The **filecontents** package[*]

Scott Pakin
scott+fc@pakin.org

May 31, 2018

1   Introduction

filecontents   There is a little-known environment called `filecontents` that is built into LaTeX 2_ε. Here is `filecontents`' description, which was taken from `ltclass.dtx`:

> The environment `filecontents` is intended for passing the contents of packages, options, or other files along with a document in a single file. It has one argument, which is the name of the file to create. If that file already exists (maybe only in the current directory if the OS supports a notion of a 'current directory' or 'default directory') then nothing happens (except for an information message) and the body of the environment is bypassed. Otherwise, the body of the environment is written verbatim to the file name given as the first argument, together with some comments about how it was produced.
>
> The environment is allowed only before `\documentclass` to ensure that all packages or options necessary for this particular run are present when needed. The begin and end tags should each be on a line by itself. There is also a star-form; this does not write extra comments into the file.

(The comment about `filecontents` being valid only before `\documentclass` is, in fact, untrue. `filecontents` is allowed anywhere in the document's preamble.)

The filecontents package provides a hacked-up version of the `filecontents` and `filecontents*` environments that lifts the two restrictions stated above, namely that existing files are never overwritten and that `filecontents` must be used before the `\documentclass` declaration (really, the `\begin{document}`). filecontents is therefore a more convenient way to write external files from within a LaTeX document than is provided by default by the LaTeX 2_ε kernel.

---

[*]This file has version number v1.4, last revised 2018/05/30.

Sample usage   filecontents works much like verbatim, except that it takes a mandatory filename argument:

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

The preceding code will write a myfile.tex file with contents resembling the following:

```
%% 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 can then be incorporated back into the document with \include or \input. Had filecontents* been used instead of filecontents, the file would have contained only the "This text gets written to \texttt{myfile.tex}." line. filecontents* is therefore useful for writing non-LaTeX files such as Encapsulated PostScript files.

If you use the ltxtable package you may find filecontents particularly useful. ltxtable is a crude conglomeration of longtable, which allows tables to cross page boundaries, and tabularx, which enables tables to stretch to a specified width. ltxtable's interface is a bit cumbersome, however; it requires that the longtable environment be contained in a separate file. With the filecontents package you can create this file right before the \LTXtable invocation—a far more convenient alternative than having to manually place the table within a separate file.

2   Implementation

Most users can stop reading at this point. The Implementation section contains the annotated source code for the filecontents package itself, which is useful only to people who want a detailed and precise explanation of how filecontents works.

To give credit where credit is due, I wrote virtually none of the filecontents code myself. It comes almost exclusively from the LaTeX 2_ε source code, specifically from the file ltclass.dtx, which is attributed to Frank Mittelbach, Chris Rowley, Alan Jeffrey, and David Carlisle. I merely made a few small changes (indicated below by bracketed blocks of code and comments) to make the filecontents environment more convenient to use.

1 ⟨∗package⟩

\filec@ntents   Except where indicated, the source—including comments—to the \filec@ntents macro was taken verbatim from ltclass.dtx.

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 %
```

In the original code a pre-existing file would not be overwritten. In the new version the file existence check is used solely to decide whether to output "`Writing file` `⟨filename⟩'`" or "`Overwriting file` `⟨filename⟩'`". Control flow then always falls through to what used to be the `\ifeof` case (file does not exist; open it), never the `\else` case (file already exists; do nothing).

```
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   \closein\@inputcheck %

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

```
20   \if@tempswa%
21     \immediate\write\reserved@c{%
22       \@percentchar\@percentchar\space%
23           \expandafter\@gobble\string\LaTeX2e file `#1'^^J%
24       \@percentchar\@percentchar\space  generated by the %
25           `\@currenvir' \expandafter\@gobblefour\string\newenvironment^^J%
26       \@percentchar\@percentchar\space from source `\jobname' on %
27           \number\year/\two@digits\month/\two@digits\day.^^J%
28       \@percentchar\@percentchar}%
29   \fi%
30   \let\do\@makeother\dospecials%
```

The inputenc packages might have marked some of the upper 128 character codes "active" (category code 13). That confuses filecontents. Hence, we locally mark each of the upper 128 character codes as "letter" (category code 11) so that they can be written correctly to a file.

```
31   \count0=128\relax %
32   \loop %
33     \catcode\count0=11\relax %
34     \advance\count0 by 1\relax %
35     \ifnum\count0<256 %
36   \repeat %
```

```
37    \edef\E{\@backslashchar end\string{\@currenvir\string}}%
38    \edef\reserved@b{%
39      \def\noexpand\reserved@b%
40          ####1\E####2\E####3\relax}%
41    \reserved@b{%
42      \ifx\relax##3\relax%
```

There was no `\end{filecontents}`

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

There was a `\end{filecontents}`, so stop this time.

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

Text before the `\end`, write it with a warning.

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

Text after the `\end`, ignore it with a warning.

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

59    \catcode`\^^L\active%
60    \let\L\@undefined%
61    \def^^L{\expandafter\ifx\csname L\endcsname\relax\fi ^^J^^J}%
62    \catcode`\^^I\active%
63    \let\I\@undefined%
64    \def^^I{\expandafter\ifx\csname I\endcsname\relax\fi\space}%
65    \catcode`\^^M\active%
66    \edef^^M##1^^M{%
67      \noexpand\reserved@b##1\E\E\relax}}%
68 \endgroup%
```

\fc@no@preamblecmds LaTeX 2_ε declares `\filecontents`, `\filecontents*`, and all of the related helper macros as `\@onlypreamble`, meaning they become invalid after the `\begin{document}`. The following code re-enables their usage anywhere in the document. It was taken from the pkgindoc package (which is generated from ltclass.dtx), but modified to re-enable only the commands needed by filecontents, not all of the class and package option-processing commands, as well.

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

```
70    \gdef\@preamblecmds{#1#3}}
71 \expandafter\fc@no@preamblecmds\@preamblecmds\relax

72 ⟨/package⟩
```

Change History

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.

5