus ltxdoc	7
2.4.6 Bugfixes	8
2.5 Ende	8
A Liste der Änderungen	8
B Index	9

*Dieses Dokument gehört zu `pauldoc v0.5`, vom 2009/11/06.

[†]Paul-Ebermann@gmx.de

1 Benutzerdoku

1.1 Package-Optionen

- chapter** Die Option `chapter` legt fest, dass Index und Liste der Änderungen in einem eigenen Kapitel (anstatt in einem Abschnitt) beginnen. Dies ist nur in einer Dokumentenklasse sinnvoll, welche überhaupt `\chapter` definiert, etwa den `book`-artigen Klassen.
- section** Die Option `section` legt fest, dass Index und Änderungsliste in einem Abschnitt beginnen. Dies ist der Vorgabewert.
- latin1** Die Option `latin1` legt fest, dass das Dokument im ISO-8859-1-Format kodiert ist. Dies ist der Vorgabewert. (Die Option wird an das `inputenc`-Paket weitergegeben.)
- utf8** Die Option `utf8` legt fest, dass das Dokument als UTF-8 kodiert ist. (Die Option wird an das `inputenc`-Paket weitergegeben.)

1.2 Neue Makros

`\ifReferenceExists` $\langle refName \rangle \{ \langle then \rangle \} \{ \langle else \rangle \}$

Mit diesem Makro kann man, abhängig davon, ob eine bestimmte Referenz mit `\label{}` gesetzt wurde, bestimmten Text ausgeben.

Falls ein `\ref{⟨refName⟩}` Erfolg hätte, wird $\langle then \rangle$ ausgewertet, ansonsten $\langle else \rangle$. Dies ist nützlich, wenn man im Beschreibungsteil verschiedenen Text aufnehmen will, abhängig davon, ob auch der Implementationsteil mitgesetzt wird – man kann dann etwa auf einen bestimmten Abschnitt darin verweisen, und andernfalls den Text im Konjunktiv formulieren („Wenn der Implementationsteil mit enthalten wäre, könnte man dort jetzt . . . finden.“).

Gegebenenfalls wird erst beim zweiten L^AT_EX-Durchlauf der richtige Zweig ausgewertet.

`\pack` $\{ \langle name \rangle \}$

In Paketbeschreibungen werden häufiger Paketnamen (der des eigenen Paketes, gelegentlich auch andere Pakete) verwendet. Wie in „How to Package Your L^AT_EX Package“ (Scott Pakin, dtxtut.pdf), Abschnitt 3.2, erläutert, setzt man Paketnamen (sofern es dafür nicht ein eigenes Logo gibt) üblicherweise in `\textsf{}` (die serifenlose Schrift der Klasse). Ich bevorzuge einen sprechenderen Namen, daher gibt es jetzt `\pack{pauldoc}` für `pauldoc`.

`\includeLicense` `\includeLicense` bindet die eventuell im T_EX-Suchpfad zu findende Lizenz (LPPL) ein. Dies ergibt – je nach dem, ob `section` oder `chapter` als Option gegeben wurde – ein neues Kapitel oder einen neuen Abschnitt. Durch Neudefinition von `\lplfilename` kann festgelegt werden, welche Datei zu verwenden ist – als Vorgabewert ist die bei L^AT_EX mitgelieferte Datei `lpl.tex` im Package festgelegt.

`\cmd` Diese fünf Makros habe ich aus `ltxdoc` kopiert, damit man sie auch verwenden
`\cs` kann, wenn eine Doku mit einer anderen Klasse formatiert wird (wie bei meiner `algs`
`\marg` `script`-Klasse). (Sie sind so formatiert, dass eine andere Definition immer vorgeht.)
`\oarg`
`\parg`

`\cmd{makro}` formatiert einen Makronamen. `\cmd{bla}` ergibt `\bla`. (Sollte der erste Buchstabe kein `\` sein, wird er damit ersetzt: `\cmd{abc}` ergibt `\bc`.) `\cs{makro}` tut das gleiche – hier wird allerdings der `\` noch angefügt. Falls `\cmd` mal nicht geht, geht gelegentlich noch `\cs`. `\cs{bla}` ergibt `\bla`.

`\marg{text}` gibt `{text}` aus. Die Abkürzung steht für „mandatory argument“. `\oarg{text}` ergibt `[text]` („optional argument“). `\parg{text}` ergibt `((text))` („picture mode argument“).

2 Implementation

```
1 %<*package>
```

2.1 Package-Optionen

`utf8` Mi der Option `utf8` bzw. `latin1` wähle ich die Kodierung der Dokumentation aus.
`latin1` `latin1` ist der Default. Die Optionen selbst merken sich ihren Wert einfach in der
`\pauldoc@ienc` Variable `\pauldoc@ienc`.

```
2 \newcommand*\pauldoc@ienc{}
3 \DeclareOption{utf8}{
4   \def\pauldoc@ienc{utf8}
5 }
6 \DeclareOption{latin1}{
7   \def\pauldoc@ienc{latin1}
8 }
9 \ExecuteOptions{latin1}
```

`chapter` `\chapter` für die Index-Überschrift.

```
10 \DeclareOption{chapter}{
11   \let\pauldoc@indexsec=\chapter
12 }
```

`section` `\section` für die Index-Überschrift.

```
13 \DeclareOption{section}{
14   \let\pauldoc@indexsec=\section
15 }
```

`\pauldoc@indexsec` Wir definieren noch `\pauldoc@indexsec`, rufen dann die Vorgabeoption `section` auf (welche das gleich neu definiert) und verarbeiten die gegebenen Optionen.

```
16 \newcommand*\pauldoc@indexsec{}
17 \ExecuteOptions{section}
18 \ProcessOptions\relax
```

2.2 geladene Pakete

Wir laden das Paket `inputenc` mit Option `latin1` oder `utf8` (je nach Paketoption), um Sonderzeichen auch direkt eingeben zu können.

`babel` sorgt für vernünftige deutsche Bezeichnungen.

```

19 \RequirePackage[\pauldoc@ienc]{inputenc}
20 \RequirePackage[ngerman]{babel}

```

fontenc mit Option T1 lädt das T1-Fontencoding, in dem etwa Umlaute direkt aus der Schriftart genommen werden können, anstatt sie mit " und den Vokalen a, o und u zusammensetzen. Dies führt auch zu einer verbesserten Silbentrennung.

```

21 \RequirePackage[T1]{fontenc}

```

2.3 Diverse Einstellungen

Wir wollen natürlich in der Doku Querverweise haben, die Änderungen registrieren (für eine entsprechende Liste) und im Index mit Zeilennummern arbeiten. Quelltext in der Doku wollen wir mit ' markieren.

```

22 \EnableCrossrefs
23 \RecordChanges
24 \CodelineIndex
25 \AtBeginDocument{\MakeShortVerb{\'}\selectlanguage{ngerman}}

```

Außerdem sollten Index und Änderungsliste auf deutsch beschriftet werden. Wie benutzen hier für die Überschrift \@pauldoc@indexsec, was entweder \section oder \chapter ist.

```

26 \renewcommand{\generalname}{Allgemein}
27
28 \GlossaryPrologue{\pauldoc@indexsec{Liste der \"Anderungen}}
29
30 \IndexPrologue{
31
32   \pauldoc@indexsec{Index}
33
34   Schr\"aggedruckte Nummern verweisen auf die Seite, auf der der
35   Eintrag beschrieben ist, unterstrichene Nummern zeigen auf die
36   Zeilennummer der Definition, sonstige Zahlen auf die Zeilennummer
37   einer Verwendung.
38
39 }

```

Der Index bekommt nur zwei statt sonst drei Spalten (wir haben einige ziemlich lange Makronamen).

```

40 \setcounter{IndexColumns}{2}

```

2.4 Neue oder geänderte Makros

2.4.1 Bedingte Anweisungen

```

\ifReferenceExists {<label>}{<then>}{<else>}

```

\ifReferenceExists prüft einfach, ob das Makro \r@<label> (dort wird die Referenz auf das Label gespeichert), existiert, und wertet entsprechend <then> oder <else> aus.

```

41 \newcommand*{\ifReferenceExists}[3]

```

```

42 {%
43 \@ifundefined{r@#1}{#3}{#2}%
44 }

```

2.4.2 Packagenamen

`\pack` Um Packagenamen wie `pauldoc` vernünftig auszeichnen zu können, dieser Befehl.

```

45 \newcommand*\pack{\textsf}

```

2.4.3 Logos

`\eTeX` Aus `etex.sty` geklaut – das ε -TeX-Logo.

```

46 \newcommand*\eTeX{$\m@th\varepsilon$-\TeX}

```

2.4.4 Lizenz

`\includeLicense` Die Lizenz wird nur eingelesen, wenn sie auch gefunden wird. Dafür wird der Dateiname, welcher in `\lplfilename` steht, eingebunden.

```

47 \newcommand*\includeLicense {
48 \IfFileExists{\lplfilename}{%

```

Zunächst stellen wir die Sprache um (aber nur Silbentrennung etc., keine Namen), mit dem passenden `babel`-Befehl.

```

49 \begin{otherlanguage*}{english}

```

Um die Lizenz als `\chapter` einzubinden und dann Referenzen darauf im Text verwenden zu können, waren mit der L^AT_EX-Version vom 13. Januar 2006 einige Verrenkungen notwendig – siehe unten.

Am 10. Februar habe ich einen Vorschlag zur Verbesserung an die LaTeX-Bug-Liste geschickt, das wurde positiv aufgenommen.¹ Ich habe dann von einer verbesserten Version von Frank Mittelbach bekommen, die inzwischen auch veröffentlicht wurde.

Wir wollen die Abschnitte der Lizenz auch nummeriert (und im Inhaltsverzeichnis) haben, deswegen die Versionen ohne `*`. Falls `chapter` als Package-Option angegeben wurde, schieben wir außerdem alles eine Ebene höher.

```

50 \ifx\pauldoc@indexsec\chapter
51 \providecommand*\LPPLsection{\chapter}
52 \providecommand*\LPPLsubsection{\section}
53 \providecommand*\LPPLsubsubsection{\subsection}
54 \providecommand*\LPPLparagraph{\subsubsection}
55 \else
56 \providecommand*\LPPLsection{\section}
57 \providecommand*\LPPLsubsection{\subsection}
58 \providecommand*\LPPLsubsubsection{\subsubsection}
59 \providecommand*\LPPLparagraph{\paragraph}
60 \fi

```

¹Siehe <http://www.latex-project.org/cgi-bin/ltxbugs2html?pr=latex/3839>.

Der folgende (auskommentierte) Code sowie die eingerückte Dokumentation dazu ist mit der neuen Version der Lizenz-Datei nicht mehr notwendig – ich lasse ihn mal trotzdem hier drin, zu Dokumentationszwecken.

Die Lizenz-Datei (zumindest die mir vorliegende Version) verwendet Unterteilungen ab `\section` – falls in unserer Dokumentation `\chapter` verwendet werden soll (d.h. die entsprechende Option gegeben wurde), stellen wir einen Kapitelbeginn davor.

```
\ifx\pauldoc@indexsec\chapter
  \chapter{Lizenz}%
\fi
```

`\section*` *Damit wir aus dem Dokument auf die Lizenz referieren können, definieren wir den Befehl `\section`, welcher zu Beginn der Lizenz verwendet wird, so um, dass er ein Label setzt. (Weil wir uns in einer Gruppe befinden (durch die `otherlanguage`-Umgebung), bleibt die Änderung lokal.)*

```
\newcommand{\lizenz@oldsection@}{}%
\let\lizenz@oldsection@\section\relax%
\def\section*{%
  \label{lplp1-chapter}%
  \lizenz@oldsection@*{##1}%
}%
```

`\emph` *Leider habe ich es nicht wirklich geschafft, den Befehl so umzudefinieren, dass nach dem `\section*{}`-Befehl noch ein Label kommt – deswegen definiere ich stattdessen den `\emph{}`-Befehl um, der zumindest in meiner Ausgabe der LPPL direkt nach dem `\section*` kommt. Ein böser Hack, ich weiß ...*

```
\newcommand{\lizenz@oldemph@}{}%
\let\lizenz@oldemph@\emph%
\def\emph{%
  \label{lplp1-section}%
  \let\emph\lizenz@oldemph%
  \emph%
}%
```

Einige Änderungen sind notwendig, um die Lizenz einzulesen: `%` soll wieder ein Kommentarzeichen sein, `'` darf kein Verbatim-Zeichen mehr sein, weil diese Zeichen in der Lizenz-Datei natürlich nicht den Doc-Konventionen entsprechend verwendet werden. Dann lesen wir die Datei ein, und machen danach die Änderungen wieder rückgängig.

```
61      \DeleteShortVerb{\'}%
62      \MakePercentComment\input{\lplp1filename}\MakePercentIgnore%
63      \MakeShortVerb{\'}%
```

... und jetzt setzen wir die Sprache wieder zurück.

```
64      \end{otherlanguage*}
```

Falls die Lizenz-Datei nicht gefunden wurde, geben wir nur eine passende Meldung aus.

```

65   }{%
66     \typeout{^^J%
67       ^^J%
68       Die Datei \lplfilename{} wurde nicht gefunden.^^J%
69       Schade, da wird die Lizenz eben nicht eingebunden.^^J%
70       ^^J%
71     }%
72   }%
73 }
```

`\lplfilename` Ich definiere noch den Dateinamen der Lizenz, unter der die meisten meiner Pakete stehen, zur Zeit ist das die unter dem Namen `lpl.tex` verbreitete LPPL.

```

74 \AtBeginDocument{%
75   \providecommand{\lplfilename}{lpl.tex}%
76 }
```

2.4.5 Kopiert aus ltxdoc

Die folgenden vier Kommandos kopierte ich aus `ltxdoc`, um sie auch verwenden zu können, wenn ich nicht diese Klasse verwende. Durch die Verwendung von `\providecommand` gibt es keine Konflikte, falls sie doch schon definiert sind.

`\cmd` Formatiert einen Makronamen, `\cmd{\bla}` ergibt `\bla`. (Sollte der erste Buchstabe kein `\` sein, wird er damit ersetzt.)

```

77 \ifundefined{cs}
78 {%
79   \providecommand*{\cmd}[1]{\cs{\expandafter\cmd@to@cs\string#1}}
80   \def\cmd@to@cs#1#2{\char\number{#2}\relax}
```

`\cs` Implementiert wurde `\cmd` mit `\cs - \cs{\bla}` ergibt `\bla`. Dies funktioniert auch an einigen Stellen, wo `\cmd{\bla}` nicht funktioniert.²

```

81 \DeclareRobustCommand*\cs[1]{\texttt{\char'\#1}}%
82 }
83 }
```

`\marg` `\marg{text}` gibt `{<text>}` aus. Die Abkürzung steht für „mandatory argument“.

```

84 \providecommand\marg[1]{%
85   {\ttfamily\char'\{} \meta{#1} {\ttfamily\char'\}}
```

`\oarg` `\oarg{text}` ergibt `[<text>]` („optional argument“).

```

86 \providecommand\oarg[1]{%
87   {\ttfamily[] \meta{#1} {\ttfamily}}}
```

`\parg` `\parg{text}` ergibt `(<text>)` („picture mode argument“).

```

88 \providecommand\parg[1]{%
89   {\ttfamily{ } \meta{#1} {\ttfamily}}}
```

²Das steht jedenfalls in der `ltxdoc`-Doku, auch wenn ich das nicht ganz verstanden habe.

2.4.6 Bugfixes

Inzwischen³ ist der folgende Bug korrigiert, daher hier auskommentiert.

`\SpecialEnvIndex` Aufgrund eines Bugs in *doc*⁴ definiere ich hier `\SpecialEnvIndex` neu, damit die *environment*-Verwendungen genauso wie die *environment*-Definitionen einsortiert werden, nicht als eigener Eintrag namens „*environments*:*<name>*“:

```
\renewcommand*\SpecialEnvIndex[1]{\@bsphack
  \index{#1\actualchar{\protect\ttfamily#1}
    (environment)\encapchar usage}%
  \index{environments:\levelchar#1\actualchar{%
    \protect\ttfamily#1}\encapchar
    usage}\@esphack}
```

2.5 Ende

... und Schluss.

```
90 \endinput
91 \</package>
```

A Liste der Änderungen

v0.0		[T1]{fontenc}	3	
	Allgemein: Erste Fassung	1	v0.4	
v0.1		\eTeX: Neu.	5	
	Allgemein: <code>\cs</code> , <code>\cmd</code> , <code>\marg</code> , <code>\oarg</code> , <code>\parg</code> hinzugefügt aus <i>ltxdoc</i> . . .	7	v0.4a	
	Erste veröffentlichte Fassung . . .	1	Allgemein: Kleinere Änderungen an der Dokumentation.	1
	<code>\SpecialEnvIndex</code> : Bugfix ist über- flüssig	8	v0.5	
v0.2		Allgemein: $\ddot{A} \Rightarrow \backslash"A$ in „Liste der Änderungen“, und entsprechend für „Schräggedruckte“	4	
	<code>\includeLicense</code> : Verwendung ei- ner neuen Version der LPPL macht Workarounds unnötig. . .	5	README jetzt in UTF-8	1
	<code>\lplfilename</code> : Dateiname von <i>lpl-1-3b.tex</i> nach <i>lpl.tex</i> geändert.	7	latin1: Parameter für die Kodie- rung, wird an <code>inputenc</code> weiter- gegeben.	3
v0.3		utf8: Parameter für die Kodie- rung, wird an <code>inputenc</code> weiter- gegeben.	3	
	Allgemein: Jetzt mit <code>\usepackage</code>			

³mit dem L^AT_EX-Beta-Release vom 3. Februar 2006 – *doc* hat das Datum 2006/02/02

⁴Siehe <http://www.latex-project.org/cgi-bin/ltxbugs2html?pr=tools/3834> – das ist (laut einer E-Mail, die ich als Antwort auf meinen Bug-Report (am 2006/01/30) bekam) schon bekannt und im Code korrigiert, ein Release steht bevor.

B Index

Schräggedruckte Nummern verweisen auf die Seite, auf der der Eintrag beschrieben ist, unterstrichene Nummern zeigen auf die Zeilennummer der Definition, sonstige Zahlen auf die Zeilennummer einer Verwendung.

	Symbols	<code>\GlossaryPrologue</code> 28
<code>\"</code>	28, 34	
<code>\'</code>	25, 61, 63	I
<code>\@ifundefined</code>	43, 77	<code>\IfFileExists</code> 48
<code>\\</code>	81	<code>\ifReferenceExists</code> <u>2, 41</u>
<code>\{</code>	85	<code>\ifx</code> 50
<code>\}</code>	85	<code>\includeLicense</code> <u>2, 47</u>
	A	<code>\IndexPrologue</code> 30
<code>\AtBeginDocument</code>	25, 74	<code>\input</code> 62
	B	L
<code>\begin</code>	49	<code>latin1 (option)</code> <u>2, 2</u>
	C	<code>\let</code> 11, 14
<code>\chapter</code>	11, 50, 51	<code>\lizenz@oldsection@</code> <u>61</u>
<code>chapter (option)</code>	<u>1, 10</u>	<code>\lpplfilename</code> <u>2, 48, 62, 68, 74</u>
<code>\char</code>	80, 81, 85	<code>\LPPLparagraph</code> 54, 59
<code>\cmd</code>	<u>2, 77</u>	<code>\LPPLsection</code> 51, 56
<code>\cmd@to@cs</code>	79, 80	<code>\LPPLsubsection</code> 52, 57
<code>\CodelineIndex</code>	24	<code>\LPPLsubsubsection</code> 53, 58
<code>\cs</code>	<u>2, 79, 81</u>	M
	D	<code>\m@th</code> 46
<code>\DeclareOption</code>	3, 6, 10, 13	<code>\MakePercentComment</code> 62
<code>\DeclareRobustCommand</code>	81	<code>\MakePercentIgnore</code> 62
<code>\def</code>	4, 7, 80	<code>\MakeShortVerb</code> 25, 63
<code>\DeleteShortVerb</code>	61	<code>\marg</code> <u>2, 84</u>
	E	<code>\meta</code> 85, 87, 89
<code>\else</code>	55	N
<code>\emph</code>	<u>61</u>	<code>\newcommand</code> <u>2, 16, 41, 45–47</u>
<code>\EnableCrossrefs</code>	22	<code>\number</code> 80
<code>\end</code>	64	O
<code>\endinput</code>	90	<code>\oarg</code> <u>2, 86</u>
<code>\eTeX</code>	<u>46</u>	Optionen:
<code>\ExecuteOptions</code>	9, 17	<code>chapter</code> <u>1, 10</u>
<code>\expandafter</code>	79	<code>latin1</code> <u>2, 2</u>
	F	<code>section</code> <u>2, 13</u>
<code>\fi</code>	60	<code>utf8</code> <u>2, 2</u>
	G	P
<code>\generalname</code>	26	<code>\pack</code> <u>2, 45</u>
		<code>\paragraph</code> 59
		<code>\parg</code> <u>2, 88</u>

<code>\pauldoc@ienc</code>	2, 19	<code>\SpecialEnvIndex</code>	90
<code>\pauldoc@indexsec</code>	11, 14, <u>16</u> , 28, 32, 50	<code>\string</code>	79
<code>\ProcessOptions</code>	18	<code>\subsection</code>	53, 57
<code>\providecommand</code>		<code>\subsubsection</code>	54, 58
..	51–54, 56–59, 75, 79, 84, 86, 88		
R			
<code>\RecordChanges</code>	23	<code>\TeX</code>	46
<code>\relax</code>	18, 80	<code>\textsf</code>	45
<code>\renewcommand</code>	26	<code>\texttt</code>	81
<code>\RequirePackage</code>	19–21	<code>\ttfamily</code>	85, 87, 89
		<code>\typeout</code>	66
S			
<code>\section</code>	14, 52, 56	U	
<code>section</code> (option)	2, <u>13</u>	<code>utf8</code> (option)	<u>2</u> , 2
<code>\section*</code>	<u>61</u>	V	
<code>\selectlanguage</code>	25	<code>\varepsilon</code>	46
<code>\setcounter</code>	40		