

# Das Pfarrei-Paket\*

Markus Kohm<sup>†</sup>

2013/10/16<sup>‡</sup>

## Zusammenfassung

In „Die T<sub>E</sub>Xnische Kömddie“, Ausgabe 1/2013 hat Christian Justen über seinen Einsatz von L<sup>A</sup>T<sub>E</sub>X im Pfarrdienst berichtet. Einige der von ihm verwendeten bash-Scripte und seine Schilderungen dazu haben mich inspiriert, eine Sammlung zu beginnen, die entsprechende Dinge leisten. Allerdings habe ich mich entschlossen, keine bash-Scripte zu verwenden, sondern die Funktionalität derselben in lua-Scripte für die Ausführung mit T<sub>E</sub>XLua zu realisieren, da T<sub>E</sub>XLua inzwischen Bestandteil aller T<sub>E</sub>X-Distributionen ist. Somit wurde das ganze besser vom Betriebssystem unabhängig.

Ergänzt habe ich das ganze durch einige Befehle und Umgebungen, die für die Erstellung von Lied- und Gebetsheftchen für den Gottesdienst praktisch sein könnten. Jedenfalls habe ich selbst Nutzen daraus gezogen. Weiteres ist geplant.

## Inhaltsverzeichnis

<b>1 Die Skripte-Seite</b>	<b>2</b>
<b>2 Die L<sup>A</sup>T<sub>E</sub>X-Seite</b>	<b>4</b>
<b>3 Implementierung der L<sup>A</sup>T<sub>E</sub>X-Seite</b>	<b>6</b>
3.1 Das L <sup>A</sup> T <sub>E</sub> X-Dokument „a5toa4.tex“	6
3.2 Das L <sup>A</sup> T <sub>E</sub> X-Paket „pfarrei“	6
<b>4 Implementierung der Skripten</b>	<b>12</b>
4.1 Der kleine Wrapper „a5toa4.tlu“	12
4.2 Das Haupt-Skript „pfarrei.tlu“	12

---

\*Sorry if you don't understand German, but currently there's only a German manual, because the idea of this is from a German article at “Die T<sub>E</sub>Xnische Komödie”. But at least you may try to call “a5toa4 -help” to get some useful information.

<sup>†</sup>komascript at gmx info

<sup>‡</sup>Version r36

# 1 Die Skripte-Seite

In besagtem Artikel wurden zwei Skripte, „a5toa4“ und „a5toa4bogen“ vorgestellt. Beide erzeugen aus PDF-Dateien, die im A5-Format vorliegen, neue PDF-Dateien im A4-quer-Format.

Das erste, „a5toa4“, arrangiert die A5-Seiten dabei so, dass jeweils zwei aufeinander folgende A5-Seiten nebeneinander ausgegeben werden. Das ist vor allem dann praktisch, wenn man nur einseitig druckt und die Seiten später geteilt werden sollen. Im folgenden wird diese Ausgabe *side-by-side* genannt.

Das zweite, „a5toa4bogen“, erzeugt hingegen ein sogenanntes Booklet. Dabei werden die Seiten so angeordnet, dass nach dem beidseitigen Druck, der ganze Stapel nur noch in der Mitte geknickt werden muss, um eine Art Heft zu erhalten. Daher wird für diese Ausgabe im Weiteren die Bezeichnung *booklet* verwendet.

- a5toa4.tlu** Ich dachte mir, das könne man in ein einziges Programm gießen. Dieses heißt bei mir dann schlicht „a5toa4.tlu“. Bei optimaler Installation kann es auch als „a5toa4“ angesprochen werden.
- version** Ruft man das Programm mit der Option „-V“ oder „--version“ auf, so gibt es lediglich eine Versionsinformation aus.
- help** Beim Aufruf mit Option „-h“ oder „--help“ wird hingegen eine ausführliche Hilfe zu den Aufrufmöglichkeiten und den Optionen ausgegeben.
- booklet** Man kann es aber auch mit einer Reihe von Durchführungsoptionen und Namen von PDF-Dateien aufrufen. In diesem Fall wird entweder eine booklet- oder eine side-by-side-Ausgabe erzeugt. Die Wahl, ob eine booklet- oder eine side-by-side-Ausgabe erfolgen soll, erfolgt über die Optionen „-b“ oder „--booklet“ für booklet bzw. „-s“ oder „--sidebyside“ für side-by-side. Voreingestellt ist die side-by-side-Ausgabe.
- Für die Durchführung wurde auch ein Verbesserungsvorschlag von Christian Justen selbst aufgegriffen. Bei ihm haben die Skripte mit einer Zwischendateien mit dem festen Namen „cj.tmp“ gearbeitet. Im Extramfall konnte dies dazu führen, dass Dateien unerwünscht überschrieben und so vernichtet wurden.
- In meiner Fassung als texlua-Skript wird mit einem temporären Verzeichnis im aktuellen Arbeitsverzeichnis gearbeitet. Darin wird eine temporäre L<sup>A</sup>T<sub>E</sub>X-Datei erzeugt und alle Ausgabedateien eines PDF<sup>L</sup>A<sub>T</sub>E<sub>X</sub>-Laufs abgelegt. Das PDF-Ergebnis des PDF<sup>L</sup>A<sub>T</sub>E<sub>X</sub>-Laufs wird dann wieder in das noch immer aktuelle Arbeitsverzeichnis kopiert. Im Normalfall wird dabei der Basisname (also ohne Extension) der Ursprungsdatei je nachdem, was erzeugt wurde, entweder um „-booklet.pdf“ oder „-sidebyside.pdf“ ergänzt. Auch dies ist eine Abweichung von Christian Justens Original-Skripten, bei denen immer die Ursprungsdatei vom Ergebnis überschrieben wurde.
- overwrite** Ein Überschreiben der Ursprungsdatei wie bei Christian Justens Original-Skripten kann man aber alternativ ebenfalls erreichen, indem man Option „-o“ oder „--overwrite“ angibt. Nach erfolgreichem PDF<sup>L</sup>A<sub>T</sub>E<sub>X</sub>-Lauf und dem Kopieren des Ergebnisses ins aktuelle Arbeitsverzeichnis, wird das temporäre Verzeichnis samt Inhalt wieder gelöscht.

Eine weitere Änderung bei mir betrifft die Tatsache, dass man für die Verarbeitung mehrerer Dateien nicht mehrere Aufrufe benötigt, man kann auch in einem Aufruf beliebig oft Optionen und Dateien hintereinander anfügen, die dann nacheinander verarbeitet werden. So würde beispielsweise mit dem Aufruf:

```
a5toa4 -b foo.pdf -s bar.pdf -o bum.pdf
```

sowohl „foo-booklet.pdf“ als auch „bar-sidebyside.pdf“ erzeugt und anschließend „bum.pdf“ durch eine side-by-side-Fassung von sich selbst überschrieben werden.

Es ist zu beachten, dass nach jeder erzeugten Datei die ÜberschreibEinstellung von Option „-o“ oder „-overwrite“ aus Sicherheitsgründen wieder aufgehoben wird. Will man also mehrere Dateien nacheinander bearbeiten und bei mehreren davon soll die Ursprungsdatei überschrieben werden, so ist vor jeder dieser Dateien die Option erneut zu setzen.

`pfarrei.tlu` Tatsächlich ist „a5toa4.tlu“ aber nur ein Wrapper für „pfarrei.tlu“. Das wurde deshalb so gemacht, damit das eigentliche Programm leichter durch Versionen in `TEXMFLOCAL` oder `TEXMFHOME` ersetzt werden kann, ohne dass jedes Mal das *Binary* ersetzt werden muss.

Eine letzte kleine Änderung meiner Fassung betrifft die Anforderungen an das Seitenformat der Quelldateien. Im Original wurde mit der `pdfpages` Option `noautoscale` gearbeitet. Damit wurden die Seiten der Quelldatei nicht an das Seitenformat der Zieldatei angepasst. War die Quelldatei also nicht im Format A5, sondern beispielsweise A6, dann wurden zwei A6-Seiten auf einer A4-quer-Seite platziert. War die Quelldatei im Format A4, passten ihre Seiten nicht einmal auf die A4-quer-Seite. Ich habe diese Option daher weggelassen. Nun werden die Seiten der Quelldatei automatisch ins A5-Format gebracht, bevor sie auf der A4-quer-Seite platziert werden.

## a5toa4 Zusammenfassung

- s Um aus einem Quell-PDF „foo.pdf“ ein Ziel-PDF „foo-sidebyside.pdf“ zu erzeugen, bei dem die Seiten des Quell-PDF aufeinander folgend, nebeneinander auf einer A4-quer-Seite platziert sind, verwendet man:

```
a5toa4 -s foo.pdf
```

Soll die Ziel-PDF hingegen die Quell-PDF überschreiben, so gibt vor dem Dateinamen zusätzlich Option „-o“ an.

- b Um aus einem Quell-PDF „foo.pdf“ ein Ziel-PDF „foo-booklet.pdf“ zu erzeugen, bei dem die Seiten des Quell-PDF so auf einer A4-quer-Seite platziert sind, dass durch Falten in der Mitte ein Heft entsteht, verwendet man:

```
a5toa4 -b foo.pdf
```

- o Soll die Ziel-PDF hingegen die Quell-PDF überschreiben, so gibt vor dem Dateinamen zusätzlich Option „-o“ an.

-V Informationen über die verwendete Version von „a5toa4“ erhält man mit:

```
a5toa4 -V
```

-h Eine Hilfe zum Programm erhält man mit:

```
a5toa4 -h
```

## 2 Die L<sup>A</sup>T<sub>E</sub>X-Seite

Die L<sup>A</sup>T<sub>E</sub>X-Seite zu den Skripten aus [Abschnitt 1](#) besteht zunächst einmal aus dem L<sup>A</sup>T<sub>E</sub>X-Dokument „a5toa4.tex“. Dieses lädt wiederum das L<sup>A</sup>T<sub>E</sub>X-Paket „pfarrei.sty“. Die eigentliche Funktionalität verbirgt sich darin.

`\AvToAiv{Original-Datei}`  
`\OriginalFile` Die Anweisung `\AvToAiv` erledigt die Hauptarbeit für `a5toa4`. Die Voreinstellung für das optionale Argument *Original-Datei* ist `\OriginalFile`. Das Skript „a5toa4.tlu“ setzt dieses Makro entsprechend.

Man kann das L<sup>A</sup>T<sub>E</sub>X-Paket `pfarrei` aber auch per

```
\usepackage{pfarrei}
```

oder

```
\usepackage[booklet]{pfarrei}
```

direkt in seinem Dokument laden. Dann stellt es Umgebungen und Befehle für die Erstellung von Textblättern oder Textheften für Pfarrer, Lektoren und andere Mitwirkende an einem Gottesdienst oder auch für die Gemeinde bereit.

`\ifbooklet{Dann-Code}{Sonst-Code}`

Es kann Code davon abhängig ausgeführt werden, ob ein Booklet erzeugt wird oder nicht. Im Falle eines Booklets wird der *Dann-Code* ausgeführt, anderenfalls der *Sonst-Code*. Dies wird auch intern, beispielsweise innerhalb von `\AvToAiv` oder innerhalb der `booklet...page`-Umgebungen verwendet, um die entsprechende Entscheidung durchzuführen.

`bookletfrontpage`  
`\bookletfrontpagestyle` Mit der Umgebung `bookletfontpage` kann eine Titelseite erstellt werden, die nur bei der Erzeugung eines Booklets ausgegeben wird. Die Umgebung sollte immer am Anfang des Dokuments stehen, anderenfalls produziert sie eine Fehlermeldung. Dies geschieht, weil die Titelseite nun einmal die erste Seite sein sollte. Sollte der Seitenstil `empty` einmal nicht für die Titelseite gewünscht werden, kann man `\bookletfrontpagestyle` auf den Namen des gewünschten Stils umdefinieren.

`\motto{Motto}`  
`\titlepicture{Bild}`  
`\title{Titel}`  
`\parish{Gemeinde}`  
`\date{Datum}`

`\makebooklettitlepage` Ähnlich zu `\maketitle` bei den Standard-Klassen kann man für Booklets (aufbauend auf der oben erklärten Umgebung `bookletfrontpage` statt auf `titlepage`) mit Hilfe von `\makebooklettitlepage` eine Titelseite für ein Booklet erzeugen. Der Titel besteht aus einem Motto (oben auf der Seite), gefolgt von einem Titelbild, gefolgt von dem Titel, Angaben zur Gemeinde und zum Schluss das Datum. Es ist zu beachten, dass man `\makebooklettitlepage` *nicht* selbst in eine Umgebung packen sollte, da diese Anweisung intern die Umgebung `bookletfrontpage` verwendet. Darüber hinaus ist zu beachten, dass als Titelbild nicht einfach ein Dateinamen angegeben wird, sondern die kompletten Anweisungen, um ein Bild einzuladen oder zu erstellen. Auch muss ggf. ein Grafikpaket selbst geladen werden.

`bookletbackpage` Für ein Booklet kann mit der Umgebung `bookletbackpage` außerdem eine Rückseite definiert werden. Die Definition kann an beliebiger Stelle erfolgen. Die Ausgabe findet auf einer durch vier teilbaren Seite am Ende des Dokuments statt. Dazu werden ggf. leere Seiten eingefügt. Den Inhalt der *leeren Seiten* kann ebenfalls mit einer Umgebung, nämlich `bookletemptypage`, definiert werden. Wie bereits beim Seitenstil für die Vorderseite, kann auch der Seitenstil für die *leeren Seiten* und die Rückseite undefiniert werden.

`bookletemptypage`  
`\bookletbackpagestyle`  
`\bookletemptypagestyle`

`samedoublepage` Wenn der Inhalt dieser Umgebung auf die aktuelle Seite passt, wird er ausgegeben. Passt er nicht auf die aktuelle Seite und die aktuelle Seite ist eine Seite mit einer ungeraden Seitenzahl, dann wird zunächst ein Seitenumbruch durchgeführt. Anschließend wird der Inhalt ausgegeben. Dadurch wird vermieden, dass beispielsweise ein Lektor oder Pfarrer unnötig innerhalb eines Textes umblättern muss. Es ist zu beachten, dass innerhalb dieser Umgebung Fußnoten, Marginalien und Gleitumgebungen nicht korrekt funktionieren.

`\setupprayer{Optionen}`  
`prayer` Die Umgebung `prayer` ist für Gebete gedacht. Die Umgebung hat ein optionales Argument. Dieses ist identisch mit `Optionen` gilt dann aber nur lokal für diese Gebetsumgebung. Als `Optionen` kann eine Komma-separierte Liste von `Schlüssel`=`Wert`-Optionen angegeben werden. Folgende Schlüssel werden verstanden:

`\noresponder`

**leader**=`Vorbeter`: Setzt die Voreinstellung für das optionale Argument von `\item` auf `Vorbeter`.

**responder**=`Antwortende(r)`: Setzt die Voreinstellung für das optionale Argument des automatisch als Antwort eingefügten `\item` auf `Antwortende(r)`.

**response**=`Antwort`: Setzt die automatisch eingefügte Antwort auf `Antwort`.

Eine automatische Antwort wird nur eingefügt, wenn sowohl `responder` als auch `response` angegeben sind. Soll nur für einzelne `\item` keine automatische Antwort erstellt werden, so ist irgendwo innerhalb des entsprechenden `\item` (*nicht* jedoch innerhalb des optionalen Arguments von `\item`) oder danach ein `\noresponder` einzufügen.

### 3 Implementierung der L<sup>A</sup>T<sub>E</sub>X-Seite

#### 3.1 Das L<sup>A</sup>T<sub>E</sub>X-Dokument „a5toa4.tex“

Das ist wirklich geradezu trivial:

```

1 \documentclass[a4paper,landscape]{article}
2 \usepackage{pfarrei}
3 \begin{document}
4 \AvToAiv
5 \end{document}

```

Das war es schon. Trotzdem hat die Verwendung einer solchen zusätzlichen Datei den Vorteil, dass man bei Bedarf lokal auch ein ganz anderes „a5toa4.tex“ speichern kann.

#### 3.2 Das L<sup>A</sup>T<sub>E</sub>X-Paket „pfarrei“

`\ifbooklet` Wir haben eine Option, die bestimmt, ob wir ein Booklet erzeugen oder nicht. Später wird das schlicht über das Makro `\ifbooklet` festgestellt. Falls ein Booklet erzeugt werden soll wird das erste, sonst das zweite Argument ausgeführt.

```

6 \newcommand*{\ifbooklet}{%
7 \let\ifbooklet\@secondoftwo
8 \DeclareOption{booklet}{\let\ifbooklet\@firstoftwo}
9 \DeclareOption{nobooklet}{\let\ifbooklet\@secondoftwo}

```

Dann werden die Optionen ausgeführt:

```
10 \ProcessOptions*
```

Dann benötigen wir ein paar Pakete:

```
11 \RequirePackage{ifpdf}
12 \RequirePackage{pdfpages}

```

`\AvToAiv` Der Name des Makros kommt von `a5toa4`. Es muss sichergestellt sein, dass dies im PDF-Modus verwendet wird.

```

13 \newcommand*{\AvToAiv}[1][\OriginalFile]{%
14 \ifpdf\else
15 \PackageError{pfarrei}{PDF mode needed}{%
16 a5toa4 needs the direct PDF mode.\MessageBreak
17 Usually this may be activated using either pdflatex, lualatex or
18 xelatex.%
19 }%
20 \input{x.tex}

```

```

21 \fi
22 \ifbooklet{%
23   \includepdf[pages=-,booklet]{#1}%
24 }{%
25   \includepdf[pages=-,nup=2x1]{#1}%
26 }%
27 }

```

`bookletpage` Da mehrfach eine Umgebung für einzelne Seiten benötigt wird, definieren wir dafür eine eigene Umgebung, die dann entsprechend individualisiert wird. Es ist zu beachten, dass diese Umgebung nicht als Umgebung aufgerufen werden sollte, sondern direkt `\bookletpage` und `\endbookletpage`.

```

28 \newenvironment*{bookletpage}{%
29   \edef\reserved@a{bookletpage}%
30   \ifx\@currenvir\reserved@a
31     \PackageError{pfarrei}{‘bookletpage’ used as ordinary environment}{%
32       Please note, that ‘bookletpage’ is a virtual environment
33       only.\MessageBreak
34       You should not use it directly as an environment, but use
35       ‘\string\bookletpage’ at the\MessageBreak
36       begin code and ‘\string\endbookletpage’ at the end code of a wrapper
37       environment.\MessageBreak
38       If you’ll continue, expect several additional errors%
39     }%
40     \let\bookletpagebox\@tempboxa
41     \let\bookletpagestyle\@empty
42   \else
43     \expandafter\let\expandafter\bookletpagebox
44     \csname \@currenvir box\endcsname
45     \expandafter\let\expandafter\bookletpagestyle
46     \csname \@currenvir style\endcsname
47   \fi
48   \edef\reserved@a{%
49     \noexpand\begin{lrbox}{\bookletpagebox}
50     \noexpand\begin{minipage}[t][\textheight][t]{\textwidth}%
51       \begingroup
52         \def\noexpand\@currenvir{\@currenvir}%
53         \def\noexpand\@currenvline{\@currenvline}%
54         \noexpand\parskip=\the\parskip
55         \noexpand\parindent=\the\parindent
56         \noexpand\parfillskip=\the\parfillskip
57       }\reserved@a
58 }{%

```

Das Ende wird in zwei Portionen aufgeteilt:

```

59 \endbookletpagebox
60 \printbookletpagebox
61 }

```

`\endbookletpagebox` das Beenden der Erzeugung der Box für die Seite

```

62 \def\endbookletpagebox{%
63     \par
64     \endgroup
65     \end{minipage}%
66     \end{lrbox}%
67     \global\setbox\bookletpagebox\box\bookletpagebox
68 }

```

`\printbookletpagebox` und die Ausgabe der Box, falls ein Booklet erzeugt wird.

```

69 \newcommand*\printbookletpagebox[1][\@currentenv]{%
70     \ifbooklet{%
71         \ifundefined{bookletpagestyle}{%
72             \ifx\bookletpagestyle\@empty\else\thispagestyle{bookletpagestyle}\fi%
73         }%
74         \clearpage\noindent\usebox\bookletpagebox\clearpage
75     }{%
76         \PackageInfo{pfarrei}{'#1' not printed}%
77     }%
78 }

```

`bookletfrontpage` Umgebung, um eine Booklet-Titelseite zu erzeugen.

`\bookletfrontpagebox` Es ist nur eine Seite erlaubt. Also packen wir das ganze in eine Box. Dafür brauchen wir eine solche:

```

79 \newsavebox\bookletfrontpagebox
80 \newcommand*\bookletfrontpagestyle{empty}
81 \newenvironment*{bookletfrontpage}{%
82     \bookletpage
83 }{%
84     \endbookletpagebox
85     \clearpage
86     \ifbooklet{%
87         \ifnum\c@page>\@ne
88             \PackageError{pfarrei}{Booklet front page not first page}{%
89                 The booklet font page should be the first page, but it seems, that it
90                 is\MessageBreak
91                 page no. \the\c@page. \space Maybe you should put it immediately after
92                 '\string\begin{document}.'%
93                 Nevertheless, if you'll continue it will be printed here%
94             }%
95         \fi
96         \printbookletpagebox
97     }{%
98         \PackageInfo{pfarrei}{Booklet front page ignored}%
99     }%
100 }

```

`bookletbackpage` Im Prinzip ist die Booklet-Rückseite fast wie bei der Titelseite, allerdings fügen wir hier so viele Leerseiten ein, dass wir wirklich auf einer durch 4 teilbaren Seite landen.

`bookletemptypage` Die Umgebung für die konfigurierbare *Leerseite*, besitzt keine eigene Ausgabe, sondern sammelt tatsächlich nur den Inhalt. Die Ausgabe erfolgt dann bei der  
`\bookletemptypagebox` Ausgabe der Booklet-Rückseite.  
`\bookletemptypagestyle`

```

101 \newsavebox\bookletemptypagebox
102 \newcommand*\bookletemptypagestyle{}{empty}
103 \newenvironment*{bookletemptypage}{%
104   \bookletpage
105 }{%
106   \endbookletpagebox
107 }
108 \newsavebox\bookletbackpagebox
109 \newcommand*\bookletbackpagestyle{}{empty}
110 \newenvironment*{bookletbackpage}{%
111   \bookletpage
112 }{%
113   \endbookletpagebox
114   \if@filesw\immediate\write\@mainaux{\string\printbookletbackpage}\fi
115 }%
```

`\printbookletbackpage` Die eigentliche Ausgabe der letzten Seite geschieht über einen Eintrag in der `aux-`  
`\@printbookletbackpage` Datei. Das ist zwar nicht ganz sauber (und `scrfile` meckert deshalb ggf.), aber die Standard-Klasse `letter` macht das ähnlich für die Etiketten. Also hoffen wir einfach, dass es gut geht.

```

116 \newcommand*\printbookletbackpage{}{}
117 \newcommand*\@printbookletbackpage{}{%
118   \ifbooklet{%
119     \clearpage
120     \let\bookletpagestyle\bookletemptypagestyle
121     \ifvoid\bookletemptypagebox
122       \let\bookletpagebox\strutbox
123     \else
124       \let\bookletpagebox\bookletemptypagebox
125     \fi
126     \@tempcnta=\c@page
127     \divide\@tempcnta by 4
128     \multiply\@tempcnta by 4
129     \ifnum \@tempcnta=\c@page\else
130       \advance\@tempcnta by 4
131       \@whilenum \c@page<\@tempcnta\do{%
132         \printbookletpagebox
133       }%
134     \fi
135     \let\bookletpagestyle\bookletbackpagestyle
136     \let\bookletpagebox\bookletbackpagebox
137     \printbookletpagebox
138   }{%
139     \PackageInfo{pfarrei}{Booklet back page ignored}%
140   }%
141 }
```

Das Drucken darf während des Lesens der `aux`-Datei in `\begindocument` nicht erfolgen. Daher definieren wir es erst danach entsprechend um. Dadurch ist es während des Lesens der `aux`-Datei in `\enddocument` funktionsfähig.

```
142 \AtBeginDocument{\let\printbookletbackpage\@printbookletbackpage}
```

`\makebooklettitlepage` Der Standard-Titel für Booklets besteht aus einem Motto, einem Bild, einem Titel, der Gemeinde und dem Datum. Wir gehen davon aus, dass die Klasse bereits `\@motto` `\date`, `\@date`, `\title` und `\@title` zur Verfügung stellt. Allerdings ergibt die `\titlepicture` Voreinstellung `\today` für das Datum hier wenig Sinn, weshalb die Voreinstellung `\@titlepicture` in einen leeren Wert geändert wird. Den Rest machen wir komplett selbst:

```

\title 143 \newcommand*\motto}[1]{\gdef\@motto{#1}}
\@title 144 \newcommand*\@motto{}
\parish 145 \newcommand*\titlepicture}[1]{\gdef\@titlepicture{#1}}
\@parish 146 \newcommand*\@titlepicture{}
\date 147 \providecommand*\title}[1]{\gdef\@title{#1}}
\@date 148 \providecommand*\@title{}
149 \newcommand*\parish}[1]{\gdef\@parish{#1}}
150 \newcommand*\@parish{}
151 \providecommand*\date}[1]{\gdef\@date{#1}}
152 \def\@date{}
153 \newcommand*\makebooklettitlepage}{%
154 \begin{bookletfrontpage}
155 \parskip.5\baselineskip
156 \parindent\z@
157 \parfillskip \z@ \@plus 1fil
158 \centering
159 \ifx\@motto\@empty\else{\Huge\@motto\par}\fi
160 \vfill
161 \ifx\@titlepicture\@empty\else\@titlepicture\par\fi
162 \vfill
163 \parskip\z@
164 \Huge
165 \@title\par
166 \@parish\par
167 \@date\par
168 \ifx\@title\@empty\ifx\@parish\@empty\ifx\@date\@empty\null\fi\fi\fi
169 \end{bookletfrontpage}
170 }
```

`samedoublepage` Es wird etwas umständlich eine vertikale Box gespeichert,

```

\samedoublepage@save@hbox 171 \let\samedoublepage@save@hbox\hbox
172 \newenvironment*\samedoublepage}{%
173 \par
174 \let\hbox\vbox
175 \begin{lrbox}{\@tempboxa}%
176 \let\hbox\samedoublepage@save@hbox
177 }{%
178 \end{lrbox}%
179 \let\hbox\samedoublepage@save@hbox
```

um diese dann in Stücke zu zerschneiden, die auf eine Seite passen,

```
180 \@tempdima\ifdim\pagegoal=\maxdimen\textheight
181     \else\dimexpr\pagegoal-\pagetotal\fi
182 \ifdim \@tempdima
183     <\dimexpr\ht\@tempboxa+\dp\@tempboxa\relax
184     \ifodd\c@page
```

Wobei aber nicht auf ungeraden Seiten mit nur einem Stück des Kuchens begonnen wird.

```
185     \newpage
186     \@tempdima\textheight
187 \else \typeout{even page}%
188 \fi
189 \@whiledim \@tempdima
190     <\dimexpr\ht\@tempboxa+\dp\@tempboxa\relax\do{%
191     \splitmaxdepth\dp\strutbox
192     \splittopskip\topskip
193     \setbox\z@\vsplit\@tempboxa to \@tempdima
194     \usebox\z@
195     \newpage
196     \@tempdima\textheight
197 }%
198 \fi
```

Und am Ende nicht vergessen, den Rest der Box auch noch auszugeben.

```
199 \ifvoid\@tempboxa\else\usebox\@tempboxa\fi
200 }
```

`prayer` Die Gebetsumgebung wird mit Hilfe von `keyval` definiert. Sie kann außer mit dem optionalen Argument auch jederzeit mit `\setupprayer` konfiguriert werden.

`\prayer@responder` Dafür werden mehrere interne Makros benötigt, um die gewünschten Werte aufzunehmen.

```
\prayer@response
\prayer@leader
201 \RequirePackage{keyval}
202 \define@key{pfarrei.prayer}{response}{\def\prayer@response{#1}}
203 \define@key{pfarrei.prayer}{responder}{\def\prayer@responder{#1:}}
204 \define@key{pfarrei.prayer}{leader}{\def\prayer@leader{#1:}}
205 \newcommand*\prayer@responder{}
206 \newcommand*\prayer@response{}
207 \newcommand*\prayer@leader{}
208 \newcommand*\setupprayer{%
209     \setkeys{pfarrei.prayer}%
210 }
```

`\ifprayer@firstitem` Da die Antwort beim ersten `\item` noch nicht ausgegeben werden darf, sondern nur nach den nachfolgenden, muss die Information, ob es das erste `\item` ist, über einen Schalter gespeichert werden.

```
211 \newif\ifprayer@firstitem
212 \newcommand*\prayer@response@item{%
213     \ifprayer@firstitem\else
```

```

214 \ifx\prayer@responder\@empty\else
215 \ifx\prayer@response\@empty\else
216 \prayer@save@item[\prayer@responder] \prayer@response
217 \fi
218 \fi
219 \fi
220 }

```

`\prayer@item` Die `\item`-Anweisung von `prayer` ist etwas anders als von anderen Listen-  
`\prayer@save@item` Umgebungen. Sie baut jedoch auf der Originaldefinition auf. Daher muss die Originaldefinition gespeichert und entsprechend erweitert verwendet werden.

```

221 \newcommand*\prayer@item[1][\prayer@leader]{%
222 \prayer@response@item
223 \prayer@firstitemfalse
224 \prayer@save@item[#{1}]%
225 }

```

`\noresponder` Das Abschalten der nächsten automatischen Antwort geschieht einfach, indem so  
getan wird, als wäre das nächste `\item` wieder das erste.

```

226 \newenvironment*{prayer}[1][ ]{%
227 \begin{description}
228 \begin{group}
229 \def\@currenvir{prayer}%
230 \setupprayer{#1}%
231 \let\prayer@save@item\item
232 \let\item\prayer@item
233 \prayer@firstitemtrue
234 \let\noresponder\prayer@firstitemtrue
235 }{%
236 \prayer@response@item
237 \end{group}
238 \end{description}
239 }

```

## 4 Implementierung der Skripten

### 4.1 Der kleine Wrapper „a5toa4.tlu“

```

240 -- $Id: pfarrei.dtx 36 2013-10-16 18:54:26Z mjk $
241
242 kpse.set_program_name(arg[-1], 'a5toa4')
243 require('pfarrei.pfarrei')

```

### 4.2 Das Haupt-Skript „pfarrei.tlu“

```

244 local version_number = string.sub( '$Revision: 36 $', 12, -2 )
245 local action_version = ' r' .. version_number .. '\n' .. [[
246
247 Copyright (c) 2013 Markus Kohm.

```

```

248 License: lppl 1.3c or later. See <http://www.latex-project.org/lppl.txt>.
249 ]]
250 local action_help = [[
251 action options:
252
253 -h, --help          Print this help message.
254 -V, --version       Print the version information.
255
256 processing options:
257 -b, --booklet       Generate a booklet instead of only two pages side by
258                     side onto one page. The whole booklet will be one
259                     signature.
260 -s, --sidebyside    Generate only two pages side by side onto one page
261                     instead of a booklet.
262 -o, --overwrite     Write the output to the <PDF file> instead of appending
263                     "-sidebyside.pdf" or "--booklet.pdf" to the basename
264                     of <PDF file>
265 ]]
266 local action_opts = {
267   ['-h']             = 'help',
268   ['--help']         = 'help',
269   ['-V']             = 'version',
270   ['--version']     = 'version',
271 }
272 local processing_opts = {
273   ['-b']             = 'booklet',
274   ['--booklet']     = 'booklet',
275   ['-s']             = 'sidebyside',
276   ['--sidebyside'] = 'sidebyside',
277   ['-o']             = 'overwrite',
278   ['--overwrite']  = 'overwrite',
279   ['-d']             = 'debug',
280   ['--debug']      = 'debug',
281 }
282
283 -- detect action options and do action
284 local i = 1
285 local action
286 while arg[i] do
287   action = action_opts[arg[i]]
288   i = i+1
289   if action == 'help' then
290     print( arg[0]..action_version );
291     print( 'Usage: ' .. arg[0] .. ' <action option>' )
292     print( '          ' .. arg[0] .. ' [<processing options>] <PDF file> ...\n' )
293     print( action_help );
294     os.exit( 0 );
295   elseif action == 'version' then
296     print( arg[0] .. action_version );
297     os.exit( 0 );

```

```

298 end
299 end
300
301 -- process options and parameters
302 local booklet = false
303 local overwrite = false
304 local debug = false
305 i = 1
306 while arg[i] do
307   action = processing_opts[arg[i]]
308   if action == 'booklet' then booklet = true
309   elseif action == 'sidebyside' then booklet = false
310   elseif action == 'overwrite' then overwrite = true
311   elseif action == 'debug' then debug = true
312   elseif action == nil then
313     -- build the temporary tex file
314     local tmpdir = os.tmpdir("pfarrei.XXXXXX" )
315     local tmpfile = string.match( arg[i], '.*/(.*)$') or arg[i]
316     local basename = string.match( tmpfile, '(*)%.[^\%]*$') or tmpfile
317     tmpfile = tmpdir..'/'..basename..' .tex'
318     local file = assert( io.open( tmpfile, 'w' ) )
319     if booklet then assert( file:write("\\PassOptionsToPackage{booklet}{pfarrei}\n") ) end
320     assert( file:write("\\def\\OriginalFile{"..arg[i].."}\n") )
321     assert( file:write("\\input{a5toa4.tex}\n") )
322     assert( file:flush() )
323     file:close()
324     -- call pdflatex
325     assert( os.execute( 'pdflatex -interaction=batchmode -output-directory='..tmpdir..' '..tmpfile ) )
326     -- copy the resulting pdf file
327     local srcfile = assert( io.open( tmpdir..'/'..basename..' .pdf', 'rb' ) )
328     if overwrite
329     then
330       tmpfile = arg[i]
331     else
332       tmpfile = string.match( arg[i], '(*)%.[^\%]*$' ) or arg[i]
333       if booklet
334       then
335         tmpfile = tmpfile..' -booklet.pdf'
336       else
337         tmpfile = tmpfile..' -sidebyside.pdf'
338       end
339     end
340     local destfile = assert( io.open( tmpfile, 'wb' ) )
341     local buffer
342     while true do
343       buffer = srcfile:read(8388608)
344       if buffer==nil then break end
345       assert( destfile:write(buffer) )
346     end
347     assert( destfile:close() )

```

```

348     srcfile:close()
349     if debug
350     then
351         print('DEBUG: Temporary files in: '..tmpdir);
352     else
353         tmpfile=tmpdir..'/'..basename
354         os.remove( tmpfile..'aux' )
355         os.remove( tmpfile..'tex' )
356         os.remove( tmpfile..'log' )
357         os.remove( tmpfile..'pdf' )
358         os.remove( tmpdir )
359     end
360     overwrite = false
361 end
362 i=i+1
363 end

```

## Change History

r26			
	\@date: neues internes Makro ...	10	
	\@motto: neues internes Makro ..	10	
	\@parish: neues internes Makro .	10	
	\@printbookletbackpage: neue interne Anweisung .....	9	
	\@title: neues internes Makro ..	10	
	\@titlepicture: neues internes Makro .....	10	
	\bookletbackpagebox: neue Box ..	8	
	\bookletbackpagestyle: neues Makro .....	8	
	\bookletemptypagebox: neue Umgebung .....	9	
	\bookletemptypagestyle: neues Makro .....	9	
	\bookletfrontpagebox: neue Box ..	8	
	\bookletfrontpagestyle: neues Makro .....	8	
	\date: neue Anweisung .....	10	
	\makebooklettitlepage: neue Anweisung .....	10	
	\motto: neue Anweisung .....	10	
	bookletbackpage: neue Umgebung ..	8	
	bookletemptypage: neue Box .....	9	
	bookletfrontpage: neue Umgebung .....	8	
	\parish: neue Anweisung .....	10	
	\printbookletbackpage: neue Anweisung .....	9	
	\title: neue Anweisung .....	10	
	\titlepicture: neue Anweisung ..	10	
r28			
	samedoublepage: neue Umgebung ..	10	
	\endbookletpagebox: neue (eigentlich interne) Anweisung .....	7	
	\ifbooklet: neue Anweisung .....	6	
	\ifprayer@firstitem: neuer interner Schalter .....	11	
	\noresponder: neue Anweisung ..	12	
	bookletpage: neue virtuelle Umgebung .....	7	
	\prayer@item: neue interne Anweisung .....	12	
	\prayer@leader: neues internes Makro .....	11	
	\prayer@responder: neues internes Makro .....	11	
	\prayer@response: neues internes Makro .....	11	
	\prayer@save@item: neue interne Anweisung .....	12	
	\printbookletpagebox: neue (eigentlich interne) Anweisung ...	8	
	prayer: neue Umgebung .....	11	

<code>\prayer@response@item</code> : neue interne Anweisung . . . . .	11	<code>es internes Makro</code> . . . . .	10
<code>\samedoublepage@save@hbox</code> : neu-		<code>\setupprayer</code> : neue Anweisung . . . . .	11

## Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

Symbols		O	
<code>\@date</code> . . . . .	<u>143</u>	<code>\OriginalFile</code> . . . . .	<i>4</i>
<code>\@motto</code> . . . . .	<u>143</u>		
<code>\@parish</code> . . . . .	<u>143</u>		
<code>\@printbookletbackpage</code> . . . . .	<u>116</u>		
<code>\@title</code> . . . . .	<u>143</u>		
<code>\@titlepicture</code> . . . . .	<u>143</u>		
A		P	
<code>\AvToAiv</code> . . . . .	<i>4, 13</i>	<code>\parish</code> . . . . .	<i>5, 143</i>
		<code>prayer</code> (environment) . . . . .	<i>5, 201</i>
B		<code>\prayer@item</code> . . . . .	<u>221</u>
<code>bookletbackpage</code> (environment) . . . . .	<i>5, 101</i>	<code>\prayer@leader</code> . . . . .	<u>201</u>
<code>\bookletbackpagebox</code> . . . . .	<u>101</u>	<code>\prayer@responder</code> . . . . .	<u>201</u>
<code>\bookletbackpagestyle</code> . . . . .	<i>5, 101</i>	<code>\prayer@response</code> . . . . .	<u>201</u>
<code>bookletemptypage</code> (environment) . . . . .	<i>5, 101</i>	<code>\prayer@response@item</code> . . . . .	<u>211</u>
<code>\bookletemptypagebox</code> . . . . .	<u>101</u>	<code>\prayer@save@item</code> . . . . .	<u>221</u>
<code>\bookletemptypagestyle</code> . . . . .	<i>5, 101</i>	<code>\printbookletbackpage</code> . . . . .	<u>116</u>
<code>bookletfrontpage</code> (environment) . . . . .	<i>4, 79</i>	<code>\printbookletpagebox</code> . . . . .	<u>69</u>
<code>\bookletfrontpagebox</code> . . . . .	<u>79</u>		
<code>\bookletfrontpagestyle</code> . . . . .	<i>4, 79</i>		
		S	
		<code>samedoublepage</code> (environment) . . . . .	<i>5, 171</i>
		<code>\samedoublepage@save@hbox</code> . . . . .	<u>171</u>
		<code>\setupprayer</code> . . . . .	<i>5, 201</i>
C		T	
		<code>\title</code> . . . . .	<i>5, 143</i>
		<code>\titlepicture</code> . . . . .	<i>5, 143</i>
D			
<code>\date</code> . . . . .	<i>5, 143</i>		
E			
<code>\endbookletpagebox</code> . . . . .	<u>62</u>		
environments:			
<code>bookletbackpage</code> . . . . .	<i>5, 101</i>		
<code>bookletemptypage</code> . . . . .	<i>5, 101</i>		
<code>bookletfrontpage</code> . . . . .	<i>4, 79</i>		
<code>bookletpage</code> . . . . .	<u>28</u>		
<code>prayer</code> . . . . .	<i>5, 201</i>		
<code>samedoublepage</code> . . . . .	<i>5, 171</i>		
I			
<code>\ifbooklet</code> . . . . .	<i>4, 6</i>		
<code>\ifprayer@firstitem</code> . . . . .	<u>211</u>		
M			
<code>\makebookletttitlepage</code> . . . . .	<i>5, 143</i>		
<code>\motto</code> . . . . .	<i>5, 143</i>		
N			
<code>\noresponder</code> . . . . .	<i>5, 226</i>		