The stampinclude package

Heiko Oberdiek*

2016/05/16 v1.1

Abstract

The package replaces `\includeonly` and selects the files for `\include` by inspecting the time stamp of the `.aux` file. The file is selected for inclusion if the `.aux` file does not yet exist or is older than the corresponding `.tex` file.

Contents

1 Documentation

1.1 Introduction

LaTeX provides two commands `\include` and `\includeonly` that helps in organizing large projects. Example for a master file:

*Please report any issues at https://github.com/ho-tex/oberdiek/issues

```
\documentclass{book}
  % \includeonly{}
\begin{document}
 \include{fileA}
 \include{fileB}
 \include{fileC}
\end{document}
```

All files are read and compiled if `\includeonly` is not executed. Otherwise you can give `\includeonly` a list of files in the preamble, e.g.:

```
\includeonly{fileA,fileC}
```

Now only files `fileA.tex` and `fileC.tex` are read and compiled.

If you change file `fileB.tex` and want to see only this file, then you must change the line with `\includeonly` to

```
\includeonly{fileB}
```

It is tedious to do this again and again, if different files are changed.

Package askinclude [1] offers a solution for this problem. It interactively asks for the files to be included and saves the user from editing the master file.

This package stampinclude goes another way. LaTeX reads and writes a separate `.aux` file for each file that is included by `\include`. There LaTeX remembers counter values. Changed `.tex` files can therefore be detected by comparing the file date stamp of the `.tex` file with the date stamp of its `.aux` file. Since version 1.30.0 pdfTeX provides `\pdffilemoddate` that reads the file date stamp. Thus this package uses this command and redefines `\include` to include the files that do not have `.aux` files yet or that are newer than its `.aux` file. `\includeonly` is ignored.

1.2 Usage

The package is loaded as normal LaTeX package without options:

```
\usepackage{stampinclude}
```

Alternatively the package may be loaded on the command line (Example for shell 'bash'):

```
latex '\AtBeginDocument{\usepackage{stampinclude}}\input{master}'
```

Without `\AtBeginDocument` (and `\RequirePackage` instead of `\usepackage`) TeX would name the document `stampinclude.dvi` instead of `master.dvi`.

1.3 Limitations

1.3.1 Other file dependencies

A file that is included by `\include` may input or reference other files:

- other TeX files using `\input`,
- graphics files (`\includegraphics`),
- listings of external files,
- ...

Updates of those files are not detected by this package. It limits the date stamp comparison of an `.aux` file to its `.tex` file.

1.3.2 `\include` dependencies

In the example, given in the introduction 1.1, three files `fileA`, `fileB`, and `fileC` are included in this order. Now file `fileA` is changed by adding four pages, `fileB` remains untouched, and `fileC` is also updated. Then the package only selects `fileA` and `fileC` for inclusion. File `fileB` is not included. But LaTeX has stored the counter values that are active at the end of `fileB` in `fileB.aux` in one of the previous runs when `fileB` was included. However the later addition of four pages in `fileA` was not known at that time. Therefore `fileB.aux` is out of date and the inclusion of file `fileC` starts with wrong counter values (especially the page counter).

1.3.3 Summary

This package stampinclude and the `\include` feature helps in accelerating the LaTeX compilation. But it is not intended for generating the final version. For the final version of the document it is better to include *all* files to get all counter values right. Then this package and any `\includeonly` lines should be commented out:

```
% \usepackage{stampinclude}
% \includeonly{...}
```

1.4 Requirements

- pdfTeX v1.30.0 (because of `\pdffilemoddate` and `\pdfstrcmp`), both modes for DVI and PDF are supported.
- Alternatively LuaTeX may be used. It lacks `\pdffilemoddate` and `\pdfstrcmp`. But its services are provided by package pdftexcmds [2] that is automatically loaded.

2 Implementation

```
1 ⟨*package⟩
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{stampinclude}
4   [2016/05/16 v1.1 Include files based on time stamps (HO)]%
5 \RequirePackage{pdftexcmds}[2007/12/12]%
6 \begingroup
7   \chardef\x=1 %
8   \expandafter\ifx\csname pdf@filemoddate\endcsname\relax
9     \chardef\x=0 %
10  \fi
11  \expandafter\ifx\csname pdf@strcmp\endcsname\relax
12    \chardef\x=0 %
13  \fi
14 \expandafter\endgroup\ifcase\x
15  \PackageWarningNoLine{stampinclude}{%
16    \string\pdffilemoddate\space or %
17    \string\pdfstrcmp\space are not found,\MessageBreak
18    that are provided by pdfTeX >= 1.30.0.\MessageBreak
19    Also LuaTeX is not detected.\MessageBreak
20    Therefore package loading is aborted%
21  }%
22  \expandafter\endinput
23 \fi
```

`\SInc@org@include`

```
24 \let\SInc@org@include\@include
```

\@include

```
\def\@include#1 {%
  \IfFileExists{#1.aux}{%
    \ifnum\pdf@strcmp{\pdf@filemoddate{#1.aux}}%
                     {\pdf@filemoddate{#1.tex}}<0 %
      \ifx\@partlist\@empty
        \gdef\@partlist{{#1}}%
      \else
        \g@addto@macro\@partlist{,{#1}}%
      \fi
    \fi
  }{%
    \ifx\@partlist\@empty
      \gdef\@partlist{{#1}}%
    \else
      \g@addto@macro\@partlist{,{#1}}%
    \fi
  }%
  \SInc@org@include{#1} \relax
}
```

\includeonly Macro \includeonly is ignored.

```
\renewcommand*{\includeonly}[1]{%
  \PackageInfo{stampinclude}{%
    Ignoring \string\includeonly
  }%
}
```

Simulate \includeonly.

```
\@partswtrue
\gdef\@partlist{}
```

Print included files at end of document.

```
\AtEndDocument{%
  \begingroup
    \expandafter\let\expandafter\@partlist\expandafter\@empty
    \expandafter\@for\expandafter\reserved@a
    \expandafter:\expandafter=\@partlist\do{%
      \ifx\@partlist\@empty
        \edef\@partlist{\reserved@a}%
      \else
        \edef\@partlist{\@partlist, \reserved@a}%
      \fi
    }%
    \typeout{********************%
             ********************%
             ********************%
             ******************%
    }%
    \ifx\@partlist\@empty
      \typeout{[stampinclude] No included files.}%
    \else
      \typeout{[stampinclude] Included files:}%
      \typeout{\@partlist}%
    \fi
    \typeout{********************%
             ********************%
             ********************%
             ******************%
    }%
  \endgroup
}
```

⟨/package⟩

3 Installation

3.1 Download

Package. This package is available on CTAN[1]:

CTAN:macros/latex/contrib/oberdiek/stampinclude.dtx The source file.

CTAN:macros/latex/contrib/oberdiek/stampinclude.pdf Documentation.

Bundle. All the packages of the bundle 'oberdiek' are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

CTAN:install/macros/latex/contrib/oberdiek.tds.zip

TDS refers to the standard "A Directory Structure for TEX Files" (CTAN:pkg/tds). Directories with `texmf` in their name are usually organized this way.

3.2 Bundle installation

Unpacking. Unpack the `oberdiek.tds.zip` in the TDS tree (also known as `texmf` tree) of your choice. Example (linux):

```
unzip oberdiek.tds.zip -d ~/texmf
```

3.3 Package installation

Unpacking. The `.dtx` file is a self-extracting docstrip archive. The files are extracted by running the `.dtx` through plain TEX:

```
tex stampinclude.dtx
```

TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as `texmf` tree):

```
stampinclude.sty → tex/latex/oberdiek/stampinclude.sty
stampinclude.pdf → doc/latex/oberdiek/stampinclude.pdf
stampinclude.dtx → source/latex/oberdiek/stampinclude.dtx
```

If you have a `docstrip.cfg` that configures and enables docstrip's TDS installing feature, then some files can already be in the right place, see the documentation of docstrip.

3.4 Refresh file name databases

If your TEX distribution (TEX Live, MiKTEX, ...) relies on file name databases, you must refresh these. For example, TEX Live users run `texhash` or `mktexlsr`.

3.5 Some details for the interested

Unpacking with LATEX. The `.dtx` chooses its action depending on the format:

plain TEX: Run docstrip and extract the files.

LATEX: Generate the documentation.

If you insist on using LATEX for docstrip (really, docstrip does not need LATEX), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{stampinclude.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

[1] CTAN:pkg/stampinclude

Generating the documentation. You can use both the `.dtx` or the `.drv` to generate the documentation. The process can be configured by the configuration file `ltxdoc.cfg`. For instance, put this line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with pdfLaTeX:

```
pdflatex stampinclude.dtx
makeindex -s gind.ist stampinclude.idx
pdflatex stampinclude.dtx
makeindex -s gind.ist stampinclude.idx
pdflatex stampinclude.dtx
```

4 References

[1] Pablo A. Straub, Heiko Oberdiek: *The askinclude package*; 2007/10/23 v2.0; CTAN:pkg/askinclude.

[2] Heiko Oberdiek: *The pdftexcmds package*; 2007/12/12 v0.3; CTAN:pkg/pdftexcmds.

5 History

[2008/07/14 v1.0]

- First version.

[2016/05/16 v1.1]

- Documentation updates.

6 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; plain numbers refer to the code lines where the entry is used.