

The hopatch package

Heiko Oberdiek*

2016/05/16 v1.3

Abstract

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

Contents

1	Documentation	1
2	Implementation	3
2.1	Catcodes and package identification	3
2.2	Resources	4
2.3	Package patching	4
3	Installation	6
3.1	Download	6
3.2	Bundle installation	6
3.3	Package installation	6
3.4	Refresh file name databases	6
3.5	Some details for the interested	7
4	References	7
5	History	7
[2011/01/30 v1.0]	7
[2011/06/24 v1.1]	7
[2012/05/28 v1.2]	7
[2016/05/16 v1.3]	8
6	Index	8

1 Documentation

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 `LATEX` does not provide this kind of hook. `\AtEndOfPackage` can be used inside the package only, because `LATEX` clears the hook right before it loads the package.

Table 1: After package hooking

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

Table 2: After begin document hooking

Macro	Provider
<code>\AtBeginDocument</code>	L ^A T _E X’s kernel
<code>\AtEndPreamble</code>	package etoolbox [1]
<code>\AfterEndPreamble</code>	package etoolbox

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^AT_EX’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.
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^AT_EX’s `\AtEndOfPackage`.
3. `\AtEndOfPackageFile`, provided by package filehook [2], is called after the package is loaded and after the hook of L^AT_EX’s `\AtEndOfPackage`.
4. `\AtEndPreamble`, provided by package etoolbox [1], is called at the beginning of `\begin{document}` before the hook of L^AT_EX’s `\AtBeginDocument`.
5. `\AtBeginDocument`, provided by L^AT_EX.
6. `\AfterEndDocument`, provided by package etoolbox [1], is called at the very end of `\begin{document}`. Preamble commands are already forbidden there.

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

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

- Preamble commands, see L^AT_EX'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^AT_EX's `\AtEndOfPackage`.

2 Implementation

```
1 \<{*package}
```

2.1 Catcodes and package identification

```
2 \begingroup\catcode61\catcode48\catcode32=10\relax%
3 \catcode13=5 % ^M
4 \endlinechar=13 %
5 \catcode123=1 % {
6 \catcode125=2 % }
7 \catcode64=11 % @
8 \def\x{\endgroup
9 \expandafter\edef\csname H0patch@AtEnd\endcsname{%
10 \endlinechar=\the\endlinechar\relax
11 \catcode13=\the\catcode13\relax
12 \catcode32=\the\catcode32\relax
13 \catcode35=\the\catcode35\relax
14 \catcode61=\the\catcode61\relax
15 \catcode64=\the\catcode64\relax
16 \catcode123=\the\catcode123\relax
17 \catcode125=\the\catcode125\relax
18 }%
19 }%
20 \x\catcode61\catcode48\catcode32=10\relax%
21 \catcode13=5 % ^M
22 \endlinechar=13 %
23 \catcode35=6 % #
24 \catcode64=11 % @
25 \catcode123=1 % {
26 \catcode125=2 % }
27 \def\TMP@EnsureCode#1#2{%
28 \edef\H0patch@AtEnd{%
29 \H0patch@AtEnd
30 \catcode#1=\the\catcode#1\relax
31 }%
32 \catcode#1=#2\relax
33 }
34 \TMP@EnsureCode{40}{12}% (
35 \TMP@EnsureCode{41}{12}% )
36 \TMP@EnsureCode{43}{12}% +
37 \TMP@EnsureCode{46}{12}% .
38 \TMP@EnsureCode{47}{12}% /
39 \TMP@EnsureCode{91}{12}% [
40 \TMP@EnsureCode{93}{12}% ]
41 \edef\H0patch@AtEnd{\H0patch@AtEnd\noexpand\endinput}
```

Package identification.

```
42 \NeedsTeXFormat{LaTeX2e}
43 \ProvidesPackage{hopatch}%
44 [2016/05/16 v1.3 Wrapper for package hooks (HO)]
```

2.2 Resources

```
45 \begingroup\expandafter\expandafter\expandafter\endgroup
46 \expandafter\ifx\csname RequirePackage\endcsname\relax
47   \def\TMP@RequirePackage#1[#2]{%
48     \begingroup\expandafter\expandafter\expandafter\endgroup
49     \expandafter\ifx\csname ver@#1.sty\endcsname\relax
50       \input #1.sty\relax
51     \fi
52   }%
53   \TMP@RequirePackage{ltxcmds}[2010/12/12]%
54 \else
55   \RequirePackage{ltxcmds}[2010/12/12]%
56 \fi
```

\HOpatch@counter

```
57 \def\HOpatch@counter{0}%
```

\HOpatch@StepCounter

```
58 \ltx@ifundefined{numexpr}{%
59   \def\HOpatch@StepCounter{%
60     \begingroup
61       \count@\HOpatch@counter\relax
62       \advance\count@\ltx@one\relax
63     \edef\x{\endgroup
64       \noexpand\def\noexpand\HOpatch@counter{\the\count@}%
65     }%
66     \x
67   }%
68 }{%
69   \def\HOpatch@StepCounter{%
70     \edef\HOpatch@counter{%
71       \the\numexpr\HOpatch@counter+\ltx@one\relax
72     }%
73   }%
74 }
```

\HOpatch@list

```
75 \def\HOpatch@list{}
```

\HOpatch@Add

```
76 \def\HOpatch@Add{%
77   \ltx@LocalAppendToMacro\HOpatch@list
78 }
```

2.3 Package patching

\hopatch@AfterPackage

```
79 \def\hopatch@AfterPackage#1{%
80   \ltx@ifpackageloaded{#1}{%
81     \ltx@firstofone
82   }{%
83     \HOpatch@AfterPackage{#1}%
84   }%
```

85 }

\HOpatch@AfterPackage

```
86 \def\HOpatch@AfterPackage#1{%
87   \edef\HOpatch@temp{#1}%
88   \HOpatch@StepCounter
89   \expandafter\HOpatch@@AfterPackage
90   \csname HOpatch@\HOpatch@counter\expandafter\endcsname{%
91     \HOpatch@temp
92   }%
93 }
```

\HOpatch@@AfterPackage

```
94 \def\HOpatch@@AfterPackage#1#2#3{%
95   \begingroup
96     \toks@{#3}%
97     \xdef\HOpatch@gtemp{%
98       \noexpand\ltx@ifpackageloaded{#2}{%
99         \noexpand\let\noexpand#1\noexpand\relax
100       \the\toks@
101     }{%}%
102   }%
103 \endgroup
104 \let#1\HOpatch@gtemp
105 \HOpatch@Add#1%
106 \HOpatch@Try{AfterPackage}{#2}#1%
107 \HOpatch@Try{AtEndPackage}{#2}#1%
108 \HOpatch@Try{AtEndOfPackageFile}{#2}#1%
109 }
```

\HOpatch@Try

```
110 \def\HOpatch@Try#1#2#3{%
111   \ltx@ifundefined{#1}{}{}%
112   \csname #1\endcsname{#2}{#3}%
113 }%
114 }

115 \AtBeginDocument{\HOpatch@list}
116 \ltx@ifundefined{AtEndPreamble}{}{}%
117 \ltx@ifundefined{@endpreamblehook}{}{%
118   \AtEndPreamble{\HOpatch@list}%
119 }%
120 }

121 \ltx@ifundefined{AfterEndPreamble}{}{%
122   \ltx@ifundefined{@afterendpreamblehook}{}{%
123     \AfterEndPreamble{%
124       \let\HOpatch@OrgIfPackageLoaded\ifpackageloaded
125       \let\HOpatch@OrgIfPackageLater\ifpackagelater
126       \let\HOpatch@OrgIfClassLoaded\ifclassloaded
127       \let\HOpatch@OrgIfClassLater\ifclasslater
128       \let\@ifpackageloaded\ltx@ifpackageloaded
129       \let\@ifpackagelater\ltx@ifpackagelater
130       \let\@ifclassloaded\ltx@ifclassloaded
131       \let\@ifclasslater\ltx@ifclasslater
132       \HOpatch@list
133       \let\@ifpackageloaded\HOpatch@OrgIfPackageLoaded
134       \let\@ifpackagelater\HOpatch@OrgIfPackageLater
135       \let\@ifclassloaded\HOpatch@OrgIfClassLoaded
```

```

136      \let\@ifclasslater\H0patch@0rgIfClassLater
137    }%
138  }%
139 }
140 \H0patch@AtEnd%
141 \</package>

```

3 Installation

3.1 Download

Package. This package is available on CTAN¹:

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

[CTAN:macros/latex/contrib/oberdiek/hopatch.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_EX 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_EX:

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

3.4 Refresh file name databases

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

¹[CTAN:pkg/hopatch](#)

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{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 `pdf \LaTeX` :

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

4 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](#)

5 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 ε - \TeX (thanks Gordon Lee).

[2016/05/16 v1.3]

- 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	
\@ifclasslater	127, 131, 136
\@ifclassloaded	126, 130, 135
\@ifpackagelater	125, 129, 134
\@ifpackageloaded	124, 128, 133
A	
\advance	62
\AfterEndPreamble	123
\AtBeginDocument	115
\AtEndPreamble	118
C	
\catcode	2, 3, 5, 6, 7, 11, 12, 13, 14, 15, 16, 17, 20, 21, 23, 24, 25, 26, 30, 32
\count@	61, 62, 64
\csname	9, 46, 49, 90, 112
E	
\endcsname	9, 46, 49, 90, 112
\endinput	41
\endlinechar	4, 10, 22
H	
\H0patch@@AfterPackage	89, <u>94</u>
\H0patch@Add	<u>76</u> , 105
\H0patch@AfterPackage	83, <u>86</u>
\hopatch@AfterPackage	2, <u>79</u>
\H0patch@AtEnd	28, 29, 41, 140
\H0patch@counter	<u>57</u> , 61, 64, 70, 71, 90
\H0patch@gtemp	97, 104
\H0patch@list ...	<u>75</u> , 77, 115, 118, 132
\H0patch@OrgIfClassLater ...	127, 136
\H0patch@OrgIfClassLoaded ..	126, 135
\H0patch@OrgIfPackageLater .	125, 134
\H0patch@OrgIfPackageLoaded	124, 133
\H0patch@StepCounter	<u>58</u> , 88
\H0patch@temp	87, 91
\H0patch@Try	106, 107, 108, <u>110</u>
I	
\ifx	46, 49
\input	50
L	
\ltx@firstofone	81
\ltx@ifclasslater	131
\ltx@ifclassloaded	130
\ltx@ifpackagelater	129
\ltx@ifpackageloaded	80, 98, 128
\ltx@ifundefined	58, 111, 116, 117, 121, 122
\ltx@LocalAppendToMacro	77
\ltx@one	62, 71
N	
\NeedsTeXFormat	42
\numexpr	71
P	
\ProvidesPackage	43
R	
\RequirePackage	55
T	
\the	10, 11, 12, 13, 14, 15, 16, 17, 30, 64, 71, 100
\TMP@EnsureCode	27, 34, 35, 36, 37, 38, 39, 40
\TMP@RequirePackage	47, 53
\toks@	96, 100
X	
\x	8, 20, 63, 66