

# The hopatch package

Heiko Oberdiek

<heiko.oberdiek at gmail.com>

2011/01/30 v1.0

## Abstract

This package 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</b>	<b>1</b>
<b>2</b>	<b>Implementation</b>	<b>3</b>
2.1	Catcodes and package identification . . . . .	3
2.2	Resources . . . . .	3
2.3	Package patching . . . . .	4
<b>3</b>	<b>Test</b>	<b>5</b>
3.1	Catcode checks for loading . . . . .	5
<b>4</b>	<b>Installation</b>	<b>8</b>
4.1	Download . . . . .	8
4.2	Bundle installation . . . . .	8
4.3	Package installation . . . . .	8
4.4	Refresh file name databases . . . . .	9
4.5	Some details for the interested . . . . .	9
<b>5</b>	<b>References</b>	<b>9</b>
<b>6</b>	<b>History</b>	<b>10</b>
	[2011/01/30 v1.0] . . . . .	10
<b>7</b>	<b>Index</b>	<b>10</b>

## 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  $\text{\LaTeX}$  does not provide this kind of hook. `\AtEndOfPackage` can be used inside the package only, because  $\text{\LaTeX}$  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`:

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

- 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.
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`, `\ifpackageafter` and `\ifclassafter`. 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`.

## 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 [2011/01/30 v1.0 Wrapper for package hooks (H0)]
```

### 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

\H0patch@counter

57 \def\H0patch@counter{0}%

\H0patch@StepCounter

58 \ltx@ifundefined{numexpr}{%
59   \def\H0patch@StepCounter{%
60     \begingroup
61       \count@\H0patch@counter\relax
62       \advance\count@\ltx@one\relax
63     \edef\x{\endgroup
64       \noexpand\def\noexpand\H0patch@counter{\the\count@}%
65     }%
66   }%
67 }{%
68   \def\H0patch@StepCounter{%
69     \edef\H0patch@counter{%
70       \the\numexpr\H0patch@counter+\ltx@one\relax
71     }%
72   }%
73 }

\H0patch@list

74 \def\H0patch@list{}

\H0patch@Add

75 \def\H0patch@Add{%
76   \ltx@LocalAppendToMacro\H0patch@list
77 }

```

## 2.3 Package patching

```

\hopatch@AfterPackage

78 \def\hopatch@AfterPackage#1{%
79   \ltx@ifpackageloaded{#1}{%
80     \ltx@firstofone
81   }{%
82     \H0patch@AfterPackage{#1}%
83   }%
84 }

\H0patch@AfterPackage

85 \def\H0patch@AfterPackage#1{%
86   \edef\H0patch@temp{#1}%
87   \H0patch@StepCounter
88   \expandafter\H0patch@@AfterPackage
89   \csname H0patch@\H0patch@counter\expandafter\endcsname{%
90     \H0patch@temp
91   }%
92 }

\H0patch@@AfterPackage

93 \def\H0patch@@AfterPackage#1#2#3{%
94   \begingroup
95   \toks@{#3}%
96   \xdef\H0patch@gtemp{%
97     \noexpand\ltx@ifpackageloaded{#2}{%
98       \noexpand\let\noexpand#1\noexpand\relax
99       \the\toks@
100    }{}%

```

```

101     }%
102 \endgroup
103 \let#1\H0patch@temp
104 \H0patch@Add#1%
105 \H0patch@Try{AfterPackage}{#2}#1%
106 \H0patch@Try{AtEndPackage}{#2}#1%
107 \H0patch@Try{AtEndOfPackageFile}{#2}#1%
108 }

\H0patch@Try
109 \def\H0patch@Try#1#2#3{%
110   \ltx@ifundefined{#1}{-}{%
111     \csname #1\endcsname{#2}{#3}%
112   }%
113 }

114 \AtBeginDocument{\H0patch@list}
115 \ltx@ifundefined{AtEndPreamble}{-}{%
116   \AtEndPreamble{\H0patch@list}%
117 }
118 \ltx@ifundefined{AfterEndPreamble}{-}{%
119   \AfterEndPreamble{%
120     \let\H0patch@OrgIfPackageLoaded\@ifpackageloaded
121     \let\H0patch@OrgIfPackageLater\@ifpackagelater
122     \let\H0patch@OrgIfClassLoaded\@ifclassloaded
123     \let\H0patch@OrgIfClassLater\@ifclasslater
124     \let\@ifpackageloaded\ltx@ifpackageloaded
125     \let\@ifpackagelater\ltx@ifpackagelater
126     \let\@ifclassloaded\ltx@ifclassloaded
127     \let\@ifclasslater\ltx@ifclasslater
128     \H0patch@list
129     \let\@ifpackageloaded\H0patch@OrgIfPackageLoaded
130     \let\@ifpackagelater\H0patch@OrgIfPackageLater
131     \let\@ifclassloaded\H0patch@OrgIfClassLoaded
132     \let\@ifclasslater\H0patch@OrgIfClassLater
133   }%
134 }

135 \H0patch@AtEnd%
136 </package>

```

### 3 Test

```

137 <test1>
138 \def\LoadCommand{\RequirePackage{hopatch}[2011/01/30]}
139 </test1>

```

#### 3.1 Catcode checks for loading

```

140 <test1>
141 \catcode'\{=1 %
142 \catcode'\}=2 %
143 \catcode'\#=6 %
144 \catcode'\@=11 %
145 \expandafter\ifx\csname count\endcsname\relax
146   \countdef\count@=255 %
147 \fi
148 \expandafter\ifx\csname @gobble\endcsname\relax
149   \long\def\@gobble#1{}%
150 \fi
151 \expandafter\ifx\csname @firstofone\endcsname\relax
152   \long\def\@firstofone#1{#1}%

```

```

153 \fi
154 \expandafter\ifx\csname loop\endcsname\relax
155   \expandafter\@firstofone
156 \else
157   \expandafter\@gobble
158 \fi
159 {%
160   \def\loop#1\repeat{%
161     \def\body{#1}%
162     \iterate
163   }%
164   \def\iterate{%
165     \body
166     \let\next\iterate
167   \else
168     \let\next\relax
169   \fi
170   \next
171 }%
172 \let\repeat=\fi
173 }%
174 \def\RestoreCatcodes{}
175 \count@=0 %
176 \loop
177   \edef\RestoreCatcodes{%
178     \RestoreCatcodes
179     \catcode\the\count@=\the\catcode\count@\relax
180   }%
181 \ifnum\count@<255 %
182   \advance\count@ 1 %
183 \repeat
184
185 \def\RangeCatcodeInvalid#1#2{%
186   \count@=#1\relax
187   \loop
188     \catcode\count@=15 %
189   \ifnum\count@<#2\relax
190     \advance\count@ 1 %
191   \repeat
192 }
193 \def\RangeCatcodeCheck#1#2#3{%
194   \count@=#1\relax
195   \loop
196     \ifnum#3=\catcode\count@
197   \else
198     \errmessage{%
199       Character \the\count@\space
200       with wrong catcode \the\catcode\count@\space
201       instead of \number#3%
202     }%
203   \fi
204   \ifnum\count@<#2\relax
205     \advance\count@ 1 %
206   \repeat
207 }
208 \def\space{ }
209 \expandafter\ifx\csname LoadCommand\endcsname\relax
210   \def\LoadCommand{\input hopatch.sty\relax}%
211 \fi
212 \def\Test{%
213   \RangeCatcodeInvalid{0}{47}%
214   \RangeCatcodeInvalid{58}{64}%

```

```

215 \RangeCatcodeInvalid{91}{96}%
216 \RangeCatcodeInvalid{123}{255}%
217 \catcode'\@=12 %
218 \catcode'\=0 %
219 \catcode'\%=14 %
220 \LoadCommand
221 \RangeCatcodeCheck{0}{36}{15}%
222 \RangeCatcodeCheck{37}{37}{14}%
223 \RangeCatcodeCheck{38}{47}{15}%
224 \RangeCatcodeCheck{48}{57}{12}%
225 \RangeCatcodeCheck{58}{63}{15}%
226 \RangeCatcodeCheck{64}{64}{12}%
227 \RangeCatcodeCheck{65}{90}{11}%
228 \RangeCatcodeCheck{91}{91}{15}%
229 \RangeCatcodeCheck{92}{92}{0}%
230 \RangeCatcodeCheck{93}{96}{15}%
231 \RangeCatcodeCheck{97}{122}{11}%
232 \RangeCatcodeCheck{123}{255}{15}%
233 \RestoreCatcodes
234 }
235 \Test
236 \csname @@end\endcsname
237 \end

238 </test1>

239 <*test2>
240 \NeedsTeXFormat{LaTeX2e}
241 \providecommand\variant{0}
242 \RequirePackage{filecontents}
243 \begin{filecontents}{foo.sty}
244 \ProvidesPackage{foo}
245 \def\msg#\{\immediate\write16}
246 \def\foo#1{%
247   \msg{\fooformat{#1}}%
248 }
249 \def\fooformat#1{[#1]}% hash-ok
250 \foo{* Executing foo at package loading}
251 \end{filecontents}
252
253 \ifnum\variant=1 %
254   \documentclass{memoir}%
255 \else
256   \documentclass{article}%
257 \fi
258
259 \ifcase\variant\relax
260 \or % 1
261 \or % 2
262   \usepackage{etoolbox}%
263 \or % 3
264   \usepackage{scrfile}%
265 \or % 4
266   \usepackage{filehook}%
267 \fi
268
269 \AtBeginDocument{\foo{* AtBeginDocument before hopatch}}
270 \usepackage{hopatch}
271 \AtBeginDocument{\foo{* AtBeginDocument after hopatch}}
272
273 \makeatletter
274 \hopatch@AfterPackage{foo}{%
275   \def\fooformat#1{<<#1>>}%
276 }

```

```

277 \makeatother
278
279 \AtBeginDocument{\foo{* AtBeginDocument before foo}}
280 \usepackage{foo}
281 \AtBeginDocument{\foo{* AtBeginDocument after foo}}
282
283 \foo{* Executing in preamble}
284
285 \begin{document}
286 \foo{* Executing in document}
287 \end{document}
288 \end{test2}

```

## 4 Installation

### 4.1 Download

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

[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  $\text{\TeX}$  Files” ([CTAN:tds/tds.pdf](#)). Directories with `texmf` in their name are usually organized this way.

### 4.2 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/

```

### 4.3 Package installation

**Unpacking.** The `.dtx` file is a self-extracting docstrip archive. The files are extracted by running the `.dtx` through plain  $\text{\TeX}$ :

```
tex hopatch.dtx
```

---

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



**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
test/hopatch-test1.tex → doc/latex/oberdiek/test/hopatch-test1.tex
test/hopatch-test2.tex → doc/latex/oberdiek/test/hopatch-test2.tex
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`.

## 4.4 Refresh file name databases

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

## 4.5 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 hopatch.pdf unpack_files output .
```

**Unpacking with  $\text{\LaTeX}$ .** The `.dtx` chooses its action depending on the format:

**plain  $\text{\TeX}$ :** Run `docstrip` and extract the files.

**$\text{\LaTeX}$ :** Generate the documentation.

If you insist on using  $\text{\LaTeX}$  for `docstrip` (really, `docstrip` does not need  $\text{\LaTeX}$ ), 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 $\text{\LaTeX}$` :

```
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:macros/latex/contrib/etoolbox/etoolbox.pdf](http://CTAN:macros/latex/contrib/etoolbox/etoolbox.pdf)
- [2] Martin Scharrer: *The filehook Package*; 2011-01-09.  
[CTAN:macros/latex/contrib/filehook/filehook.pdf](http://CTAN:macros/latex/contrib/filehook/filehook.pdf)

- [3] Heiko Oberdiek: *The ltxcmds Package*; 2010-12-12.  
[CTAN:macros/latex/contrib/oberdiek/ltxcmds.pdf](#)
- [4] Peter Wilson, Lars Madsen: *The Memoir Class for Configurable Typesetting, User Guide*; 2010. [CTAN:macros/latex/contrib/memoir/memman.pdf](#)
- [5] Markus Kohm, Jens-Uwe Morawski: *The Guide KOMA-Script*; 2011-01-20.  
[CTAN:macros/latex/contrib/koma-script/scrguien.pdf](#)

## 6 History

[2011/01/30 v1.0]

- First public 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			
\#	143	\endcsname	9, 46, 49, 89, 111, 145, 148, 151, 154, 209, 236
\%	219	\endinput	41
\@	144, 217	\endlinechar	4, 10, 22
\@firstofone	152, 155	\errmessage	198
\@gobble	149, 157	<b>F</b>	
\@ifclasslater	123, 127, 132	\foo	246, 250, 269, 271, 279, 281, 283, 286
\@ifclassloaded	122, 126, 131	\fooformat	247, 249, 275
\@ifpackagelater	121, 125, 130	<b>H</b>	
\@ifpackageloaded	120, 124, 129	\H0patch@@AfterPackage	88, 93
\@	218	\H0patch@Add	75, 104
\{	141	\H0patch@AfterPackage	82, 85
\}	142	\hopatch@AfterPackage	2, 78, 274
<b>A</b>		\H0patch@AtEnd	28, 29, 41, 135
\advance	62, 182, 190, 205	\H0patch@counter	57, 61, 64, 69, 70, 89
\AfterEndPreamble	119	\H0patch@gtemp	96, 103
\AtBeginDocument	114, 269, 271, 279, 281	\H0patch@list	74, 76, 114, 116, 128
\AtEndPreamble	116	\H0patch@OrgIfClassLater	123, 132
<b>B</b>		\H0patch@OrgIfClassLoaded	122, 131
\begin	243, 285	\H0patch@OrgIfPackageLater	121, 130
\body	161, 165	\H0patch@OrgIfPackageLoaded	120, 129
<b>C</b>		\H0patch@StepCounter	58, 87
\catcode	2, 3, 5, 6, 7, 11, 12, 13, 14, 15, 16, 17, 20, 21, 23, 24, 25, 26, 30, 32, 141, 142, 143, 144, 179, 188, 196, 200, 217, 218, 219	\H0patch@temp	86, 90
\count@	61, 62, 64, 146, 175, 179, 181, 182, 186, 188, 189, 190, 194, 196, 199, 200, 204, 205	\H0patch@Try	105, 106, 107, 109
\countdef	146	<b>I</b>	
\csname	9, 46, 49, 89, 111, 145, 148, 151, 154, 209, 236	\ifcase	259
<b>D</b>		\ifnum	181, 189, 196, 204, 253
\documentclass	254, 256	\ifx	46, 49, 145, 148, 151, 154, 209
<b>E</b>		\immediate	245
\end	237, 251, 287	\input	50, 210
		\iterate	162, 164, 166
		<b>L</b>	
		\LoadCommand	138, 210, 220
		\loop	160, 176, 187, 195
		\ltx@firstofone	80
		\ltx@ifclasslater	127

