

# The auxhook package

Heiko Oberdiek  
<heiko.oberdiek at gmail.com>

2011/03/04 v1.3

## Abstract

Package `auxhook` provides hooks for adding stuff at the begin of `.aux` files.

## Contents

<b>1</b>	<b>User interface</b>	<b>1</b>
<b>2</b>	<b>Implementation</b>	<b>2</b>
2.1	Identification . . . . .	2
2.2	Hook setup . . . . .	2
2.3	User macros . . . . .	3
2.4	Patches . . . . .	3
2.4.1	<code>\document</code> . . . . .	3
2.4.2	<code>\@include</code> . . . . .	4
<b>3</b>	<b>Installation</b>	<b>4</b>
3.1	Download . . . . .	4
3.2	Bundle installation . . . . .	5
3.3	Package installation . . . . .	5
3.4	Refresh file name databases . . . . .	5
3.5	Some details for the interested . . . . .	5
<b>4</b>	<b>References</b>	<b>6</b>
<b>5</b>	<b>History</b>	<b>6</b>
	[2006/05/31 v1.0] . . . . .	6
	[2007/04/06 v1.1] . . . . .	6
	[2009/12/14 v1.2] . . . . .	6
	[2011/03/04 v1.3] . . . . .	6
<b>6</b>	<b>Index</b>	<b>6</b>

## 1 User interface

There are two kinds of `.aux` files, the main `.aux` file and the `.aux` file that belongs to an included file, specified by `\include`.

Some packages write macros in the auxiliary files. If the user stops using the package, these macros will usually cause error messages because of unknown commands. Prominent example is package `babel`'s `\select@language`.

But such a package could be written more cooperative. It can also provide a definition in the auxiliary file (`\providecommand`) that silently disables the macros of the package if the package is no longer in use.

In case of the main auxiliary file, `\AtBeginDocument` can be used for this purpose. Especially if several packages are involved, the order cannot be controlled

always (e.g., see package `hypdestopt` that hooks into `hyperref`'s macros). And there isn't any hook for the auxiliary files of the `\include` feature.

Thus this package patches L<sup>A</sup>T<sub>E</sub>X's macros `\document` and `\@include` to add the hooks where the auxiliary files are opened and the first line with `\relax` is written.

The patching can fail, if these macros are redefined by some other package. If the other package still uses the original definition, then load package `auxhook` earlier. (With `\RequirePackage` the package also can be loaded before the class). If the redefinition doesn't use the original meaning, then you can try to load package `auxhook` afterwards, but you need luck that the patch succeeds.

The hooks are macros:

**`\@beginmainauxhook`:** Start of the main auxiliary file. The hook is called after the first line with `\relax` is written.

**`\@beginpartauxhook`:** The same for the auxiliary files that belongs to the files that are included by `\include`.

If you want to add something to these hooks, you can use `\g@addto@macro` from L<sup>A</sup>T<sub>E</sub>X's kernel. But the package provides macros to add code that adds a line to the auxiliary file:

```
\AddLineBeginMainAux{\line}
\AddLineBeginPartAux{\line}
\AddLineBeginAux{\line}
```

The `\line` is added at the begin of the main auxiliary file by `\AddLineBeginMainAux` and at the begin of the auxiliary files of included files by `\AddLineBeginPartAux`. `\AddLineBeginAux` writes in both kinds of auxiliary files.

Examples, see packages `hypdestopt` ([1]) and `zref` ([3]).

## 2 Implementation

### 2.1 Identification

```
1 (*package)
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{auxhook}%
4 [2011/03/04 v1.3 Hooks for auxiliary files (HO)]%
```

### 2.2 Hook setup

**`\@beginmainauxhook`** The hook for the main auxiliary file, initially empty.

```
5 \providecommand*\@beginmainauxhook{}
```

**`\@beginpartauxhook`** The hook for auxiliary files of included files, initially empty.

```
6 \providecommand*\@beginpartauxhook{}
```

```
7 \ifx\AtBeginDocument\@firstofone
8   \global\let\@beginmainauxhook\relax
9 \else
10  \g@addto@macro{\@beginmainauxhook}{%
11    \global\let\@beginmainauxhook\relax
12  }%
13 \fi
```

## 2.3 User macros

`\AddLineBeginMainAux`

```
14 \newcommand{\AddLineBeginMainAux}[1]{%
15   \ifx\@beginmainauxhook\relax
16     \if@filesw
17       \PackageInfo{auxhook}{%
18         \@backslashchar AddLineBeginMainAux comes a little late,%
19         \MessageBreak
20         because the main .aux file is already opened%
21       }%
22       \immediate\write\@mainaux{#1}%
23     \fi
24   \else
25     \g@addto@macro\@beginmainauxhook{%
26       \immediate\write\@mainaux{#1}%
27     }%
28   \fi
29 }
```

`\AtBeginPartAuxLine`

```
30 \newcommand{\AddLineBeginPartAux}[1]{%
31   \g@addto@macro\@beginpartauxhook{%
32     \immediate\write\@partaux{#1}%
33   }%
34 }
```

`\AddLineBeginAux`

```
35 \newcommand{\AddLineBeginAux}[1]{%
36   \AddLineBeginMainAux{#1}%
37   \AddLineBeginPartAux{#1}%
38 }
```

## 2.4 Patches

### 2.4.1 `\document`

```
39 \begingroup
40   \@ifundefined{beamer@origdocument}{%
41     \def\auxhook@document{\document}%
42   }{%
43     \def\auxhook@document{\beamer@origdocument}%
44   }%
45   \long\def\y#1\immediate\write\@mainaux#2#3\auxhook@nil{%
46     \toks@{%
47       #1\immediate\write\@mainaux{#2}%
48       \@beginmainauxhook
49       #3%
50     }%
51     \expandafter\xdef\auxhook@document{\the\toks@}%
52   \endgroup
53 }%
54 \long\def\x#1\immediate\write\@mainaux#2#3\auxhook@nil{%
55   \toks@{#3}%
56   \edef\x{\the\toks@}%
57   \ifx\x\empty
58     \PackageWarningNoLine{auxhook}{%
59       Cannot patch \expandafter\string\auxhook@document,%
60       \MessageBreak
61       using \string\AtBeginDocument\space instead%
62     }%
63   \endgroup
```

```

64     \AtBeginDocument{%
65         \if@files
66         \beginmainauxhook
67     \fi
68 }%
69 \else
70     \expandafter\expandafter\expandafter\y\auxhook@document
71     \auxhook@nil
72 \fi
73 }%
74 \expandafter\expandafter\expandafter\x\auxhook@document
75 \immediate\write\@mainaux{}\auxhook@nil

```

## 2.4.2 \@include

```

76 \begingroup
77 \long\def\y#1\immediate\write\@partaux#2#3\auxhook@nil#4{%
78     \endgroup
79     \def#4##1 {%
80         #1\immediate\write\@partaux{#2}%
81         \@beginpartauxhook
82         #3%
83     }%
84 }%
85 \long\def\x#1\immediate\write\@partaux#2#3\auxhook@nil#4{%
86     \toks@{#3}%
87     \edef\x{\the\toks@}%
88     \ifx\x\@empty
89         \PackageWarningNoLine{auxhook}{%
90             Cannot patch \string#4,\MessageBreak
91             patch dropped%
92         }%
93     \endgroup
94 \else
95     \expandafter\y#4{##1} \auxhook@nil#4%
96 \fi
97 }%
98 \ifundefined{ReFiCh@org@include}{%
99     \expandafter\x\@include{#1} %
100     \immediate\write\@partaux{}\auxhook@nil\@include
101 }{%
102     \expandafter\x\ReFiCh@org@include{#1} %
103     \immediate\write\@partaux{}\auxhook@nil\ReFiCh@org@include
104 }%
105 </package>

```

## 3 Installation

### 3.1 Download

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

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

[CTAN:macros/latex/contrib/oberdiek/auxhook.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](#)

---

<sup>1</sup><http://ftp.ctan.org/tex-archive/>

*TDS* refers to the standard “A Directory Structure for T<sub>E</sub>X 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 T<sub>E</sub>X:

```
tex auxhook.dtx
```

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

```
auxhook.sty → tex/latex/oberdiek/auxhook.sty
auxhook.pdf → doc/latex/oberdiek/auxhook.pdf
auxhook.dtx → source/latex/oberdiek/auxhook.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 T<sub>E</sub>X distribution (teT<sub>E</sub>X, miK<sub>T</sub><sub>E</sub>X, ...) relies on file name databases, you must refresh these. For example, teT<sub>E</sub>X users run `texhash` or `mktextlsr`.

## 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 auxhook.pdf unpack_files output .
```

**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{auxhook.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 auxhook.dtx
makeindex -s gind.ist auxhook.idx
pdflatex auxhook.dtx
makeindex -s gind.ist auxhook.idx
pdflatex auxhook.dtx
```

## 4 References

- [1] Heiko Oberdiek: *The hypdestopt package*; 2006/05/30 v1.0; [CTAN:macros/latex/contrib/oberdiek/hypdestopt.pdf](#).
- [2] Sebastian Rahtz, Heiko Oberdiek: *The hyperref package*; 2006/08/16 v6.75c; [CTAN:macros/latex/contrib/hyperref/](#).
- [3] Heiko Oberdiek: *The zref package*; 2006/05/25 v1.2; [CTAN:macros/latex/contrib/oberdiek/zref.pdf](#).

## 5 History

[2006/05/31 v1.0]

- First version.

[2007/04/06 v1.1]

- Fix for class beamer.

[2009/12/14 v1.2]

- Support for package rerunfilecheck added (`\@include`).

[2011/03/04 v1.3]

- `\AddLineBeginMainAux` also supports write requests after the main `.aux` file is opened.

## 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>\@backslashchar</code> . . . . .	18
<code>\@beginmainauxhook</code> . . . . .	5, 8, 10, 11, 15, 25, 48, 66
<code>\@beginpartauxhook</code> . . . . .	6, 31, 81
<code>\@empty</code> . . . . .	57, 88
<code>\@firstofone</code> . . . . .	7
<code>\@ifundefined</code> . . . . .	40, 98
<code>\@include</code> . . . . .	99, 100
<code>\@mainaux</code> . . . . .	22, 26, 45, 47, 54, 75
<code>\@partaux</code> . . . . .	32, 77, 80, 85, 100, 103
<b>A</b>	
<code>\AddLineBeginAux</code> . . . . .	35
<code>\AddLineBeginMainAux</code> . . . . .	2, 14, 36
<code>\AddLineBeginPartAux</code> . . . . .	30, 37

<code>\AtBeginDocument</code> .....	7, 61, 64	<b>P</b>	
<code>\AtBeginPartAuxLine</code> .....	<u>30</u>	<code>\PackageInfo</code> .....	17
<code>\auxhook@document</code> .....	41, 43, 51, 59, 70, 74	<code>\PackageWarningNoLine</code> .....	58, 89
<code>\auxhook@nil</code> .....	45,	<code>\providecommand</code> .....	5, 6
	54, 71, 75, 77, 85, 95, 100, 103	<code>\ProvidesPackage</code> .....	3
<b>B</b>		<b>R</b>	
<code>\beamer@origdocument</code> .....	43	<code>\ReFiCh@org@include</code> .....	102, 103
<b>D</b>		<b>S</b>	
<code>\document</code> .....	41	<code>\space</code> .....	61
<b>G</b>		<b>T</b>	
<code>\g@addto@macro</code> .....	10, 25, 31	<code>\the</code> .....	51, 56, 87
<b>I</b>		<code>\toks@</code> .....	46, 51, 55, 56, 86, 87
<code>\if@files</code> .....	16, 65	<b>W</b>	
<code>\ifx</code> .....	7, 15, 57, 88	<code>\write</code> .....	22, 26, 32,
<code>\immediate</code> .....	22, 26, 32,		45, 47, 54, 75, 77, 80, 85, 100, 103
	45, 47, 54, 75, 77, 80, 85, 100, 103		
<b>M</b>		<b>X</b>	
<code>\MessageBreak</code> .....	19, 60, 90	<code>\x</code> ...	54, 56, 57, 74, 85, 87, 88, 99, 102
<b>N</b>		<b>Y</b>	
<code>\NeedsTeXFormat</code> .....	2	<code>\y</code> .....	45, 70, 77, 95
<code>\newcommand</code> .....	14, 30, 35		