

# The hopatch package

Heiko Oberdiek\*

2020-08-01 v1.4

## Abstract

This packages provides a wrapper to various package hooks provided by other packages or classes, but does not define own hooks.

## Contents

<b>1</b>	<b>Documentation for version 1.4</b>	<b>2</b>
<b>2</b>	<b>Documentation for version 1.3</b>	<b>2</b>
<b>3</b>	<b>Implementation</b>	<b>3</b>
3.1	v.1.4: New implementation using the LaTeX kernel hooks . . . . .	3
3.2	Catcodes and package identification . . . . .	4
3.3	Resources . . . . .	4
3.4	Package patching . . . . .	5
<b>4</b>	<b>Installation</b>	<b>7</b>
4.1	Download . . . . .	7
4.2	Package installation . . . . .	7
4.3	Refresh file name databases . . . . .	7
4.4	Some details for the interested . . . . .	7
<b>5</b>	<b>References</b>	<b>8</b>
<b>6</b>	<b>History</b>	<b>8</b>
[2011/01/30 v1.0]	. . . . .	8
[2011/06/24 v1.1]	. . . . .	8
[2012/05/28 v1.2]	. . . . .	8
[2016/05/16 v1.3]	. . . . .	8
[2020-08-01 v1.4]	. . . . .	8
<b>7</b>	<b>Index</b>	<b>8</b>

---

\*Please report any issues at <https://github.com/ho-tex/hopatch/issues>

# 1 Documentation for version 1.4

Starting with this version

```
\hopatch@AfterPackage {⟨package⟩} {⟨patch code⟩}
```

will use the kernel `package/after/⟨package⟩` hook if a LaTeX format later than 2020/20/01 is detected. This can mean that the hook code is executed earlier than with version 1.3 (but always after the `⟨package⟩`). If `⟨package⟩` is already loaded, the `⟨patch code⟩` is executed immediately. If an older format is detected it will fall back to the code of version 1.3.

## 2 Documentation for version 1.3

Sometimes I want to add code right after a package has been loaded. Examples are bug fixes, adaptations, or added features as needed by package `hyperref`, for instance.

Unhappily L<sup>A</sup>T<sub>E</sub>X does not provide this kind of hook. `\AtEndOfPackage` can be used inside the package only, because L<sup>A</sup>T<sub>E</sub>X clears the hook right before it loads the package.

However, there are already many packages and classes that provide hooks that are executed after the package is loaded, see table 1.

Package `hopatch` can be used without the packages of table 1. But for an early executing right after a package is loaded, one of the following class or packages should be loaded before using `\hopatch@AfterPackage`:

- package `filehook`
- package `scrfile`
- class `memoir`

Therefore I skip writing a new package for hooking into L<sup>A</sup>T<sub>E</sub>X's package management and use this package to provide a wrapper to patch a package after it is loaded.

```
\hopatch@AfterPackage {⟨package⟩} {⟨patch code⟩}
```

If the package is already loaded, the `⟨patch code⟩` is executed immediately. Otherwise the `⟨patch code⟩` is stored in a command and tried at later locations until the package is available.

The patch is tried in the following order:

1. If the package is already loaded, the patch is applied immediately. Further locations are not tried.

Table 1: After package hooking

Macro	Provider
<code>\AfterPackage</code>	package <code>scrfile</code> [5]
<code>\AtEndOfPackageFile</code>	package <code>filehook</code> [2]
<code>\AtEndPackage</code>	class <code>memoir</code> [4]

Table 2: After begin document hooking

Macro	Provider
<code>\AtBeginDocument</code>	L <sup>A</sup> T <sub>E</sub> X's kernel
<code>\AtEndPreamble</code>	package etoolbox [1]
<code>\AfterEndPreamble</code>	package etoolbox

2. `\AtEndPackage`, provided by class memoir [4], and `\AfterPackage`, provided by package scrfile [5], are called right after the package file is input before the hook of L<sup>A</sup>T<sub>E</sub>X's `\AtEndOfPackage`.
3. `\AtEndOfPackageFile`, provided by package filehook [2], is called after the package is loaded and after the hook of L<sup>A</sup>T<sub>E</sub>X's `\AtEndOfPackage`.
4. `\AtEndPreamble`, provided by package etoolbox [1], is called at the beginning of `\begin{document}` before the hook of L<sup>A</sup>T<sub>E</sub>X's `\AtBeginDocument`.
5. `\AtBeginDocument`, provided by L<sup>A</sup>T<sub>E</sub>X.
6. `\AfterEndDocument`, provided by package etoolbox [1], is called at the very end of `\begin{document}`. Preamble commands are already forbidden there.

Because of the various locations the patch code is restricted to limitations:

- Preamble commands, see L<sup>A</sup>T<sub>E</sub>X's `\@onlypreamble` throw an error if used after `\begin{document}`. This is already the case for `\AfterEndDocument`. Therefore preamble commands are forbidden in the patching code. There are four exceptions `\@ifpackageloaded`, `\@ifclassloaded`, `\@ifpackagelater` and `\@ifclasslater`. They are redefined during `\AfterEndDocument` using the counterparts of package ltxcmds [3].
- `\AfterPackage` of package scrfile and `\AtEndPackage` of class memoir call the hook before L<sup>A</sup>T<sub>E</sub>X's `\AtEndOfPackage`.

### 3 Implementation

1 `\package`

#### 3.1 v.1.4: New implementation using the LaTeX kernel hooks

```

2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{hopatch}%
4   [2020-08-01 v1.4 Wrapper for package hooks (HO)]
5 \providecommand\IfFormatAtLeastTF{\@ifl@t@r\fmtversion}
6 \IfFormatAtLeastTF{2020/10/01}{-}{\input{hopatch-2016-05-16.sty}}
7 \IfFormatAtLeastTF{2020/10/01}{-}{\endinput}
8
9 \newcommand\hopatch@AfterPackage[1]{%
10  \@ifpackageloaded{#1}{%
11    \@firstofone
12  }{%
13    \AddToHook{package/after/#1}%
14  }}
15

```

```

16 </package>
17 <*packagefrozen>

```

### 3.2 Catcodes and package identification

```

18 \begingroup\catcode61\catcode48\catcode32=10\relax%
19 \catcode13=5 % ^~M
20 \endlinechar=13 %
21 \catcode123=1 % {
22 \catcode125=2 % }
23 \catcode64=11 % @
24 \def\x{\endgroup
25 \expandafter\edef\csname H0patch@AtEnd\endcsname{%
26 \endlinechar=\the\endlinechar\relax
27 \catcode13=\the\catcode13\relax
28 \catcode32=\the\catcode32\relax
29 \catcode35=\the\catcode35\relax
30 \catcode61=\the\catcode61\relax
31 \catcode64=\the\catcode64\relax
32 \catcode123=\the\catcode123\relax
33 \catcode125=\the\catcode125\relax
34 }%
35 }%
36 \x\catcode61\catcode48\catcode32=10\relax%
37 \catcode13=5 % ^~M
38 \endlinechar=13 %
39 \catcode35=6 % #
40 \catcode64=11 % @
41 \catcode123=1 % {
42 \catcode125=2 % }
43 \def\TMP@EnsureCode#1#2{%
44 \edef\H0patch@AtEnd{%
45 \H0patch@AtEnd
46 \catcode#1=\the\catcode#1\relax
47 }%
48 \catcode#1=#2\relax
49 }
50 \TMP@EnsureCode{40}{12}% (
51 \TMP@EnsureCode{41}{12}% )
52 \TMP@EnsureCode{43}{12}% +
53 \TMP@EnsureCode{46}{12}% .
54 \TMP@EnsureCode{47}{12}% /
55 \TMP@EnsureCode{91}{12}% [
56 \TMP@EnsureCode{93}{12}% ]
57 \edef\H0patch@AtEnd{\H0patch@AtEnd\noexpand\endinput}

Package identification.
58 \NeedsTeXFormat{LaTeX2e}
59 \ProvidesPackage{hopatch}%
60 [2020-08-01 v1.4 Wrapper for package hooks / legacy code (v1.3) (H0)]

```

### 3.3 Resources

```

61 \begingroup\expandafter\expandafter\expandafter\endgroup
62 \expandafter\ifx\csname RequirePackage\endcsname\relax
63 \def\TMP@RequirePackage#1[#2]{%
64 \begingroup\expandafter\expandafter\expandafter\endgroup
65 \expandafter\ifx\csname ver@#1.sty\endcsname\relax
66 \input #1.sty\relax
67 \fi

```

```

68 }%
69 \TMP@RequirePackage{ltxcmds}[2010/12/12]%
70 \else
71 \RequirePackage{ltxcmds}[2010/12/12]%
72 \fi

```

\H0patch@counter

```
73 \def\H0patch@counter{0}%
```

\H0patch@StepCounter

```

74 \ltx@ifundefined{numexpr}{%
75 \def\H0patch@StepCounter{%
76 \begingroup
77 \count@\H0patch@counter\relax
78 \advance\count@\ltx@one\relax
79 \edef\x{\endgroup
80 \noexpand\def\noexpand\H0patch@counter{\the\count@}%
81 }%
82 \x
83 }%
84 }{%
85 \def\H0patch@StepCounter{%
86 \edef\H0patch@counter{%
87 \the\numexpr\H0patch@counter+\ltx@one\relax
88 }%
89 }%
90 }

```

\H0patch@list

```
91 \def\H0patch@list{}
```

\H0patch@Add

```

92 \def\H0patch@Add{%
93 \ltx@LocalAppendToMacro\H0patch@list
94 }

```

### 3.4 Package patching

\hopatch@AfterPackage

```

95 \def\hopatch@AfterPackage#1{%
96 \ltx@ifpackageloaded{#1}{%
97 \ltx@firstofone
98 }{%
99 \H0patch@AfterPackage{#1}%
100 }%
101 }

```

\H0patch@AfterPackage

```

102 \def\H0patch@AfterPackage#1{%
103 \edef\H0patch@temp{#1}%
104 \H0patch@StepCounter
105 \expandafter\H0patch@@AfterPackage
106 \csname H0patch@\H0patch@counter\expandafter\endcsname{%
107 \H0patch@temp
108 }%
109 }

```

\HOpatch@@AfterPackage

```
110 \def\HOpatch@@AfterPackage#1#2#3{%
111   \begingroup
112   \toks@{#3}%
113   \xdef\HOpatch@gtemp{%
114     \noexpand\ltx@ifpackageloaded{#2}{-%
115       \noexpand\let\noexpand#1\noexpand\relax
116       \the\toks@
117     }{}}%
118   }%
119 \endgroup
120 \let#1\HOpatch@gtemp
121 \HOpatch@Add#1%
122 \HOpatch@Try{AfterPackage}{#2}#1%
123 \HOpatch@Try{AtEndPackage}{#2}#1%
124 \HOpatch@Try{AtEndOfPackageFile}{#2}#1%
125 }
```

\HOpatch@Try

```
126 \def\HOpatch@Try#1#2#3{%
127   \ltx@ifundefined{#1}{}{-%
128     \csname #1\endcsname{#2}{#3}%
129   }%
130 }

131 \AtBeginDocument{\HOpatch@list}
132 \ltx@ifundefined{AtEndPreamble}{}{-%
133   \ltx@ifundefined{@endpreamblehook}{}{-%
134     \AtEndPreamble{\HOpatch@list}}%
135   }%
136 }
137 \ltx@ifundefined{AfterEndPreamble}{}{-%
138   \ltx@ifundefined{@afterendpreamblehook}{}{-%
139     \AfterEndPreamble{%
140       \let\HOpatch@OrgIfPackageLoaded\@ifpackageloaded
141       \let\HOpatch@OrgIfPackageLater\@ifpackagelater
142       \let\HOpatch@OrgIfClassLoaded\@ifclassloaded
143       \let\HOpatch@OrgIfClassLater\@ifclasslater
144       \let\@ifpackageloaded\ltx@ifpackageloaded
145       \let\@ifpackagelater\ltx@ifpackagelater
146       \let\@ifclassloaded\ltx@ifclassloaded
147       \let\@ifclasslater\ltx@ifclasslater
148       \HOpatch@list
149       \let\@ifpackageloaded\HOpatch@OrgIfPackageLoaded
150       \let\@ifpackagelater\HOpatch@OrgIfPackageLater
151       \let\@ifclassloaded\HOpatch@OrgIfClassLoaded
152       \let\@ifclasslater\HOpatch@OrgIfClassLater
153     }%
154   }%
155 }

156 \HOpatch@AtEnd%
157 </packagefrozen>
```

## 4 Installation

### 4.1 Download

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

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

[CTAN:macros/latex/contrib/hopatch/hopatch.pdf](#) Documentation.

### 4.2 Package installation

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

```
tex hopatch.dtx
```

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

```
hopatch.sty → tex/latex/hopatch/hopatch.sty
hopatch.pdf → doc/latex/hopatch/hopatch.pdf
hopatch.dtx → source/latex/hopatch/hopatch.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`.

### 4.3 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 `mktextlsr`.

### 4.4 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{hopatch.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:

---

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

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

## 5 References

- [1] Philipp Lehman: *The etoolbox Package* 2011-01-03. [CTAN:pkg/etoolbox](#)
- [2] Martin Scharrer: *The filehook Package*; 2011-01-09. [CTAN:pkg/filehook](#)
- [3] Heiko Oberdiek: *The ltxcmds Package*; 2010-12-12. [CTAN:pkg/ltxcmds](#)
- [4] Peter Wilson, Lars Madsen: *The Memoir Class for Configurable Typesetting, User Guide*; 2010. [CTAN:pkg/memoir](#)
- [5] Markus Kohm, Jens-Uwe Morawski: *The Guide KOMA-Script*; 2011-01-20. [CTAN:pkg/koma-script](#)

## 6 History

**[2011/01/30 v1.0]**

- First public version.

**[2011/06/24 v1.1]**

- Fix the use of `\AtEndPreamble` and `\AfterEndPreamble`. They are redefined by package `etoolbox` after their hooks are used and generate an error message then.

**[2012/05/28 v1.2]**

- Fix for use without  $\varepsilon$ -TeX (thanks Gordon Lee).

**[2016/05/16 v1.3]**

- Documentation updates.

**[2020-08-01 v1.4]**

- Starting with this version the package will use the hooks provided by the LaTeX kernel if the format is newer than 2020/10/01.

## 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.

<b>Symbols</b>	<code>\ifclassloaded</code> . . . . .	142, 146, 151
<code>\@firstofone</code> . . . . .	<code>\@ifl@t@r</code> . . . . .	11 5
<code>\@ifclasslater</code> . . . . .	<code>\@ifpackagelater</code> . . . . .	143, 147, 152 141, 145, 150



<code>\@ifpackageloaded</code> ..	10, 140, 144, 149		
		<b>A</b>	
<code>\AddToHook</code> .....	13		
<code>\advance</code> .....	78		
<code>\AfterEndPreamble</code> .....	139		
<code>\AtBeginDocument</code> .....	131		
<code>\AtEndPreamble</code> .....	134		
		<b>C</b>	
<code>\catcode</code> .....	18,		
	19, 21, 22, 23, 27, 28, 29, 30, 31,		
	32, 33, 36, 37, 39, 40, 41, 42, 46, 48		
<code>\count@</code> .....	77, 78, 80		
<code>\csname</code> .....	25, 62, 65, 106, 128		
		<b>E</b>	
<code>\endcsname</code> .....	25, 62, 65, 106, 128		
<code>\endinput</code> .....	7, 57		
<code>\endlinechar</code> .....	20, 26, 38		
		<b>F</b>	
<code>\fmtversion</code> .....	5		
		<b>H</b>	
<code>\HOPatch@@AfterPackage</code> ....	105, <u>110</u>		
<code>\HOPatch@Add</code> .....	92, <u>121</u>		
<code>\HOPatch@AfterPackage</code> .....	99, <u>102</u>		
<code>\hopatch@AfterPackage</code> ....	2, 2, 9, <u>95</u>		
<code>\HOPatch@AtEnd</code> .....	44, 45, 57, 156		
<code>\HOPatch@counter</code> .....	<u>73</u> , 77, 80, 86, 87, 106		
<code>\HOPatch@temp</code> .....	113, 120		
<code>\HOPatch@list</code> ..	<u>91</u> , 93, 131, 134, 148		
<code>\HOPatch@OrgIfClassLater</code> ...	143, 152		
<code>\HOPatch@OrgIfClassLoaded</code> ..	142, 151		
<code>\HOPatch@OrgIfPackageLater</code> .	141, 150		
<code>\HOPatch@OrgIfPackageLoaded</code>	140, 149		
<code>\HOPatch@StepCounter</code> .....	<u>74</u> , 104		
<code>\HOPatch@temp</code> .....	103, 107		
<code>\HOPatch@Try</code> .....	122, 123, 124, <u>126</u>		
		<b>I</b>	
<code>\IfFormatAtLeastTF</code> .....	5, 6, 7		
<code>\ifx</code> .....	62, 65		
<code>\input</code> .....	6, 66		
		<b>L</b>	
<code>\ltx@firstofone</code> .....	97		
<code>\ltx@ifclasslater</code> .....	147		
<code>\ltx@ifclassloaded</code> .....	146		
<code>\ltx@ifpackagelater</code> .....	145		
<code>\ltx@ifpackageloaded</code> ...	96, 114, 144		
<code>\ltx@ifundefined</code> .....			
	74, 127, 132, 133, 137, 138		
<code>\ltx@LocalAppendToMacro</code> .....	93		
<code>\ltx@one</code> .....	78, 87		
		<b>N</b>	
<code>\NeedsTeXFormat</code> .....	2, 58		
<code>\newcommand</code> .....	9		
<code>\numexpr</code> .....	87		
		<b>P</b>	
<code>\providecommand</code> .....	5		
<code>\ProvidesPackage</code> .....	3, 59		
		<b>R</b>	
<code>\RequirePackage</code> .....	71		
		<b>T</b>	
<code>\the</code> .....	26, 27, 28,		
	29, 30, 31, 32, 33, 46, 80, 87, 116		
<code>\TMP@EnsureCode</code> .....			
	43, 50, 51, 52, 53, 54, 55, 56		
<code>\TMP@RequirePackage</code> .....	63, 69		
<code>\toks@</code> .....	112, 116		
		<b>X</b>	
<code>\x</code> .....	24, 36, 79, 82		