

# 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

<b>1</b>	<b>Documentation</b>	<b>1</b>
1.1	Introduction	1
1.2	Usage	2
1.3	Limitations	2
1.3.1	Other file dependencies	2
1.3.2	<code>\include</code> dependencies	3
1.3.3	Summary	3
1.4	Requirements	3
<b>2</b>	<b>Implementation</b>	<b>3</b>
<b>3</b>	<b>Installation</b>	<b>5</b>
3.1	Download	5
3.2	Bundle installation	5
3.3	Package installation	5
3.4	Refresh file name databases	6
3.5	Some details for the interested	6
<b>4</b>	<b>References</b>	<b>6</b>
<b>5</b>	<b>History</b>	<b>6</b>
	[2008/07/14 v1.0]	6
	[2016/05/16 v1.1]	6
<b>6</b>	<b>Index</b>	<b>7</b>

## 1 Documentation

### 1.1 Introduction

L<sup>A</sup>T<sub>E</sub>X 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 `\pdffilemdate` 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 ore 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

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

## 2 Implementation

```
1 \langle *package\rangle
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{stampinclude}
4 [2016/05/16 v1.1 Include files based on time stamps (H0)]%
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

```
25 \def\@include#1 {%
26   \IfFileExists{#1.aux}{%
27     \ifnum\pdf@strcmp{\pdf@filemoddate{#1.aux}}%
28       {\pdf@filemoddate{#1.tex}}<0 %
29     \ifx\@partlist\@empty
30       \gdef\@partlist{#1}%
31     \else
32       \g@addto@macro\@partlist{,#1}%
33     \fi
34   \fi
35 }{%
36   \ifx\@partlist\@empty
37     \gdef\@partlist{#1}%
38   \else
39     \g@addto@macro\@partlist{,#1}%
40   \fi
41 }%
42 \Sinc@org@include{#1} \relax
43 }
```

\includeonly Macro \includeonly is ignored.

```
44 \renewcommand*\includeonly[1]{%
45   \PackageInfo{stampinclude}{%
46     Ignoring \string\includeonly
47   }%
48 }
```

Simulate \includeonly.

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

Print included files at end of document.

51 \AtEndDocument{%
52   \begingroup
53     \expandafter\let\expandafter\@partlist\expandafter\@empty
54     \expandafter\@for\expandafter\reserved@a
55     \expandafter:\expandafter=\@partlist\do{%
56       \ifx\@partlist\@empty
57         \edef\@partlist{\reserved@a}%
58       \else
59         \edef\@partlist{\@partlist, \reserved@a}%
60       \fi
61     }%
62     \typeout{*****%
63             *****%
64             *****%
65             *****%
66     }%
67     \ifx\@partlist\@empty
68       \typeout{[stampinclude] No included files.}%
69     \else
70       \typeout{[stampinclude] Included files:}%
71       \typeout{\@partlist}%
72     \fi
```

```

73 \typeout{*****%
74          *****%
75          *****%
76          *****%
77 }%
78 \endgroup
79 }
80 </package>

```

## 3 Installation

### 3.1 Download

**Package.** This package is available on CTAN<sup>1</sup>:

[CTAN:macros/latex/contrib/oberdiek/stampinclud.dtx](#) The source file.

[CTAN:macros/latex/contrib/oberdiek/stampinclud.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 T<sub>E</sub>X 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 T<sub>E</sub>X:

```
tex stampinclud.dtx
```

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

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

---

<sup>1</sup>[CTAN:pkg/stampinclud](#)

### 3.4 Refresh file name databases

If your T<sub>E</sub>X distribution (T<sub>E</sub>X Live, MiK<sub>T</sub>E<sub>X</sub>, ...) relies on file name databases, you must refresh these. For example, T<sub>E</sub>X Live users run `texhash` or `mktextlsr`.

### 3.5 Some details for the interested

**Unpacking with L<sup>A</sup>T<sub>E</sub>X.** The `.dtx` chooses its action depending on the format:

**plain T<sub>E</sub>X:** Run `docstrip` and extract the files.

**L<sup>A</sup>T<sub>E</sub>X:** Generate the documentation.

If you insist on using L<sup>A</sup>T<sub>E</sub>X for `docstrip` (really, `docstrip` does not need L<sup>A</sup>T<sub>E</sub>X), 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.

**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 pdfL<sup>A</sup>T<sub>E</sub>X:

```
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.

Symbols	
<code>\@empty</code> . . . . .	29, 36, 53, 56, 67
<code>\@for</code> . . . . .	54
<code>\@include</code> . . . . .	24, <u>25</u>
<code>\@partlist</code> . . . . .	29, 30, 32, 36, 37, 39, 50, 53, 55, 56, 57, 59, 67, 71
<code>\@partswtrue</code> . . . . .	49
<b>A</b>	
<code>\AtEndDocument</code> . . . . .	51
<b>C</b>	
<code>\chardef</code> . . . . .	7, 9, 12
<code>\csname</code> . . . . .	8, 11
<b>D</b>	
<code>\do</code> . . . . .	55
<b>E</b>	
<code>\endcsname</code> . . . . .	8, 11
<code>\endinput</code> . . . . .	22
<b>G</b>	
<code>\g@addto@macro</code> . . . . .	32, 39
<code>\gdef</code> . . . . .	30, 37, 50
<b>I</b>	
<code>\ifcase</code> . . . . .	14
<code>\IfFileExists</code> . . . . .	26
<code>\ifnum</code> . . . . .	27
<code>\ifx</code> . . . . .	8, 11, 29, 36, 56, 67
<b>M</b>	
<code>\MessageBreak</code> . . . . .	17, 18, 19
<b>N</b>	
<code>\NeedsTeXFormat</code> . . . . .	2
<b>P</b>	
<code>\PackageInfo</code> . . . . .	45
<code>\PackageWarningNoLine</code> . . . . .	15
<code>\pdf@filemoddate</code> . . . . .	27, 28
<code>\pdf@strcmp</code> . . . . .	27
<code>\pdf@filemoddate</code> . . . . .	16
<code>\pdfstrcmp</code> . . . . .	17
<code>\ProvidesPackage</code> . . . . .	3
<b>R</b>	
<code>\renewcommand</code> . . . . .	44
<code>\RequirePackage</code> . . . . .	5
<code>\reserved@a</code> . . . . .	54, 57, 59
<b>S</b>	
<code>\Sinc@org@include</code> . . . . .	24, 42
<code>\space</code> . . . . .	16, 17
<b>T</b>	
<code>\typeout</code> . . . . .	62, 68, 70, 71, 73
<b>X</b>	
<code>\x</code> . . . . .	7, 9, 12, 14