

The stampinclude package

Heiko Oberdiek
<heiko.oberdiek at gmail.com>

2008/07/14 v1.0

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
1.1	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
2	Implementation	3
3	Installation	5
3.1	Download	5
3.2	Bundle installation	5
3.3	Package installation	5
3.4	Refresh file name databases	5
3.5	Some details for the interested	6
4	Catalogue	6
5	References	7
6	History	7
	[2008/07/14 v1.0]	7
7	Index	7

1 Documentation

1.1 Introduction

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

```
\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 `pdf \TeX` 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 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 <*package>
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{stampinclude}
4 [2008/07/14 v1.0 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`

```
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{}
51 \Print included files at end of document.
52 \AtEndDocument{%
53   \begingroup
54     \expandafter\let\expandafter\@partlist\expandafter\@empty
55     \expandafter\@for\expandafter\reserved@a
56     \expandafter:\expandafter=\@partlist\do{%
57       \ifx\@partlist\@empty
58         \edef\@partlist{\reserved@a}%
59       \else
60         \edef\@partlist{\@partlist, \reserved@a}%
61       \fi
62     }%
63     \typeout{*****%
64             *****%
65             *****%
66             *****%
67             }%
68     \ifx\@partlist\@empty
69       \typeout{[stampinclude] No included files.}%
70     \else
71       \typeout{[stampinclude] Included files:}%
72       \typeout{\@partlist}%
73     \fi
74     \typeout{*****%
75             *****%
76             *****%
77             *****%
78             }%
79   \endgroup
80 </package>
```

3 Installation

3.1 Download

Package. This package is available on CTAN¹:

[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 \TeX Files” ([CTAN:tds/tds.pdf](#)). 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
```

Script installation. Check the directory `TDS:scripts/oberdiek/` for scripts that need further installation steps. Package `attachfile2` comes with the Perl script `pdfatfi.pl` that should be installed in such a way that it can be called as `pdfatfi`. Example (linux):

```
chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/
```

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 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`.

3.4 Refresh file name databases

If your \TeX distribution (te \TeX , mi \TeX , ...) relies on file name databases, you must refresh these. For example, te \TeX users run `texhash` or `mktextlsr`.

¹<ftp://ftp.ctan.org/tex-archive/>

3.5 Some details for the interested

Attached source. The PDF documentation on CTAN also includes the `.dtx` source file. It can be extracted by AcrobatReader 6 or higher. Another option is `pdftk`, e.g. unpack the file into the current directory:

```
pdftk stampinclude.pdf unpack_files output .
```

Unpacking with L^AT_EX. The `.dtx` chooses its action depending on the format:

plain T_EX: Run `docstrip` and extract the files.

L^AT_EX: Generate the documentation.

If you insist on using L^AT_EX for `docstrip` (really, `docstrip` does not need L^AT_EX), 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 `pdfLATEX`:

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

4 Catalogue

The following XML file can be used as source for the [T_EX Catalogue](#). The elements `caption` and `description` are imported from the original XML file from the Catalogue. The name of the XML file in the Catalogue is `stampinclude.xml`.

```
81 (*catalogue)
82 <?xml version='1.0' encoding='us-ascii'?>
83 <!DOCTYPE entry SYSTEM 'catalogue.dtd'>
84 <entry datestamp='$Date$' modifier='$Author$' id='stampinclude'>
85   <name>stampinclude</name>
86   <caption>Inclusion based on .aux file date stamps.</caption>
87   <authorref id='auth:oberdiek' />
88   <copyright owner='Heiko Oberdiek' year='2008' />
89   <license type='lppl1.3' />
90   <version number='1.0' />
91   <description>
92     This package replaces <tt>\includeonly</tt> and selects the files for
93     <tt>\include</tt> by inspecting the timestamp of the <tt>.aux</tt> file.
94     The file is selected for inclusion if the <tt>.aux</tt> file does
95     not yet exist or is older than the corresponding <tt>.tex</tt> file.
96     <p />
97     The package is part of the <xref refid='oberdiek'>oberdiek</xref>
98     bundle.
99   </description>
100   <documentation details='Package documentation'
101     href='ctan:/macros/latex/contrib/oberdiek/stampinclude.pdf' />
```

```

102 <ctan file='true' path='/macros/latex/contrib/oberdiek/stampinclude.dtx'>
103 <miktex location='oberdiek'>
104 <texlive location='oberdiek'>
105 <install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip'>
106 </entry>
107 </catalogue>

```

5 References

- [1] Pablo A. Straub, Heiko Oberdiek: *The askinclude package*; 2007/10/23 v2.0; [CTAN:macros/latex/contrib/oberdiek/askinclude.pdf](#).
- [2] Heiko Oberdiek: *The pdftexcmds package*; 2007/12/12 v0.3; [CTAN:macros/latex/contrib/oberdiek/pdftexcmds.pdf](#).

6 History

[2008/07/14 v1.0]

- First version.

7 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>\includeonly</code> 44, 92
<code>\@empty</code>	29, 36, 53, 56, 67	
<code>\@for</code>	54	M
<code>\@include</code>	24, 25	<code>\MessageBreak</code> 17, 18, 19
<code>\@partlist</code>	29, 30, 32, 36,	N
	37, 39, 50, 53, 55, 56, 57, 59, 67, 71	<code>\NeedsTeXFormat</code> 2
<code>\@partswtrue</code>	49	
	A	P
<code>\AtEndDocument</code>	51	<code>\PackageInfo</code> 45
	C	<code>\PackageWarningNoLine</code> 15
<code>\chardef</code>	7, 9, 12	<code>\pdf@filemoddate</code> 27, 28
<code>\csname</code>	8, 11	<code>\pdf@stricmp</code> 27
	D	<code>\pdf@filemoddate</code> 16
<code>\do</code>	55	<code>\pdf@stricmp</code> 17
	E	<code>\ProvidesPackage</code> 3
<code>\endcsname</code>	8, 11	R
<code>\endinput</code>	22	<code>\renewcommand</code> 44
	G	<code>\RequirePackage</code> 5
<code>\g@addto@macro</code>	32, 39	<code>\reserved@a</code> 54, 57, 59
<code>\gdef</code>	30, 37, 50	S
	I	<code>\Sinc@org@include</code> 24, 42
<code>\ifcase</code>	14	<code>\space</code> 16, 17
<code>\ifFileExists</code>	26	T
<code>\ifnum</code>	27	<code>\typeout</code> 62, 68, 70, 71, 73
<code>\ifx</code>	8, 11, 29, 36, 56, 67	X
<code>\include</code>	93	<code>\x</code> 7, 9, 12, 14