The setouterhbox package

Heiko Oberdiek*

2016/05/16 v1.8

Abstract

If math stuff is set in an `\hbox`, then TeX performs some optimization and omits the implicite penalties `\binoppenalty` and `\relpenalty`. This packages tries to put stuff into an `\hbox` without getting lost of those penalties.

Contents

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

1 Documentation

1.1 Introduction

There is a situation in hyperref's driver for dvips where the user wants to have links that can be broken across lines. However dvips doesn't support the feature. With option breaklinks hyperref sets the links as usual, put them in a box and write the link data with box dimensions into the appropriate `\special`s. Then, however, it does not set the complete unbreakable box, but it unwrappes the material inside to allow line breaks. Of course line breaking and glue setting will falsify the link dimensions, but line breaking was more important for the user.

1.2 Acknowledgement

Jonathan Fine, Donald Arsenau and me discussed the problem in the newsgroup comp.text.tex where Damian Menscher has started the thread, see [1].

The discussion was productive and generated many ideas and code examples. In order to have a more permanent result I wrote this package and tried to implement most of the ideas, a kind of summary of the discussion. Thus I want and have to thank Jonathan Fine and Donald Arsenau very much.

Two weeks later David Kastrup (posting in comp.text.tex, [2]) remembered an old article of Michael Downes ([3]) in TUGboat, where Michael Downes already presented the method we discuss here. Nowadays we have ε-TEX that extends the tool set of a TEX macro programmer. Especially useful ε-TEX was in this package for detecting and dealing with errorneous situations.

However also nowadays a perfect solution for the problem is still missing at macro level. Probably someone has to go deep in the internals of the TEX compiler to implement a switch that let penalties stay where otherwise TEX would remove them for optimization reasons.

1.3 Usage

Package loading. LATEX: as usually:

```
\usepackage{setouterhbox}
```

The package can also be included directly, thus plain TEX users write:

```
\input setouterhbox.sty
```

Register allocation. The material will be put into a box, thus we need to know these box number. If you need to allocate a new box register:

LATEX: `\newsavebox{\`⟨*name*⟩`}`

plain TEX: `\newbox\`⟨*name*⟩

Then `\`⟨*name*⟩ is a command that held the box number.

Box wrapping. LATEX users put the material in the box with an environment similar to `lrbox`. The environment `setouterhbox` uses the same syntax and offers the same features, such as verbatim stuff inside:

`\begin{setouterhbox}{`⟨*box number*⟩`}`...`\end{setouterhbox}`

Users with plain TEX do not have environments, they use instead:

`\setouterhbox{`⟨*box number*⟩`}`...`\endsetouterhbox`

In both cases the material is put into an `\hbox` and assigned to the given box, denoted by ⟨*box number*⟩. Note the assignment is local, the same way `lrbox` behaves.

Unwrapping. The box material is ready for unwrapping:

`\unhbox`⟨*box number*⟩

1.4 Option hyperref

Package url uses math mode for typesetting urls. Break points are inserted by `\binoppenalty` and `\relpenalty`. Unhappily these break points are removed, if hyperref is used with option breaklinks and drivers that depend on pdfmark: dvips, vtexpdfmark, textures, and dvipsone. Thus the option hyperref enables the method of this package to avoid the removal of `\relpenalty` and `\binoppenalty`. Thus you get more break points. However, the link areas are still wrong for these drivers, because they are not supporting broken links.

Note, you need version 2006/08/16 v6.75c of package hyperref, because starting with this version the necessary hook is provided that package setouterhbox uses.

```
\usepackage[...]{hyperref}[2006/08/16]
\usepackage[hyperref]{setouterhbox}
```

Package order does not matter.

1.5 Example

```
⟨*example⟩
\documentclass[a5paper]{article}
\usepackage{url}[2005/06/27]
\usepackage{setouterhbox}

\newsavebox{\testbox}

\setlength{\parindent}{0pt}
\setlength{\parskip}{2em}

\begin{document}
\raggedright

\url{http://this.is.a.very.long.host.name/followed/%
by/a/very_long_long_long_path.html}%

\sbox\testbox{%
  \url{http://this.is.a.very.long.host.name/followed/%
  by/a/very_long_long_long_path.html}%
}%
\unhbox\testbox

\begin{setouterhbox}{\testbox}%
  \url{http://this.is.a.very.long.host.name/followed/%
  by/a/very_long_long_long_path.html}%
\end{setouterhbox}
\unhbox\testbox

\end{document}
⟨/example⟩
```

2 Implementation

Internal macros are prefixed by `\setouterhbox`, `@` is not used inside names, thus we do not need to care of its catcode if we are not using it as LaTeX package.

2.1 Package start stuff

```
⟨*package⟩
```

Prevent reloading more than one, necessary for plain TeX: Reload check, especially if the package is not used with LaTeX.

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \catcode35=6 % #
  \catcode39=12 % '
  \catcode44=12 % ,
  \catcode45=12 % -
  \catcode46=12 % .
  \catcode58=12 % :
  \catcode64=11 % @
  \catcode123=1 % {
  \catcode125=2 % }
  \expandafter\let\expandafter\x\csname ver@setouterhbox.sty\endcsname
  \ifx\x\relax % plain-TeX, first loading
  \else
    \def\empty{}%
    \ifx\x\empty % LaTeX, first loading,
      % variable is initialized, but \ProvidesPackage not yet seen
    \else
      \expandafter\ifx\csname PackageInfo\endcsname\relax
        \def\x#1#2{%
          \immediate\write-1{Package #1 Info: #2.}%
        }%
      \else
        \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
      \fi
      \x{setouterhbox}{The package is already loaded}%
      \aftergroup\endinput
    \fi
  \fi
\endgroup%
```

Package identification:

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \catcode35=6 % #
  \catcode39=12 % '
  \catcode40=12 % (
  \catcode41=12 % )
  \catcode44=12 % ,
  \catcode45=12 % -
  \catcode46=12 % .
  \catcode47=12 % /
  \catcode58=12 % :
  \catcode64=11 % @
  \catcode91=12 % [
  \catcode93=12 % ]
  \catcode123=1 % {
  \catcode125=2 % }
  \expandafter\ifx\csname ProvidesPackage\endcsname\relax
    \def\x#1#2#3[#4]{\endgroup
      \immediate\write-1{Package: #3 #4}%
      \xdef#1{#4}%
    }%
  \else
    \def\x#1#2[#3]{\endgroup
      #2[{#3}]%
      \ifx#1\@undefined
        \xdef#1{#3}%
      \fi
```

```
      \ifx#1\relax
        \xdef#1{#3}%
      \fi
    }%
  \fi
\expandafter\x\csname ver@setouterhbox.sty\endcsname
\ProvidesPackage{setouterhbox}%
  [2016/05/16 v1.8 Set hbox in outer horizontal mode (HO)]%
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \catcode123=1 % {
  \catcode125=2 % }
  \catcode64=11 % @
  \def\x{\endgroup
    \expandafter\edef\csname setouterhboxAtEnd\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\setouterhboxAtEnd{%
    \setouterhboxAtEnd
    \catcode#1=\the\catcode#1\relax
  }%
  \catcode#1=#2\relax
}
\TMP@EnsureCode{40}{12}% (
\TMP@EnsureCode{41}{12}% )
\TMP@EnsureCode{44}{12}% ,
\TMP@EnsureCode{45}{12}% -
\TMP@EnsureCode{46}{12}% .
\TMP@EnsureCode{47}{12}% /
\TMP@EnsureCode{58}{12}% :
\TMP@EnsureCode{60}{12}% <
\TMP@EnsureCode{62}{12}% >
\TMP@EnsureCode{91}{12}% [
\TMP@EnsureCode{93}{12}% ]
\TMP@EnsureCode{96}{12}% `
\edef\setouterhboxAtEnd{\setouterhboxAtEnd\noexpand\endinput}
```

2.2 Interface macros

`\setouterhboxBox` The method requires a global box assignment. To be on the safe side, a new box register is allocated for this global box assignment.

```
\newbox\setouterhboxBox
```

`\setouterhboxFailure` Error message for both plain TEX and LATEX

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname RequirePackage\endcsname\relax
```

```
  \input infwarerr.sty\relax
\else
  \RequirePackage{infwarerr}[2016/05/16]%
\fi
\edef\setouterhboxFailure#1#2{%
  \expandafter\noexpand\csname @PackageError\endcsname
      {setouterhbox}{#1}{#2}%
}
```

2.3 Main part

eTeX provides much better means for checking error conditions. Thus lines marked by "E" are executed if eTeX is available, otherwise the lines marked by "T" are used.

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname lastnodetype\endcsname\relax
  \catcode`T=9 % ignore
  \catcode`E=14 % comment
\else
  \catcode`T=14 % comment
  \catcode`E=9 % ignore
\fi
```

`\setouterhboxRemove` Remove all kern, glue, and penalty nodes; poor man's version, if ε-TEX is not available

```
\def\setouterhboxRemove{%
E \ifnum\lastnodetype<11 %
E   \else
E   \ifnum\lastnodetype>13 %
E   \else
      \unskip\unkern\unpenalty
E     \expandafter\expandafter\expandafter\setouterhboxRemove
E   \fi
E \fi
}%
```

`\setouterhbox` Passing the box contents by macro parameter would prevent catcode changes in the box contents like by `\verb`. Also `\bgroup` and `\egroup` does not work, because stuff has to be added at the begin and end of the box, thus the syntax `\setouterhbox{`⟨*box number*⟩`}`...`\endsetouterhbox` is used. Also we automatically get an environment `setouterhbox` if LaTeX is used.

```
\def\setouterhbox#1{%
  \begingroup
    \def\setouterhboxNum{#1}%
    \setbox0\vbox\bgroup
T     \kern.123pt\relax % marker
T     \kern0pt\relax % removed by \setouterhboxRemove
      \begingroup
        \everypar{}%
        \noindent
}
```

`\endsetouterhbox` Most of the work is done in the end part, thus the heart of the method follows:

```
\def\endsetouterhbox{%
      \endgroup
```

Omit the first pass to get the penalties of the second pass.

```
      \pretolerance-1 %
```

We don't want a third pass with `\emergencystretch`.

```
      \tolerance10000 %
      \hsize\maxdimen
```

Line is not underfull:

```
      \parfillskip 0pt plus 1filll\relax
      \leftskip0pt\relax
```

Suppress underful \hbox warnings, is explicit line breaks are used.

```
      \rightskip0pt plus 1fil\relax
      \everypar{}%
```

Ensure that there is a paragraph and prevents \endgraph from eating terminal glue:

```
      \kern0pt%
      \endgraf
      \setouterhboxRemove
E     \ifnum\lastnodetype=1 %
E       \global\setbox\setouterhboxBox\lastbox
E       \loop
E         \setouterhboxRemove
E       \ifnum\lastnodetype=1 %
E         \setbox0=\lastbox
E         \global\setbox\setouterhboxBox=\hbox{%
E           \unhbox0 %
```

Remove \rightskip, a penalty with -10000 is part of the previous line.

```
E           \unskip
E           \unhbox\setouterhboxBox
E         }%
E       \repeat
E     \else
E       \setouterhboxFailure{%
E         Something is wrong%
E       }{%
E         Could not find expected line.%
E         \MessageBreak
E         (\string\lastnodetype: \number\lastnodetype, expected: 1)%
E       }%
E     \fi
E     \setouterhboxRemove
T     \global\setbox\setouterhboxBox\lastbox
T     \loop
T       \setouterhboxRemove
T       \setbox0=\lastbox
T     \ifcase\ifvoid0 1\else0\fi
T       \global\setbox\setouterhboxBox=\hbox{%
T         \unhbox0 %
```

Remove \rightskip, a penalty with -10000 is part of the previous line.

```
T         \unskip
T         \unhbox\setouterhboxBox
T       }%
T     \repeat
T     \ifdim.123pt=\lastkern
T     \else
T       \setouterhboxFailure{%
T         Something is wrong%
T       }{%
T         Unexpected stuff was detected before the line.%
T       }%
T     \fi
T   \egroup
T   \ifcase \ifnum\wd0=0 \else 1\fi
T           \ifdim\ht0=.123pt \else 1\fi
T           \ifnum\dp0=0 \else 1\fi
T           0 %
E     \ifnum\lastnodetype=-1 %
```

There was just one line that we have caught.

```
      \else
        \setouterhboxFailure{%
            Something is wrong%
        }{%
            After fetching the line there is more unexpected stuff.%
E           \MessageBreak
E           (\string\lastnodetype: \number\lastnodetype, expected: -1)%
        }%
      \fi
E   \egroup
  \expandafter\endgroup
  \expandafter\setouterhboxFinish\expandafter{%
    \number\setouterhboxNum
  }%
}
```

2.4 Environment support

Check `\@currenvir` for the case that `\setouterhbox` was called as environment. Then the box assignment must be put after the `\endgroup` of `\end{...}`.

```
\def\setouterhboxCurr{setouterhbox}
\def\setouterhboxLast#1{%
  \setbox#1\hbox{%
    \unhbox\setouterhboxBox
    \unskip % remove \rightskip glue
    \unskip % remove \parfillskip glue
    \unpenalty % remove paragraph ending \penalty 10000
    \unkern % remove explicit kern inserted above
  }%
}
```

`\setouterhboxFinish` `#1` is an explicit number.

```
\def\setouterhboxFinish#1{%
  \begingroup\expandafter\expandafter\expandafter\endgroup
  \expandafter\ifx\csname @currenvir\endcsname\setouterhboxCurr
    \aftergroup\setouterhboxLast
    \aftergroup{%
    \setouterhboxAfter #1\NIL
    \aftergroup}%
  \else
    \setouterhboxLast{#1}%
  \fi
}
```

`\setouterhboxAfter` `#1` is an explicit number.

```
\def\setouterhboxAfter#1#2\NIL{%
  \aftergroup#1%
  \ifx\\#2\\%
  \else
    \setouterhboxReturnAfterFi{%
      \setouterhboxAfter#2\NIL
    }%
  \fi
}
```

`\setouterhboxReturnAfterFi` A utility macro to get tail recursion.

```
\long\def\setouterhboxReturnAfterFi#1\fi{\fi#1}
```

Restore catcodes we have need to distinguish between the implementation with and without ε-TEX.

```
\catcode69=11\relax % E
\catcode84=11\relax % T
```

2.5 Option hyperref

```
\begingroup
  \def\x{LaTeX2e}%
\expandafter\endgroup
\ifx\x\fmtname
\else
  \expandafter\setouterhboxAtEnd
\fi%
```

\Hy@setouterhbox `\Hy@setouterhbox` is the internal hook that hyperref uses since 2006/02/12 v6.75a.

```
\DeclareOption{hyperref}{%
  \long\def\Hy@setouterhbox#1#2{%
    \setouterhbox{#1}#2\endsetouterhbox
  }%
}

\ProcessOptions\relax

\setouterhboxAtEnd%
⟨/package⟩
```

3 Installation

3.1 Download

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

CTAN:macros/latex/contrib/oberdiek/setouterhbox.dtx The source file.

CTAN:macros/latex/contrib/oberdiek/setouterhbox.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 TEX 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 TEX:

```
tex setouterhbox.dtx
```

[1] CTAN:pkg/setouterhbox

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

```
setouterhbox.sty          → tex/generic/oberdiek/setouterhbox.sty
setouterhbox.pdf          → doc/latex/oberdiek/setouterhbox.pdf
setouterhbox-example.tex → doc/latex/oberdiek/setouterhbox-example.tex
setouterhbox.dtx          → source/latex/oberdiek/setouterhbox.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 TEX distribution (TEX Live, MiKTEX, ...) relies on file name databases, you must refresh these. For example, TEX Live users run `texhash` or `mktexlsr`.

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{setouterhbox.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:

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

4 References

[1] Damian Menscher, news:comp.text.tex, *overlong lines in List of Figures*, <dh058tqbd1@news.ks.uiuc.edu>, 23rd September 2005. https://groups.google.com/group/comp.text.tex/msg/79648d4cf1f8bc13

[2] David Kastrup, news:comp.text.tex, *Re: ANN: outerhbox.sty – collect horizontal material, for unboxing into a paragraph*, <85y855lrx3.fsf@lola.goethe.zz>, 7th October 2005. https://groups.google.com/group/comp.text.tex/msg/7cf0a345ef932e52

[3] Michael Downes, *Line breaking in \unhboxed Text*, TUGboat 11 (1990), pp. 605–612.

[4] Sebastian Rahtz, Heiko Oberdiek: *The hyperref package*; 2006/08/16 v6.75c; CTAN:pkg/hyperref.

5 History

[2005/10/05 v1.0]

- First version.

[2005/10/07 v1.1]

- Option hyperref added.

[2005/10/18 v1.2]

- Support for explicit line breaks added.

[2006/02/12 v1.3]

- DTX format.
- Documentation extended.

[2006/08/26 v1.4]

- Date of hyperref updated.

[2007/04/26 v1.5]

- Use of package infwarerr.

[2007/05/17 v1.6]

- Standard header part for generic files.

[2007/09/09 v1.7]

- Catcode section added.

[2016/05/16 v1.8]

- 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.