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

---

*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 then 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 if 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 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.

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 scrlfile
- class memoir

Therefore I skip writing a new package for hooking into LaTeX'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**
`\AfterPackage`	package scrlfile [5]
`\AtEndOfPackageFile`	package filehook [2]
`\AtEndPackage`	class memoir [4]

Table 2: After begin document hooking

Macro	Provider
`\AtBeginDocument`	LaTeX's kernel
`\AtEndPreamble`	package etoolbox [1]
`\AfterEndPreamble`	package etoolbox

2. `\AtEndPackage`, provided by class memoir [4], and `\AfterPackage`, provided by package scrlfile [5], are called right after the package file is input before the hook of LaTeX's `\AtEndOfPackage`.
3. `\AtEndOfPackageFile`, provided by package filehook [2], is called after the package is loaded and after the hook of LaTeX's `\AtEndOfPackage`.
4. `\AtEndPreamble`, provided by package etoolbox [1], is called at the beginning of `\begin{document}` before the hook of LaTeX's `\AtBeginDocument`.
5. `\AtBeginDocument`, provided by LaTeX.
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 LaTeX'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 scrlfile and `\AtEndPackage` of class memoir call the hook before LaTeX's `\AtEndOfPackage`.

3 Implementation

```
⟨*package⟩
```

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

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{hopatch}%
  [2020-08-01 v1.4 Wrapper for package hooks (HO)]
```

```
\providecommand\IfFormatAtLeastTF{\@ifl@t@r\fmtversion}
\IfFormatAtLeastTF{2020/10/01}{}{\input{hopatch-2016-05-16.sty}}
\IfFormatAtLeastTF{2020/10/01}{}{\endinput}

\newcommand\hopatch@AfterPackage[1]{%
  \@ifpackageloaded{#1}{%
    \@firstofone
  }{%
    \AddToHook{package/after/#1}%
  }}

```

```
⟨/package⟩
```

```
⟨*packagefrozen⟩
```

3.2 Catcodes and package identification

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \catcode123=1 % {
  \catcode125=2 % }
  \catcode64=11 % @
  \def\x{\endgroup
    \expandafter\edef\csname HOpatch@AtEnd\endcsname{%
      \endlinechar=\the\endlinechar\relax
      \catcode13=\the\catcode13\relax
      \catcode32=\the\catcode32\relax
      \catcode35=\the\catcode35\relax
      \catcode61=\the\catcode61\relax
      \catcode64=\the\catcode64\relax
      \catcode123=\the\catcode123\relax
      \catcode125=\the\catcode125\relax
    }%
  }%
\x\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 % ^^M
\endlinechar=13 %
\catcode35=6 % #
\catcode64=11 % @
\catcode123=1 % {
\catcode125=2 % }
\def\TMP@EnsureCode#1#2{%
  \edef\HOpatch@AtEnd{%
    \HOpatch@AtEnd
    \catcode#1=\the\catcode#1\relax
  }%
  \catcode#1=#2\relax
}
\TMP@EnsureCode{40}{12}% (
\TMP@EnsureCode{41}{12}% )
\TMP@EnsureCode{43}{12}% +
\TMP@EnsureCode{46}{12}% .
\TMP@EnsureCode{47}{12}% /
\TMP@EnsureCode{91}{12}% [
\TMP@EnsureCode{93}{12}% ]
\edef\HOpatch@AtEnd{\HOpatch@AtEnd\noexpand\endinput}
```

Package identification.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{hopatch}%
  [2020-08-01 v1.4 Wrapper for package hooks / legacy code (v1.3) (HO)]
```

3.3 Resources

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname RequirePackage\endcsname\relax
  \def\TMP@RequirePackage#1[#2]{%
    \begingroup\expandafter\expandafter\expandafter\endgroup
    \expandafter\ifx\csname ver@#1.sty\endcsname\relax
      \input #1.sty\relax
    \fi
  }%
  \TMP@RequirePackage{ltxcmds}[2010/12/12]%
\else
  \RequirePackage{ltxcmds}[2010/12/12]%
\fi
```

\HOpatch@counter

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

\HOpatch@StepCounter

```
\ltx@ifundefined{numexpr}{%
  \def\HOpatch@StepCounter{%
    \begingroup
      \count@\HOpatch@counter\relax
      \advance\count@\ltx@one\relax
    \edef\x{\endgroup
      \noexpand\def\noexpand\HOpatch@counter{\the\count@}%
    }%
    \x
  }%
}{%
  \def\HOpatch@StepCounter{%
    \edef\HOpatch@counter{%
      \the\numexpr\HOpatch@counter+\ltx@one\relax
    }%
  }%
}
```

\HOpatch@list

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

\HOpatch@Add

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

3.4 Package patching

\hopatch@AfterPackage

```
\def\hopatch@AfterPackage#1{%
  \ltx@ifpackageloaded{#1}{%
    \ltx@firstofone
  }{%
    \HOpatch@AfterPackage{#1}%
  }%
}
```

\HOpatch@AfterPackage

```
\def\HOpatch@AfterPackage#1{%
  \edef\HOpatch@temp{#1}%
  \HOpatch@StepCounter
  \expandafter\HOpatch@@AfterPackage
  \csname HOpatch@\HOpatch@counter\expandafter\endcsname{%
    \HOpatch@temp
  }%
}
```

\HOpatch@@AfterPackage

```
\def\HOpatch@@AfterPackage#1#2#3{%
  \begingroup
    \toks@{#3}%
    \xdef\HOpatch@gtemp{%
      \noexpand\ltx@ifpackageloaded{#2}{%
        \noexpand\let\noexpand#1\noexpand\relax
        \the\toks@
      }{}%
    }%
  \endgroup
  \let#1\HOpatch@gtemp
  \HOpatch@Add#1%
  \HOpatch@Try{AfterPackage}{#2}#1%
  \HOpatch@Try{AtEndPackage}{#2}#1%
  \HOpatch@Try{AtEndOfPackageFile}{#2}#1%
}
```

\HOpatch@Try

```
\def\HOpatch@Try#1#2#3{%
  \ltx@ifundefined{#1}{}{%
    \csname #1\endcsname{#2}{#3}%
  }%
}
```

```
\AtBeginDocument{\HOpatch@list}
\ltx@ifundefined{AtEndPreamble}{}{%
  \ltx@ifundefined{@endpreamblehook}{}{%
    \AtEndPreamble{\HOpatch@list}%
  }%
}
\ltx@ifundefined{AfterEndPreamble}{}{%
  \ltx@ifundefined{@afterendpreamblehook}{}{%
    \AfterEndPreamble{%
      \let\HOpatch@OrgIfPackageLoaded\@ifpackageloaded
      \let\HOpatch@OrgIfPackageLater\@ifpackagelater
      \let\HOpatch@OrgIfClassLoaded\@ifclassloaded
      \let\HOpatch@OrgIfClassLater\@ifclasslater
      \let\@ifpackageloaded\ltx@ifpackageloaded
      \let\@ifpackagelater\ltx@ifpackagelater
      \let\@ifclassloaded\ltx@ifclassloaded
      \let\@ifclasslater\ltx@ifclasslater
      \HOpatch@list
      \let\@ifpackageloaded\HOpatch@OrgIfPackageLoaded
      \let\@ifpackagelater\HOpatch@OrgIfPackageLater
      \let\@ifclassloaded\HOpatch@OrgIfClassLoaded
      \let\@ifclasslater\HOpatch@OrgIfClassLater
    }%
  }%
}
```

```
\HOpatch@AtEnd%
```

```
⟨/packagefrozen⟩
```

4 Installation

4.1 Download

Package. This package is available on CTAN[1]:

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 `mktexlsr`.

4.4 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 pdfLaTeX:

---

[1] 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 ε-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.