Das filecontents Paket*

Scott Pakin
scott+fc@pakin.org

Übersetzung ins Deutsche:
Ronny Berndt
ronny.berndt@uni-jena.de

17. September 2011

1 Einleitung

`filecontents` Es existiert eine weniger bekannte Umgebung namens `filecontents`, welche in LaTeX 2ε integriert ist. Eine Beschreibung von `filecontents`, welche aus `ltclass.dtx` stammt, lautet folgendermaßen:

> Die `filecontents` Umgebung ist für die Weitergabe von Inhalten aus Paketen, Optionen oder anderen Dateien bestimmt. Hierdurch werden die Inhalte aus mehreren Dateien in eine Einzelne eingefügt. Der Befehl benötigt ein Argument (den Dateinamen), wodurch eine neue Datei erzeugt wird. Wenn diese schon existiert (im aktuellen Verzeichnis, falls das Betriebssystem eine Notation für das 'aktuelles Verzeichnis' oder 'Standard Verzeichnis' bereit stellt) passiert nichts (außer dass eine Information ausgegeben wird) und der Inhalt des Bereiches der Umgebung wird übersprungen. Andernfalls wird der Inhalt wörtlich in die Datei geschrieben, dessen Namen man als erstes Argument angegeben hat. Zusätzlich werden noch Kommentare in diese Datei geschrieben um den Benutzer zu zeigen, wie diese erzeugt wurde.
>
> Die Umgebung ist nur vor `\documentclass` erlaubt, um sicherzustellen, dass alle Pakete oder Optionen, welche für diesen speziellen Durchlauf gebraucht werden, vorhanden sind. Die `begin`- bzw. `end`-Marke sollten jeweils für sich auf einer einzelnen Zeile stehen. Es besteht die Möglichkeit die Sternform des Befehls zu verwenden, welche keinerlei Kommentare in die Datei mit einfügt.

*Dateiversion v1.2, überarbeitet 2009/03/17.

(Der Hinweis, dass `filecontents` nur vor `\documentclass` verwendet werden kann, ist schlichtweg falsch. `filecontents` kann überall in der Präambel verwendet werden.)

Das filecontents-Paket bietet eine verbesserte Version der `filecontents`- und `filecontents*`-Umgebung, welche die zwei oben genannten Einschränkungen (Überschreiben von vorhandene Dateien und das `filecontents` vor der Deklaration von `\documentclass` verwendet werden muss) aufhebt. (In Wirklichkeit ist es nämlich `\begin{document}`). filecontents bietet damit einen bequemeren Weg, um externe Dateien in ein LaTeX Dokument zu schreiben, als es der LaTeX 2_ε-Kernel bietet.

Verwendung `filecontents` funktioniert ähnlich wie `verbatim`, außer dass es zwingend einen Dateinamen als Argument fordert:

```
\begin{filecontents}{myfile.tex}
This text gets written to \texttt{myfile.tex}.
\end{filecontents}
```

Der vorangegangene Quelltext wird eine Datei names `myfile.tex` erzeugen und mit folgendem ähnlichen Inhalt füllen:

```
%% LaTeX2e file `myfile.tex'
%% generated by the `filecontents' environment
%% from source `mydocument' on 2001/07/31.
%%
This text gets written to \texttt{myfile.tex}.
```

`myfile.tex` kann daraufhin durch `\include` oder `\input` zurück in das Dokument eingefügt werden. Hat man stattdesssen `filecontents*` benutzt, wird in die Datei nur die Zeile "`This text gets written to \texttt{myfile.tex}.`" geschrieben. `filecontents*` ist demzufolge sinnvoll zum Schreiben von nicht-LaTeX Dateien, wie zum Beispiel eingebetteden `PostScript`-Dateien.

Wenn Sie das Paket ltxtable benutzen, werden Sie filecontents besonders nützlich finden. ltxtable ist eine grobe Ansammlung von longtable, welche über Seitengrenzen hinaus gehen und tabularx, wodurch sich Tabellen auf eine bestimmte Breite weiten lassen. ltxtable's Schnittstelle ist ein bisschen unhandlich. Diese fordert, dass sich die `longtable` Umgebung in einer separaten Datei befindet. Mit dem filecontents-Paket können Sie diese Datei direkt vor dem Aufruf von `\LTXtable` erzeugen – eine weit bequemere Alternative, als die Tabelle manuell in einer separaten Datei zu platzieren.

2 Implementierung

Die meisten Benutzer können an diesem Punkt mit dem Lesen aufhören. Der Implementierungsteil enthält den kommentierten Quelltext für das filecontents-

Paket, welches nur für Personen von Interesse ist, welche eine ausführliche und genaue Erklärung, wie filecontents funktioniert, benötigen.

Ehre, wem Ehre gebührt. Ich habe praktisch keinen Quelltext von filecontents selbst geschrieben. Dieser kommt fast ausschließlich von dem LaTeX 2_ε-Quellcode, insbesondere aus der Datei `ltclass.dtx`, welcher Frank Mittelbach, Chris Rowley, Alan Jeffrey, und David Carlisle zugeschrieben werden. Ich habe diesen durch ein paar kleine Änderungen (unten durch Quelltextblöcke und Kommentare angezeigt) angepasst, um die `filecontents`- Umgebung bequemer zu benutzen.

```
1 ⟨*package⟩
```

`\filec@ntents` Sofern nicht anders angegeben, wurde der Quelltext (einschließlich der Kommentare) des `\filec@ntents` Makro wörtlich aus `ltclass.dtx` übernommen.

```
2 \begingroup%
3 \catcode‘\*=11 %
4 \catcode‘\^^M\active%
5 \catcode‘\^^L\active\let^^L\relax%
6 \catcode‘\^^I\active%

7 \gdef\filec@ntents#1{%
8   \openin\@inputcheck#1 %
```

Im ursprünglichen Quelltext wurde eine existierende Datei nicht überschrieben. In der neuen Version dient die Existenzprüfung der Datei ausschließlich zur Entscheidung, ob "`Writing file ‘`⟨*filename*⟩`’`" oder "`Overwriting file ‘`⟨*filename*⟩`’`" ausgegeben wird. Die Ablaufsteuerung dient dazu, immer den `\ifeof` Zweig (Datei existiert nicht; öffne diese) und nie den `\else` Zweig (Datei existiert; mache nichts) auszuführen.

```
9   \ifeof\@inputcheck%
10    \@latex@warning@no@line%
11        {Writing file ‘\@currdir#1’}%
12  \else %
13    \@latex@warning@no@line%
14        {Overwriting file ‘\@currdir#1’}%
15  \fi %

16  \chardef\reserved@c15 %
17  \ch@ck7\reserved@c\write%
18  \immediate\openout\reserved@c#1\relax%

19  \if@tempswa%
20    \immediate\write\reserved@c{%
21      \@percentchar\@percentchar\space%
22          \expandafter\@gobble\string\LaTeX2e file ‘#1’^^J%
23      \@percentchar\@percentchar\space  generated by the %
24        ‘\@currenvir’ \expandafter\@gobblefour\string\newenvironment^^J%
25      \@percentchar\@percentchar\space from source ‘\jobname’ on %
26         \number\year/\two@digits\month/\two@digits\day.^^J%
```

```
      \@percentchar\@percentchar}%
  \fi%
  \let\do\@makeother\dospecials%
```

Das inputenc-Paket hat unter Umständen einige der oberen 128 Zeichencodes als "aktiv" gekennzeichnet (category code 13), welches filecontents verwirrt. Deshalb markieren wir jeden der oberen 128 Zeichencodes als "Buchstabe" (category code 11), um sie korrekt in eine Datei schreiben zu können.

```
  \count0=128\relax %
  \loop %
    \catcode\count0=11\relax %
    \advance\count0 by 1\relax %
    \ifnum\count0<256 %
  \repeat %
```

```
  \edef\E{\@backslashchar end\string{\@currenvir\string}}%
  \edef\reserved@b{%
    \def\noexpand\reserved@b%
         ####1\E####2\E####3\relax}%
  \reserved@b{%
    \ifx\relax##3\relax%
```

Es gab kein `\end{filecontents}`

```
      \immediate\write\reserved@c{##1}%
    \else%
```

Es existierte ein `\end{filecontents}`, stoppe diesmal.

```
      \edef^^M{\noexpand\end{\@currenvir}}%
      \ifx\relax##1\relax%
      \else%
```

Text vor dem `\end`, schreibe diesen mit einer Warnung.

```
          \@latex@warning{Writing text `##1' before %
             \string\end{\@currenvir}\MessageBreak as last line of #1}%
        \immediate\write\reserved@c{##1}%
      \fi%
      \ifx\relax##2\relax%
      \else%
```

Text nach dem `\end`, ignoriere diesen mit einer Warnung.

```
        \@latex@warning{%
          Ignoring text `##2' after \string\end{\@currenvir}}%
      \fi%
    \fi%
    ^^M}%
```

```
  \catcode`\^^L\active%
  \let\L\@undefined%
  \def^^L{\@ifundefined L^^J^^J^^J}%
  \catcode`\^^I\active%
```

```
  \let\I\@undefined%
  \def^^I{\@ifundefined I\space\space}%
  \catcode`\^^M\active%
  \edef^^M##1^^M{%
    \noexpand\reserved@b##1\E\E\relax}}%
\endgroup%
```

`\fc@no@preamblecmds` LaTeX 2_ε deklariert `\filecontents`, `\filecontents*` und alle zugehörigen Hilfsmakros als `\@onlypreamble`, was dazu führt, dass diese nach einem `\begin{document}` unwirksam werden. Der nachfolgende Code aktiviert die Wiederverwendung dieser Befehle überall im Dokument. Dieser wurde aus dem Paket pkgindoc (generiert durch `ltclass.dtx`) entnommen und verändert. Dabei wurden nur die Befehle angepasst, die durch filecontents gebraucht werden, um diese wieder zu aktivieren. Andere Klassen- und Paketoptionen, sowie verarbeitende Befehle, blieben unberührt.

```
\def\fc@no@preamblecmds#1\do\filecontents#2\do\filec@ntents#3\relax{%
  \gdef\@preamblecmds{#1#3}}
\expandafter\fc@no@preamblecmds\@preamblecmds\relax
```

⟨/package⟩