The \texttt{atbegshi} package

Heiko Oberdiek\textsuperscript{*}
\texttt{<heiko.oberdiek at googlemail.com>}

2016/06/09 v1.18

Abstract

This package is a modern reimplementation of package \texttt{everyshi} without the burden of compatibility. It makes use of $\varepsilon$-\TeX{}’s if available. Both \LaTeX{} and plain \TeX{} are supported.

Contents

1 Documentation ................................................. 2
  1.1 Examples ............................................. 4
      1.1.1 Example: circle in background .................. 4
      1.1.2 Example: adding TrimBox for dvipdfmx .......... 5

2 Method of \texttt{\textbackslash shipout} overloading ................. 6
  2.1 \texttt{\textbackslash shipout} .......................... 6
  2.2 \texttt{\textbackslash afterassignment} .................. 6
  2.3 Test for direct or indirect boxes .................... 7
      2.3.1 With $\varepsilon$-\TeX{} .......................... 7
      2.3.2 Without $\varepsilon$-\TeX{} ....................... 7
      2.3.3 \texttt{\textbackslash lastkern} method .......... 8
  2.4 Output .................................................. 9
  2.5 Separate box register ................................ 9
  2.6 Summary ................................................ 9
      2.6.1 With $\varepsilon$-\TeX{} .......................... 9
      2.6.2 Without $\varepsilon$-\TeX{}, traditional way .......... 10
      2.6.3 \texttt{\textbackslash lastkern} method .......... 10

3 Implementation .............................................. 11
  3.1 Reload check and package identification ............... 11
  3.2 Catcodes ............................................. 12
  3.3 Preparations .......................................... 13
  3.4 Additions to the shipout box .......................... 17
  3.5 Positioning ............................................ 19
  3.6 Patches ................................................ 20
      3.6.1 Package \texttt{crop} .............................. 20
      3.6.2 Package \texttt{everyshi} ......................... 22
      3.6.3 Class \texttt{memoir} ............................ 23

4 Test ......................................................... 26
  4.1 Catcode checks for loading ............................. 26

\textsuperscript{*}Please report any issues at https://github.com/ho-tex/oberdiek/issues
1 Documentation

Package \atbegshi redefines \shipout to insert hooks for user code that is executed before the page is shipped out. The code may modify or even discard the output page. Three hooks are implemented:

1. A hook that is executed for every page, see
   \AtBeginShipout

2. A hook that is executed for the next page only, see
   \AtBeginShipoutNext

3. A hook that is only executed for the first page, see
   \AtBeginShipoutFirst

The hooks are executed in this order. The following three macros provide the user interface for adding code to these hooks:

\begin{verbatim}
\AtBeginShipout {\langle code\rangle}
\AtBeginShipoutBox
\end{verbatim}

Execute the \langle code\rangle for every page. The page contents is held in box register \AtBeginShipoutBox and may be modified. Use \AtBeginShipoutDiscard if you want to discard the page.

Note: Package everyshi uses box register 255. With package atbegshi you must use \AtBeginShipoutBox instead.
If \LaTeX calls \texttt{\textbackslash shipout} in \texttt{@outputpage} (part of its output routine), the meaning of \texttt{\textbackslash protect} is not expanded. \LaTeX sets \texttt{\textbackslash protect} to the appropriate \texttt{\textbackslash type-set@protect} in the box that is shipped out. This is too late for the hooks, they are called earlier in the redefined \texttt{\textbackslash shipout}. Therefore package \texttt{atbegshi} sets \texttt{\textbackslash protect} to \texttt{\textbackslash type-set@protect} before it calls the hooks. (In \texttt{\textbackslash EveryShipout} of package \texttt{everyshi} the user is responsible for the correct setting of \texttt{\textbackslash protect}.)

\texttt{\textbackslash AtBeginShipoutNext \{\texttt{(code)}\}}

This reimplements package \texttt{everyshi}'s \texttt{\textbackslash AtNextShipout}. The \texttt{(code)} is executed at shipout time of the next page only. It is just a convenience macro, it can be easily replaced by something like:

\begin{verbatim}
\newcommand{\MyShipoutHook}{% 
  \AtBeginShipout{\MyShipoutHook} 
  \gdef{\MyShipoutHook}{% 
    ... do something with next page ... 
  } 
}
\end{verbatim}

(This can be necessary, if hook order does matter).

\texttt{\textbackslash AtBeginShipoutFirst \{\texttt{(code)}\}}

This reimplements \LaTeX's \texttt{\textbackslash AtBeginDvi}. This hook is usually used for \special commands that include PostScript header files. The \texttt{(code)} is directly executed in a \texttt{\vbox} that is put at the beginning of the output page. Dealing with the output box \texttt{\AtBeginShipoutBox} is not necessary and not permitted here.

\texttt{\textbackslash AtBeginShipoutDiscard}

This macro notifies package \texttt{atbegshi} that the output page is discarded. The remaining hook code and the remaining hooks are not executed and the page is thrown away. Also \texttt{\textbackslash deadcycles} is cleared to zero like an ordinary \texttt{\textbackslash shipout} would do.

\texttt{\textbackslash AtBeginShipoutInit}

Usually the redefinition of \texttt{\textbackslash shipout} is delayed by \texttt{\AtBeginDocument} (if this macro exists). This can be too late, if other packages also redefines \texttt{\textbackslash shipout} and the order does matter. \texttt{\AtBeginShipoutInit} forces the immediate redefinition of \texttt{\textbackslash shipout}.

\texttt{\textbackslash AtBeginShipoutAddToBox \{\texttt{(stuff)}\}}
\texttt{\textbackslash AtBeginShipoutAddToBoxForeground \{\texttt{(stuff)}\}}

A quite common use case is the addition of \special or other whatsits to the page output box. Macro \texttt{\AtBeginShipoutAddToBox} puts \texttt{(stuff)} in a box with zeroed dimensions. The box with the \texttt{(stuff)} is put in the upper left corner of the shipout box \texttt{\AtBeginShipoutBox}. Macro \texttt{\AtBeginShipoutAddToBoxForeground} in the foreground after the original shipout box contents is set.

A void shipout box (that means a discarded page) remains void that means \texttt{(stuff)} is ignored in this case. The box type of \texttt{\AtBeginShipoutBox} is preserved. Also the box nesting level for the original contents of \texttt{\AtBeginShipoutBox} remains, for example, to avoid trouble with links across pages in case of pdf\LaTeX.
\AtBeginShipoutUpperLeft \langle\text{background material}\rangle

This is a macro that puts material in the background of box \AtBeginShipoutBox. The \langle\text{background material}\rangle is set in an \hbox, the reference point is the upper left corner of the output page. In case of pdf\TeX in PDF mode, the settings of \pdfhorigin and \pdfvorigin are respected.

The macro \AtBeginShipoutUpperLeft is intended to be used in one of the hook setting macros, such as \AtBeginShipout, \AtBeginShipoutFirst, or \AtBeginShipoutNext.

For \LaTeX users the \langle\text{background material}\rangle is set inside a \texttt{picture} environment:

\begin{picture}(0,0)
\setlength{\unitlength}{1pt}\
\langle\text{background material}\rangle
\end{picture}

\AtBeginShipoutUpperLeftForeground \langle\text{foreground material}\rangle

See \AtBeginShipoutUpperLeft. The difference is that the material is put in the foreground.

\AtBeginShipoutOriginalShipout \langle\text{box}\rangle

It stores the meaning of \texttt{\shipout} at the time this package is loaded.

\AtBeginShipoutBoxWidth \AtBeginShipoutBoxHeight \AtBeginShipoutBoxDepth

These macros store the dimensions of the output box \AtBeginShipoutBox before the original shipout is called. If \texttt{\shipout} is not redefined before the package loading or the box dimensions are not changed by the redefined \texttt{\shipout}, these macros contain the dimensions of the shipout box. These values can be remembered by \texttt{\label} and \texttt{\ref}. For example, this is done by the package module \texttt{zref-pagelayout} of project \texttt{zref}. The dimensions of the shipout page can be used in some \TeX engines (pdf\TeX in PDF mode, \Xe\TeX) to calculate the media size of the shipout page if \texttt{\pdfpagewidth} and \texttt{\pdfpageheight} are not set.

1.1 Examples

1.1.1 Example: circle in background

In this example we put a circle in the background in the middle of the paper.

1 \{\example1\}
2 \documentclass[a4paper]{article}
3 \usepackage{color}
4 \usepackage[atbegshi]{picture}

Package picture makes life a little easier, because we can now also use length specifications in picture’s commands.

5 \usepackage{picture}

Now we draw the circle in the middle of the paper. \texttt{\put} moves downwards, because the origin is at the top of the page, not at its bottom.

6 \AtBeginShipout{\%
7 \AtBeginShipoutUpperLeft{\%
8 \put(0.5\textwidth,-0.5\textheight)\{\texttt{\circle(10)}\}}\%
1.1.2 Example: adding TrimBox for dvipdfmx

Now an example from “real life” follows. Someone from the mailing list for dvipdfmx wants to put a TrimBox on every page. If we use \AtBeginShipout, we have to put the \special inside the box \AtBeginShipoutBox that gets shipped out.

\documentclass{minimal}
\usepackage{atbegshi}
\usepackage[dvipdfm, paperwidth=630bp, paperheight=810bp]{geometry}
\AtBeginShipout{%
  \setbox\AtBeginShipoutBox=\hbox{\special{pdf: put @thispage <</TrimBox[9 9 621 801]>>}}
  \box\AtBeginShipoutBox
%
}
\begin{document}
First page
\newpage
Second page
\end{document}

Remember, in \AtBeginShipoutBoxFirst the \setbox wrapper code is implicitly given and the \special is used directly.
2 Method of \shipout overloading

2.1 \shipout

The \TeX{} primitive command \texttt{\shipout} takes a box specification and puts the box as a new page in the output file. There are two kinds of box specifications:

Direct boxes: They are given by \texttt{\hbox}, \texttt{\vbox}, or \texttt{\vtop}, e.g. \texttt{\shipout\hbox{Hello World}}.

Indirect boxes: \texttt{\box} or \texttt{\copy} references a box register by number. The box register contains the contents of the box.

Note: \texttt{\box} also clears the box register globally.

Then we have to differentiate between void and empty boxes:

Void: Initially or after \texttt{\box} there is no box in the box register. In this case the box register is not empty, but \texttt{void}.

Empty: A box with empty contents, such as \texttt{\hbox{}} (\texttt{= \null}) or \texttt{\vbox{}} is an empty \texttt{\hbox} or empty \texttt{\vbox}. If a box register holds such a box, the box still exists, therefore the box register is \texttt{not void}.

2.2 \afterassignment

We want to overload \texttt{\shipout} to do something with the box. It is quite impossible to do this reliably by catching the box using macro arguments. The variety of box specifications is too large. Examples:

\begin{verbatim}
\shipout\null
\shipout\vbox{...}
\shipout\vtop\bgroup ...\egroup
\shipout\box255
\end{verbatim}

Even worse, the braces don’t need to be balanced:

\begin{verbatim}
\shipout\hbox\bgroup}
\shipout\vbox{\egroup
\end{verbatim}

Happily \TeX{} provides a reliable way via \texttt{\afterassignment}. It takes a macro name and executes it just after the assignment.

Now we can redefine \texttt{\shipout}. The box specification that follows \texttt{\shipout} is caught by \texttt{\setbox}. This is an assignment to a box register. \texttt{\afterassignment} notifies \TeX{}, that we want to call \texttt{\@test} right after the assignment:

\begin{verbatim}
\shipout :=
\afterassignment\@test
\setbox\mybox=
\end{verbatim}

We have seen different box specifications. Indirect boxes are easy to understand:

\begin{verbatim}
\shipout\box0 \Rightarrow \setbox\mybox=\box0 \@test
\end{verbatim}

However direct boxes can have arbitrary contents with lots of other assignments. It would be quite unpredictable if \TeX{} would put \texttt{\@test} after the first of such an assignment or after the box specification if the box lacks of assignments. Therefore \TeX{} puts \texttt{\@test} right at the beginning of the box specification, e.g:

\begin{verbatim}
\shipout\hbox{Hello World}
\Rightarrow \setbox\mybox=\hbox{\@test Hello World}
\end{verbatim}
2.3 Test for direct or indirect boxes

Now we want to execute `\@test`, but where are we? We can be after the completed box assignment, if `\shipout` was called with an indirect box. Or we are right at the beginning of a direct box.

2.3.1 With \(\varepsilon\)-\TeX

With the \(\varepsilon\)-\TeX's extensions the answer is very easy: Being inside the direct box means that we are inside a new group. The new primitive command `\currentgrouplevel` tells how deeply the groups are currently nested. Macro `\@test` just compares the previously stored group level with the current one:

```latex
\shipout :=
  \edef\saved@grouplevel{\number\currentgrouplevel}
\afterassignment\@test
\setbox\mybox=
\@test :=
  \ifnum\saved@grouplevel=\currentgrouplevel
  % case: indirect box, the assignment is completed
  \@output
  \else
  % case: direct box, we are inside the box
  \aftergroup\@outbox
  \fi
```

2.3.2 Without \(\varepsilon\)-\TeX

Life becomes complicated without \(\varepsilon\)-\TeX. We cannot ask the group level. However, if we are inside a direct box, the box register `\mybox` is not yet changed by `\setbox`. Thus we need a special initial value and compare it in `\@test` with the current value of the box.

What can be used as initial value? Arbitrary box contents cannot be compared. \TeX only tells us a few properties:

- Box type:`\ifhbox`, `\ifvbox`
- Dimensions: `\wd`, `\ht`, `\dp`
- Voidness:`\ifvoid`

Unhappily all these qualities even combined are not sufficient for constructing an initial box value, because `\shipout` can be called with a box that is accidently just the same as the chosen initial value.

Nevertheless we have two alternatives for an initial value:

- A box of some type with some funny settings that are unlikely to occur in real life, e.g a height of `4911\text{sp}\text{-maxdimen}`.
- A void box.

A collision between this initial value and an indirect `\shipout` box with just the same value is possible. Then `\@test` will make a wrong decision that it is executed inside a direct box and delays `\@output` by `\aftergroup`. Thus `\@output` is not called at the place we want. In contrary, the result is an uncertainty about the place:

- `\shipout` is used in a group that perhaps closes some pages later. A bad place for `\@output`.
- Without a surrounding group `\aftergroup` effectively kills its argument.
In the first case of a box with special dimensions we can even loose the page. However in the case of the void box, this effect is even desired, because the original \texttt{shipout} does not output void boxes. All we have to do is to ensure that our box \texttt{mybox} is always void except for the phase when the overloaded \texttt{shipout} is executed. And secondly we must keep this semantics of \texttt{shipout} for the void case in our macros, namely \texttt{\@output}.

\begin{verbatim}
\texttt{\@output := \\
  \% trick to get a void box mybox
  \begingroup
    \setbox\mybox=\box\mybox
  \endgroup
  \afterassignment\@test
  \setbox\mybox=}

  \@test := \\
  \ifvoid\mybox
    \aftergroup\@output
  \else
    \@output
  \fi
\end{verbatim}

The nasty case is \texttt{\@output\box\voidb@x} where the indirect box is void and that must not generate an output page. If a surrounding group is missing the output is ignored because of \texttt{\aftergroup}. Otherwise output is called some time later when the surrounding group closes. But \texttt{mybox} is void outside the execution phase of the redefined \texttt{shipout}. Also \texttt{\@output} checks for a void box and cancels the page output. The disadvantage remains that the hook in \texttt{\@output} is called for a page that will not be output.

2.3.3 \texttt{\lastkern} method

At the beginning of a new box, there is no \texttt{kern}, the contents of the box is still empty and \texttt{\lastkern} returns 0 pt. This can be used to distinguish between direct and indirect boxes: We execute \texttt{\setbox} in a box with a preceding non-zero kern. After an indirect box, \texttt{\lastkern} sees this kern, otherwise it returns 0 pt.

\begin{verbatim}
\texttt{\@test := \\
  \ifdim\lastkern=0pt
    \% direct box
    \aftergroup\egroup
  \else
    \aftergroup\@output
  \fi
}\end{verbatim}

We have two \texttt{\setbox} commands. The first creates a controlled context box where we can safely insert a \texttt{kern}. We get rid of this temporarily used context box by putting the local \texttt{kern} in a group.

After the group we want to have our shipout box in \texttt{mybox}. Therefore we use a global assignment here.
2.4 Output

With or without \$\varepsilon\$-\TeX\ we ensure the original behaviour of \texttt{\shipout} that void boxes do not generate output pages.

Now we can place the hook \texttt{@hook} for the user code that wants to manipulate the output box.

\begin{verbatim}
@output :=
  ifvoid\mybox
    % cancel output of void box
  else
    @hook
  ifvoid\mybox
    % user code in @hook could has voided the box
  else
    \original@shipout\box\mybox
  fi
fi
\end{verbatim}

2.5 Separate box register

So far we have said nothing about the box number of \texttt{\mybox}. The following case that outputs the same page twice shows that we are not free in the use of the box register:

\begin{verbatim}
\shipout\copy<num> \shipout\box<num>
\end{verbatim}

We manipulate the box by the hook and without \$\varepsilon\$-\TeX\ the box must even be voided. However, the use case above requires that the box contents does not change at all. Therefore we must reserve a separate box register to avoid collisions with user box registers.

\textit{Note:} Box register number 255 is special for the output routine, because \TeX\ complains if this box is not voided by the output routine. However, this requirement does not apply to \texttt{\shipout} at all. In fact \texttt{\shipout} does not change any box register. This is usually done by a call of \texttt{\box}, but the output routine can do it later \textit{after} invoking of \texttt{\shipout}.

2.6 Summary

2.6.1 With \$\varepsilon\$-\TeX\n
Putting the pieces together we get for \$\varepsilon\$-\TeX:\n
\begin{verbatim}
\newbox\mybox
\let\original@shipout\shipout

\shipout := \edef\saved@grouplevel{\number\currentgrouplevel}
  \afterassignment@test
  \setbox\mybox=

@test := \ifnum\saved@grouplevel<\currentgrouplevel
  \expandafter\aftergroup
else
  \copynum\saved@grouplevel\currentgrouplevel
  \expandafter\aftergroup
fi
@output

@output := \ifvoid\mybox
  % cancel output of void box
else
  @hook
  \ifvoid\mybox
\end{verbatim}
\% user code in \@hook could have voided the box
\else
  \original@shipout\box\mybox
\fi
\fi

2.6.2 Without $\varepsilon$-TEX, traditional way

And for \TeX without $\varepsilon$-\TeX:

\newbox\mybox
\begingroup
  \setbox\mybox=\box\mybox \% ensure \mybox is void
\endgroup
\let\original@shipout\shipout

\setbox\mybox=
\begingroup
  \setbox\mybox=\box\mybox
\endgroup
\afterassignment\@test
\setbox\mybox=
\@test :=
\ifvoid\mybox
  \expandafter\aftergroup
\fi
\@output
\@output :=
\ifvoid\mybox
  \% cancel output of void box \mybox
\else
  \@hook
    \ifvoid\mybox
      \% user code in \@hook could have voided the box
    \else
      \original@shipout\box\mybox
    \fi
\fi
\fi

2.6.3 \texttt{\lastkern} method

And for \TeX without $\varepsilon$-\TeX using the \texttt{\lastkern} method:

\newbox\mybox
\let\original@shipout\shipout

\setbox\mybox=
\begingroup
  \setbox\mybox=\hbox\bgroup
    \kern1pt
  \afterassignment\@test
  \setbox\mybox=
\@test :=
\ifdim\lastkern=0pt
  \expandafter\aftergroup
\fi
\@output
\@output :=
3 Implementation

Package `atbegshi` uses e-TeX’s `\currentgrouplevel`, if it is available. Otherwise the `\lastkern` method is used.

3.1 Reload check and package identification

Reload check, especially if the package is not used with LATEX.

3 Implementation

Package `atbegshi` uses e-TeX’s `\currentgrouplevel`, if it is available. Otherwise the `\lastkern` method is used.

3.1 Reload check and package identification

Reload check, especially if the package is not used with LATEX.
\catcode41=12 % )
\catcode44=12 % ,
\catcode45=12 % -
\catcode46=12 % .
\catcode47=12 % /
\catcode48=12 % :
\catcode49=12 % @
\catcode58=12 % :
\catcode61=12 % [
\catcode62=12 % ]
\catcode63=12 % {
\catcode64=12 % }
\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@atbegshi.sty\endcsname
\ProvidesPackage{atbegshi}[
[2016/06/09 v1.18 At begin shipout hook (HO)]%

3.2 Catcodes
\begingroup\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 % ^^M
\catcode123=1 % {
\catcode125=2 % }
\def\TMP@EnsureCode#1#2{\
\edef\AtBegShi@AtEnd{\
\AtBegShi@AtEnd
\catcode#1=\the\catcode#1\relax
\catcode#2=\the\catcode#2\relax
\catcode35=\the\catcode35\relax
\catcode64=\the\catcode64\relax
\catcode123=\the\catcode123\relax
\catcode125=\the\catcode125\relax
}\}%
\x\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 % ^^M
\catcode35=6 % #
\catcode64=11 % @
\catcode123=1 % {
\catcode125=2 % }
\def\TMP@EnsureCode#1#2{\
\edef\AtBegShi@AtEnd{\
\AtBegShi@AtEnd
\catcode#1=\the\catcode#1\relax
\catcode#2=\the\catcode#2\relax
\catcode35=\the\catcode35\relax
\catcode64=\the\catcode64\relax
\catcode123=\the\catcode123\relax
\catcode125=\the\catcode125\relax
}\}%
\x\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 % ^^M
\catcode35=6 % #
\catcode64=11 % @
\catcode123=1 % {
\catcode125=2 % }
\def\TMP@EnsureCode#1#2{\
\edef\AtBegShi@AtEnd{\
\AtBegShi@AtEnd
\catcode#1=\the\catcode#1\relax
\catcode#2=\the\catcode#2\relax
\catcode35=\the\catcode35\relax
\catcode64=\the\catcode64\relax
\catcode123=\the\catcode123\relax
\catcode125=\the\catcode125\relax
}\}%
3.3 Preparations

\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}
\else
\RequirePackage{infwarerr}[2007/09/09]%
\RequirePackage{ltxcmds}[2010/03/01]%
\fi
\AtBegShi@CheckDefinable
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname @ifdefinable\endcsname\relax
\def\AtBegShi@CheckDefinable#1{%
\ifcase\ifx#1\relax
\ltx@one
\else
\ifx#1\@undefined
\ltx@one
\else
\ltx@zero
\fi
\fi
\@PackageError{atbegshi}{\string#1\space is already defined%}
\@ehd
\fi
\else
\def\AtBegShi@CheckDefinable#1{%
\@ifdefinable{#1}{}%
\fi
\ifAtBegShi@Discarded
\ltx@newif\ifAtBegShi@Discarded
\AtBeginShipoutDiscard
\AtBegShi@CheckDefinable\AtBeginShipoutDiscard
\def\AtBeginShipoutDiscard{%
\@PackageError{atbegshi}{%}
\string#1\space is already defined%
\}}\@ehd
\fi
\else
\def\AtBegShi@CheckDefinable#1{%
\@ifdefinable{#1}{%}
\}@\dfrac{\^\text{(superscript)}}{}
\def\AtBeginShipoutDiscard{%
\@PackageError{atbegshi}{%}
\string#1\space is already defined%
\}}\@ehd
\fi
\fi
\AtBeginShipoutDiscard
\def\AtBegShi@CheckDefinable{%
\@ifdefinable{#1}{%}
\@PackageError{atbegshi}{%}
\string#1\space is already defined%
\}}\@ehd
\fi
\else
\def\AtBegShi@CheckDefinable{%
\@ifdefinable{#1}{%}
\@PackageError{atbegshi}{%}
\string#1\space is already defined%
\}}\@ehd
\fi
\ifAtBegShi@Discarded
\ltx@newif\ifAtBegShi@Discarded
\AtBeginShipoutDiscard
\AtBegShi@CheckDefinable\AtBeginShipoutDiscard
\def\AtBeginShipoutDiscard{%
\@PackageError{atbegshi}{%}
\string#1\space is already defined%
\}}\@ehd
\fi
\else
\def\AtBegShi@CheckDefinable{%
\@ifdefinable{#1}{%}
\@PackageError{atbegshi}{%}
\string#1\space is already defined%
\}}\@ehd
\fi
\deadcycles=\ltx@zero
\global\AtBegShi@Discardedtrue
}
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname currentgrouplevel\endcsname\relax
\catcode`X=9 % ignore
\catcode`E=14 % comment
\else
\catcode`X=14 % comment
\catcode`E=9 % ignore
\fi
\AtBegShi@Shipout
\def\AtBegShi@Shipout{%
X \begingroup
X \setbox\AtBeginShipoutBox=\hbox\begingroup
X \kern\p@
E \edef\AtBegShi@GroupLevel{\number\currentgrouplevel}%
\afterassignment\AtBegShi@Test
X \global
\setbox\AtBeginShipoutBox=%
}
\AtBegShi@Test
\def\AtBegShi@Test{%
X \ifdim\lastkern=0pt %
E \ifnum\AtBegShi@GroupLevel<\currentgrouplevel
\expandafter\aftergroup
\fi
\AtBegShi@Output
\}
\AtBegShi@Output
\def\AtBegShi@Output{%
X \egroup
X \endgroup
\ifvoid\AtBeginShipoutBox
@PackageWarning{atbegshi}{Ignoring void shipout box}%
\else
\let\AtBegShi@OrgProtect\protect
\csname set@typeset@protect\endcsname
\global\AtBegShi@Discardedfalse
\AtBegShi@Hook
\expandafter\gdef\expandafter\AtBegShi@HookNext
\expandafter{\expandafter}
\AtBegShi@HookNext
\ifAtBegShi@Discarded
@PackageInfoNoLine{atbegshi}{Shipout page discarded}%
\global\AtBegShi@Discardedfalse
\begingroup
\setbox\AtBeginShipoutBox\box\AtBeginShipoutBox
\let\protect\AtBegShi@OrgProtect
\else
\AtBegShi@First
\let\protect\AtBegShi@OrgProtect
\AtBegShi@GetBoxSize\AtBeginShipoutBox
\ltx@ifundefined{AtNextShipout}{%
\global\AtBegShi@Discardedfalse
\begingroup
\setbox\AtBeginShipoutBox=\box\AtBeginShipoutBox
\endgroup
\let\protect\AtBegShi@OrgProtect
\else
\AtBegShi@First
\let\protect\AtBegShi@OrgProtect
\AtBegShi@GetBoxSize\AtBeginShipoutBox
\ltx@ifundefined{AtNextShipout}{%
\global\AtBegShi@Discardedfalse
\begingroup
%
\AtBegShi@GetBoxSize
\def\AtBegShi@GetBoxSize#1{% 
\xdef\AtBeginShipoutBoxWidth{\the\wd#1}%
\xdef\AtBeginShipoutBoxHeight{\the\ht#1}%
\xdef\AtBeginShipoutBoxDepth{\the\dp#1}%
}
\AtBeginShipoutBoxWidth
\def\AtBeginShipoutBoxWidth{0pt}
\AtBeginShipoutBoxHeight
\def\AtBeginShipoutBoxHeight{0pt}
\AtBeginShipoutBoxDepth
\def\AtBeginShipoutBoxDepth{0pt}
\catcode`X=11 %
\catcode`E=11 %
\AtBegShi@First
\def\AtBegShi@First{%
\ifx\AtBegShi@HookFirst\ltx@empty
\else
\AtBeginShipoutAddToBox{\AtBegShi@HookFirst}%
\fi
\global\let\AtBegShi@First\ltx@empty
\global\let\AtBeginShipoutFirst\AtBegShi@FirstDisabled
}
\AtBegShi@Hook
\gdef\AtBegShi@Hook{}
\AtBegShi@HookNext
\gdef\AtBegShi@HookNext{}
\AtBegShi@HookFirst
\gdef\AtBegShi@HookFirst{}
\AtBeginShipout
\AtBegShi@CheckDefinable\AtBeginShipout
\def\AtBeginShipout{%
\AtBegShi@AddHook\AtBegShi@Hook
}
\AtBeginShipoutNext
\AtBegShi@CheckDefinable\AtBeginShipoutNext
\def\AtBeginShipoutNext{%
\AtBegShi@AddHook\AtBegShi@HookNext
}
\AtBeginShipoutFirst
\AtBegShi@CheckDefinable\AtBeginShipoutFirst
\def\AtBeginShipoutFirst{%
\AtBegShi@AddTo\AtBegShi@HookFirst
}
\AtBeginShipoutFirstDisabled
\long\def\AtBeginShipoutFirstDisabled#1{%
  \PackageWarning{atbegshi}{%
  First page is already shipped out, ignoring\MessageBreak
  string\AtBeginShipoutFirst%
}%

\AtBeginShipoutInit
\def\AtBeginShipoutInit{%
  \IfUndefined{newbox}{%
    \PackageError{atbegshi}{%
      \string\AtBeginShipoutInit \space failed\MessageBreak
      because of missing \expandafter\string\cname newbox\endsname
    }\@ehc
  }{%}
  \csname newbox\endcsname\AtBeginShipoutBox
  \AtBeginShipoutOriginalShipout
  \global\let\shipout\AtBeginShipoutShipout
  \global\let\AtBeginShipoutShipout\shipout
}%
\gdef\AtBeginShipoutInit{}

\AtBeginShipoutAddTo
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname g@addto@macro\endcsname\relax
\long\def\AtBeginShipoutAddTo#1#2{%
  \begingroup
  \toks\ltx@zero\expandafter{#1#2}%
  \xdef#1{\the	oks\ltx@zero}%
  \endgroup
}%
\else
\let\AtBeginShipoutAddTo\g@addto@macro
\fi

\AtBeginShipoutAddHook
\long\def\AtBeginShipoutAddHook#1#2{%
  \AtBeginShipoutAddTo#1{\AtBeginShipoutItem{#2}}%
}

\AtBeginShipoutItem
\long\def\AtBeginShipoutItem#1{%
  \ifAtBeginShipout@Discarded
  #1%
  \else
  \ifvoid\AtBeginShipoutBox
    \PackageWarning{atbegshi}{%
    Shipout box was voided by hook,\MessageBreak
    ignoring shipout box%
  }%
  \AtBeginShipoutDiscard
  \fi
  \fi
  \fi
}

\AtBeginShipoutDiscard
\@PackageWarning{atbegshi}{%
Discarding shipout box to avoid endless loop%
}%

\AtBeginShipoutBox
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname g@addto@macro\endcsname\relax
\long\def\AtBeginShipoutBox#1{%
  \begingroup
  \toks\ltx@zero\expandafter{#1}%
  \xdef#1{\the	oks\ltx@zero}%
  \endgroup
}%
\else
\let\AtBeginShipoutBox\g@addto@macro
\fi
3.4 Additions to the shipout box

\AtBeginShipoutAddToBox

\def\AtBeginShipoutAddToBox#1{%}
\ifhbox\AtBeginShipoutBox
\edef\AtBegShi@restore{%
\hfuzz=\the\hfuzz\relax
\hbadness=\the\hbadness\relax
}%
\hfuzz=1073741823sp\relax
\hbadness=2147483647\relax
\setbox\AtBeginShipoutBox=\hbox to \wd\AtBeginShipoutBox{%
\setbox\ltx@zero=\hbox{%
\begingroup
\AtBegShi@restore
#1%
\endgroup
}
\wd\ltx@zero=0pt\relax
\ht\ltx@zero=0pt\relax
\dp\ltx@zero=0pt\relax
\raise\ht\AtBeginShipoutBox\copy\ltx@zero
\unhcopy\AtBeginShipoutBox
}%
\AtBegShi@restore
\else
\ifvbox\AtBeginShipoutBox
\edef\AtBegShi@restore{%
\vfuzz=\the\vfuzz\relax
\vbadness=\the\vbadness\relax
\dimen\ltx@zero=\the\dimen\ltx@zero\relax
}%
\edef\AtBegShi@restorebox{%
\ht\AtBeginShipoutBox=\the\ht\AtBeginShipoutBox\relax
\dp\AtBeginShipoutBox=\the\dp\AtBeginShipoutBox\relax
}%
\vfuzz=1073741823sp\relax
\vbadness=2147483647\relax
\dimen\ltx@zero=\ht\AtBeginShipoutBox
\advance\dimen\ltx@zero by \dp\AtBeginShipoutBox
\setbox\AtBeginShipoutBox=\vbox to \dimen\ltx@zero{%
\setbox\ltx@zero=\hbox{%
\begingroup
\AtBegShi@restore
#1%
\endgroup
}
\wd\ltx@zero=0pt\relax
\ht\ltx@zero=0pt\relax
\dp\ltx@zero=0pt\relax
\baselineskip=0pt\relax
\lineskip=0pt\relax
\lineskiplimit=0pt\relax
\copy\ltx@zero
\unvbox\AtBeginShipoutBox
\kern0pt%
}
3.5 Positioning

```latex
\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
}\fi
\TMP@RequirePackage{ifpdf}[2011/01/30]%
\else
  \RequirePackage{ifpdf}[2011/01/30]%
\fi
\ifpdf
  \def\AtBegShi@horigin{%
    \ifx\pdfhorigin\@undefined\pdfvariable horigin\else\pdfhorigin\fi}%
  \def\AtBegShi@vorigin{%
    \ifx\pdfvorigin\@undefined\pdfvariable vorigin\else\pdfvorigin\fi}%
\else
  \def\AtBegShi@horigin{72.27pt}%
  \def\AtBegShi@vorigin{72.27pt}%
\fi
\begingroup
\ifcase
  \expandafter\ifx\csname picture\endcsname\relax
  1%
  \else
  \expandafter\ifx\csname endpicture\endcsname\relax
  1%
  \else
  0%
  \fi
\fi
\fi
\endgroup
\def\AtBegShi@BeginPicture{%
  \begingroup
  \picture(0,0)\relax
  \begingroup\expandafter\expandafter\expandafter\endgroup
  \expandafter\ifx\csname unitlength\endcsname\relax
    \else
      \unitlength=1pt\relax
  \fi
  \ignorespaces
\endgroup
\def\AtBegShi@EndPicture{%
  \endpicture
  \endgroup
\endgroup
```

\AtBeginShipoutUpperLeft \ignorespaces
\AtBeginShipoutUpperLeft

A surrounding \rlap is not necessary, because the stuff is put in an \hbox with zero width.

\AtBeginShipoutUpperLeftForeground

3.6 Patches

Patches for \LaTeX{} packages that redefine \shipout. \LaTeX{} is now supposed to use \varepsilon-\TeX{}. Thus we do not patch, without \LaTeX{} and \varepsilon-\TeX{}.

3.6.1 Package crop

Fix of method and box.
3.6.2 Package everyshi

Fix of method. Use of box 255 is not changed.

\def\AtBeginShipout@Everyshi@Test{}
\begin{group}
  \long\def\AtBeginShipout@Everyshi@Test{\global\setbox\@cclv=\vbox{\@PackageInfoNoLine{everyshi}{Package `crop' patched}}%
  \begingroup
  \fi
  \fi
  \fi
  \fi
  \endgroup
  \let\AtBeginShipout@Everyshi=\relax
\end{group}
\@ifpackageloaded{crop}{}
\AtBeginDocument{\AtBeginShipout@Everyshi}
3.6.3 Class memoir

Fix of method and box.

\def\AtBegShi@PatchMemoir{%
\begingroup
\def\AtBegShi@Memoir@shipout{%
\afterassignment\mem@shipi
\setbox\@cclv=%
}%
\def\AtBegShi@Memoir@shipi{%
\ifvoid\@cclv\expandafter\aftergroup\fi
\mem@shipii
}%
\def\AtBegShi@Memoir@shipiiA{%
\mem@oldshipout\vbox{%
trimmarks
\unvbox\@cclv
}%
}%
\def\AtBegShi@Memoir@shipiiB{%
\ifvoid\@cclv
\mem@oldshipout\box\@cclv
\else
\mem@oldshipout\vbox{%
trimmarks
}%
}%
\begingroup
\ifx\AtBegShi@found\relax
\else
\expandafter\endgroup
\expandafter\def\AtBegShi@found{%
\edef\AtBegShi@GroupLevel{\number\currentgrouplevel}%
\afterassignment\EveryShipout@Test
\setbox\AtBeginShipoutBox=%
}%
\def\@EveryShipout@Test{%
\ifnum\AtBegShi@GroupLevel=\currentgrouplevel
\else
\expandafter\aftergroup
\fi
\AtBegShi@Everyshi@Output
}%
\def\AtBegShi@Everyshi@Output{%
\ifvoid\AtBeginShipoutBox
\else
\global\setbox\ltx@cclv\box\AtBeginShipoutBox
\expandafter\@EveryShipout@Output
\fi
}%
}@PackageInfoNoLine{atbegshi}{Package 'everyshi' patched}%
\begingroup
\fi 
\fi
\endgroup
\let\AtBegShi@PatchEveryshi\relax
}%
\@ifpackageloaded{everyshi}{{%
\AtBegShi@PatchEveryshi
}%
\AtBeginDocument{\AtBegShi@PatchEveryshi} %
\def\AtBegShi@Memoir@PatchAB{%
  \ifvoid\AtBeginShipoutBox
    \else
      \setbox\AtBeginShipoutBox=\vbox{%
        \trimmarks
        \ifvbox\AtBeginShipoutBox
          \unvbox\AtBeginShipoutBox
        \else
          \box\AtBeginShipoutBox
        \fi
      }%
      \AtBegShi@GetBoxSize\AtBeginShipoutBox
      \expandafter\mem@oldshipout\expandafter\box\expandafter\AtBeginShipoutBox
    \fi}
\def\AtBegShi@Memoir@shipiiC{% 2008/08/07 v1.6180339a
  \ifvoid\@cclv
    \mem@oldshipout\box\@cclv
  \else
    \ifshowtrims
      \mem@oldshipout\vbox{\trimmarks\unvbox\@cclv}%
    \else
      \mem@oldshipout\box\@cclv
    \fi
  \fi
}\def\AtBegShi@Memoir@shipiiD{% 2011/03/06 v3.6j
  \ifvoid\@cclv
    \mem@oldshipout\box\@cclv
  \else
    \ifshowtrims
      \mem@oldshipout\vbox{\trimmarks\nointerlineskip\box\@cclv}%
    \else
      \mem@oldshipout\box\@cclv
    \fi
  \fi
}\def\AtBegShi@Memoir@PatchCD{%
  \ifvoid\AtBeginShipoutBox
    \else
      \ifshowtrims
        \setbox\AtBeginShipoutBox=\vbox{%
          \trimmarks
          \nointerlineskip
          \box\AtBeginShipoutBox
        }%
      \else
        \AtBegShi@GetBoxSize\AtBeginShipoutBox
        \expandafter\mem@oldshipout\expandafter\box\expandafter\AtBeginShipoutBox
      \fi
  \fi
}\def\AtBegShi@Memoir@shipiiE{% 2011/03/06 v3.6j
  \ifvoid\@cclv
    \mem@oldshipout\box\@cclv
  \else
    \ifshowtrims
      \mem@oldshipout\vbox{\trimmarks\nointerlineskip\box\@cclv}%
    \else
      \mem@oldshipout\box\@cclv
    \fi
  \fi
}\def\AtBegShi@Memoir@PatchEE{%
  \ifvoid\AtBeginShipoutBox
    \else
      \ifshowtrims
        \setbox\AtBeginShipoutBox=\vbox{%
          \trimmarks
          \nointerlineskip
          \box\AtBeginShipoutBox
        }%
      \else
        \AtBegShi@GetBoxSize\AtBeginShipoutBox
        \expandafter\mem@oldshipout\expandafter\box\expandafter\AtBeginShipoutBox
      \fi
  \fi
}
4 Test

4.1 Catcode checks for loading

\begin{verbatim}
4.0 \documentclass\{article\}
4.1 \catcode\{=1
4.2 \catcode\}=2
4.3 \catcode\#=6
4.4 \catcode\@=11
4.5 \expandafter\ifx\csname count@\endcsname\relax
4.6 \countdef\count@=255
4.7 \fi
4.8 \expandafter\ifx\csname @gobble\endcsname\relax
4.9 \long\def\@gobble#1{}\fi
4.10 \expandafter\ifx\csname @firstofone\endcsname\relax
4.11 \long\def\@firstofone#1{#1}\fi
4.12 \expandafter\if\csname loop\endcsname\relax
4.13 \def\iterate{\body\let\next\iterate\else\let\next\relax\fi\next}
4.14 \def\RestoreCatcodes{}\count@=0
4.15 \loop\edef\RestoreCatcodes{\RestoreCatcodes\catcode\the\count@=\the\catcode\count@\relax}
4.16 \ifnum\count@<255 \advance\count@ 1 \repeat
4.17 \def\RangeCatcodeInvalid#1#2\{\count@=#1\relax\loop\ifnum\count@<#2\relax\advance\count@ 1 \repeat}
4.18 \def\RangeCatcodeCheck#1#2#3\{\count@=#1\relax\loop\ifnum#3=\catcode\count@ \else\fi\next\let\repeat=\fi\}\let\repeat=\fi
4.19 \def\RestoreCatcodes{}\count@=0
4.20 \loop\edef\RestoreCatcodes{\RestoreCatcodes\catcode\the\count@=\the\catcode\count@\relax}
4.21 \ifnum\count@<255 \advance\count@ 1 \repeat
4.22 \def\RangeCatcodeInvalid#1#2\{\count@=#1\relax\loop\ifnum\count@<#2\relax\advance\count@ 1 \repeat}
4.23 \def\RangeCatcodeCheck#1#2#3\{\count@=#1\relax\loop\ifnum#3=\catcode\count@ \else\fi\next\let\repeat=\fi\}\let\repeat=\fi
\end{verbatim}

\end{document}
\errmessage{Character \the\count@ \space with wrong catcode \the\catcode\count@ \space instead of \number\#3\%}
\fi
\ifnum\count@<#2\relax
\advance\count@ 1 \%
\repeat
\def\space{ }
\expandafter\ifx\csname LoadCommand\endcsname\relax
\def\LoadCommand{\input atbegshi.sty\relax}
\fi
\def\Test{\RangeCatcodeInvalid{0}{47}\%
\RangeCatcodeInvalid{58}{64}\%
\RangeCatcodeInvalid{91}{96}\%
\RangeCatcodeInvalid{123}{255}\%
\catcode`\@=12 \%
\catcode`\?=0 \%
\catcode`\%=14 \%
\LoadCommand\RangeCatcodeCheck{0}{36}{15}\%
\RangeCatcodeCheck{37}{37}{14}\%
\RangeCatcodeCheck{38}{47}{15}\%
\RangeCatcodeCheck{48}{57}{12}\%
\RangeCatcodeCheck{58}{63}{15}\%
\RangeCatcodeCheck{64}{64}{12}\%
\RangeCatcodeCheck{65}{90}{11}\%
\RangeCatcodeCheck{91}{91}{15}\%
\RangeCatcodeCheck{92}{92}{0}\%
\RangeCatcodeCheck{93}{96}{15}\%
\RangeCatcodeCheck{97}{122}{11}\%
\RangeCatcodeCheck{123}{255}{15}\%
\RestoreCatcodes}
\Test
\csname @@end\endcsname
\end
⟨/test1⟩
⟨*test2⟩
\input atbegshi.sty\relax
\def\msg#1#2{\msg{}\msg{*** Test with box (#1), expected page output [#2]\% hash-ok}
\}
\newbox\voidbox
\def\void{\box\voidbox}
\begingroup
\setbox\voidbox=\void
\endgroup
\count0=0\relax
\AtBeginShipout{\global\advance\count0 by 1\relax
\msg{* Inside \string\AtBeginShipout: \the\count0}\%}
}
Hello World
\testmsg{\string
null\space(discarded)}{}
\AtBeginShipout{%
  \msg{\string \begin{verbatim}
    * Inside \string \AtBeginShipout: DISCARD
  \end{verbatim}
  }
  \AtBeginShipoutDiscard
}
\shipout\null
\end\langle/test2\rangle
\NeedsTeXFormat{LaTeX2e}
\ProvidesFile{atbegshi-test3.tex}[2016/06/09 v1.18 Test file for LaTeX]
\RequirePackage{color}
\pagecolor{yellow}
\documentclass[a5paper,showtrims]{memoir}
\usepackage{atbegshi}
\AtBeginShipout{%
  \setbox\AtBeginShipoutBox=\\vbox{%
    \vbox to 0pt{%
      \kern-1.5in %
      \hbox to 0pt{%
        \kern-1.5in %
        \color{blue}%
        \rule{1in}{1in}%
        \hss
      }%
    }%
    \vss
  }%
  \hrule
  \hbox{\vrule\box\AtBeginShipoutBox\vrule}%
  \hrule
%
}%
\usepackage{eso-pic}
\makeatletter
\ifdefined{@EveryShipout@Init}{%
  \typeout{Test skipped}%
  \@@end%
}{}
\@EveryShipout@Init
\let\@EveryShipout@Init\relax
\makeatother
\AddToShipoutPicture{%
  \hspace{.52\paperwidth}%
  \hbox{\hspace{.48\paperwidth}%}
  \colorbox{cyan}{%
    \rule{0mm}{\paperheight}%
    \rule{0mm}{\paperheight}%
    \hspace{.48\paperwidth}%
  }%
}%
\usepackage{eso-pic}
\makeatletter
\ifdefined{@EveryShipout@Init}{%
  \typeout{Test skipped}%
  \@@end%
}{}
\@EveryShipout@Init
\let\@EveryShipout@Init\relax
\makeatother
\AddToShipoutPicture{%
  \hspace{.52\paperwidth}%
  \hbox{\hspace{.48\paperwidth}%}
  \colorbox{cyan}{%
    \rule{0mm}{\paperheight}%
    \rule{0mm}{\paperheight}%
    \hspace{.48\paperwidth}%
  }%
}%
\usepackage{eso-pic}
\makeatletter
\ifdefined{@EveryShipout@Init}{%
  \typeout{Test skipped}%
  \@@end%
}{}
\@EveryShipout@Init
\let\@EveryShipout@Init\relax
\makeatother
\AddToShipoutPicture{%
  \hspace{.52\paperwidth}%
  \hbox{\hspace{.48\paperwidth}%}
  \colorbox{cyan}{%
    \rule{0mm}{\paperheight}%
    \rule{0mm}{\paperheight}%
    \hspace{.48\paperwidth}%
  }%
}%
\usepackage{eso-pic}
\makeatletter
\ifdefined{@EveryShipout@Init}{%
  \typeout{Test skipped}%
  \@@end%
}{}
\@EveryShipout@Init
\let\@EveryShipout@Init\relax
\makeatother
\AddToShipoutPicture{%
  \hspace{.52\paperwidth}%
  \hbox{\hspace{.48\paperwidth}%}
  \colorbox{cyan}{%
    \rule{0mm}{\paperheight}%
    \rule{0mm}{\paperheight}%
    \hspace{.48\paperwidth}%
  }%
}%
\usepackage{eso-pic}
\makeatletter
\ifdefined{@EveryShipout@Init}{%
  \typeout{Test skipped}%
  \@@end%
}{}
\@EveryShipout@Init
\let\@EveryShipout@Init\relax
\makeatother
\AddToShipoutPicture{%
  \hspace{.52\paperwidth}%
  \hbox{\hspace{.48\paperwidth}%}
  \colorbox{cyan}{%
    \rule{0mm}{\paperheight}%
    \rule{0mm}{\paperheight}%
    \hspace{.48\paperwidth}%
  }%
}%
\usepackage{eso-pic}
\makeatletter
\ifdefined{@EveryShipout@Init}{%
  \typeout{Test skipped}%
  \@@end%
}{}
\@EveryShipout@Init
\let\@EveryShipout@Init\relax
\makeatother
\AddToShipoutPicture{%
  \hspace{.52\paperwidth}%
  \hbox{\hspace{.48\paperwidth}%}
  \colorbox{cyan}{%
    \rule{0mm}{\paperheight}%
    \rule{0mm}{\paperheight}%
    \hspace{.48\paperwidth}%
  }%
}%
\usepackage{eso-pic}
\makeatletter
\ifdefined{@EveryShipout@Init}{%
  \typeout{Test skipped}%
  \@@end%
}{}
\@EveryShipout@Init
\let\@EveryShipout@Init\relax
\makeatother
\AddToShipoutPicture{%
  \hspace{.52\paperwidth}%
  \hbox{\hspace{.48\paperwidth}%}
  \colorbox{cyan}{%
    \rule{0mm}{\paperheight}%
    \rule{0mm}{\paperheight}%
    \hspace{.48\paperwidth}%
  }%
}%
\usepackage{eso-pic}
\makeatletter
\ifdefined{@EveryShipout@Init}{%
  \typeout{Test skipped}%
  \@@end%
}{}
\@EveryShipout@Init
\let\@EveryShipout@Init\relax
\makeatother
\AddToShipoutPicture{%
  \hspace{.52\paperwidth}%
  \hbox{\hspace{.48\paperwidth}%}
  \colorbox{cyan}{%
    \rule{0mm}{\paperheight}%
    \rule{0mm}{\paperheight}%
    \hspace{.48\paperwidth}%
  }%
}%
\usepackage{eso-pic}
\makeatletter
\ifdefined{@EveryShipout@Init}{%
  \typeout{Test skipped}%
  \@@end%
}{}
\@EveryShipout@Init
\let\@EveryShipout@Init\relax
\makeatother
\AddToShipoutPicture{%
  \hspace{.52\paperwidth}%
  \hbox{\hspace{.48\paperwidth}%}
  \colorbox{cyan}{%
    \rule{0mm}{\paperheight}%
    \rule{0mm}{\paperheight}%
    \hspace{.48\paperwidth}%
  }%
}%
\usepackage{eso-pic}
\makeatletter
\ifdefined{@EveryShipout@Init}{%
  \typeout{Test skipped}%
  \@@end%
}{}
\@EveryShipout@Init
\let\@EveryShipout@Init\relax
\makeatother
\AddToShipoutPicture{%
  \hspace{.52\paperwidth}%
  \hbox{\hspace{.48\paperwidth}%}
  \colorbox{cyan}{%
    \rule{0mm}{\paperheight}%
    \rule{0mm}{\paperheight}%
    \hspace{.48\paperwidth}%
  }%
}%
\usepackage{eso-pic}
\makeatletter
\ifdefined{@EveryShipout@Init}{%
  \typeout{Test skipped}%
  \@@end%
}{}
\@EveryShipout@Init
\let\@EveryShipout@Init\relax
\makeatother
\AddToShipoutPicture{%
  \hspace{.52\paperwidth}%
  \hbox{\hspace{.48\paperwidth}%}
  \colorbox{cyan}{%
    \rule{0mm}{\paperheight}%
    \rule{0mm}{\paperheight}%
    \hspace{.48\paperwidth}%
  }%
}%
\usepackage{eso-pic}
\makeatletter
\ifdefined{@EveryShipout@Init}{%
  \typeout{Test skipped}%
  \@@end%
}{}
\@EveryShipout@Init
\let\@EveryShipout@Init\relax
\makeatother
\AddToShipoutPicture{%
  \hspace{.52\paperwidth}%
  \hbox{\hspace{.48\paperwidth}%}
  \colorbox{cyan}{%
    \rule{0mm}{\paperheight}%
    \rule{0mm}{\paperheight}%
    \hspace{.48\paperwidth}%
  }%
}%
Newer versions of class \texttt{memoir} emulate package \texttt{crop} and prevents its loading.
This is undone in next line for this test file.
\expandafter\let\csname ver@crop.sty\endcsname\relax
\usepackage[color=red,cross,a4,center]{crop}
\begin{document}
\shipout\null
\shipout\box\csname voidb@x\endcsname
\section{Hello World}
\end{document}
5 Installation

5.1 Download

Package. This package is available on CTAN:\footnote{\url{http://ctan.org/pkg/atbegshi}}:

- \texttt{CTAN:macros/latex/contrib/oberdiek/atbegshi.dtx} The source file.

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.

\texttt{CTAN:install/macros/latex/contrib/oberdiek.tds.zip}

\textit{TDS} refers to the standard “A Directory Structure for \TeX_X Files” (\texttt{CTAN:tds/tds.pdf}). Directories with \texttt{texmf} in their name are usually organized this way.

5.2 Bundle installation

Unpacking. Unpack the \texttt{oberdiek.tds.zip} in the TDS tree (also known as \texttt{texmf} tree) of your choice. Example (linux):

```
unzip oberdiek.tds.zip -d ~/texmf
```

Script installation. Check the directory \texttt{TDS:scripts/oberdiek/} for scripts that need further installation steps. Package \texttt{attachfile2} comes with the Perl script \texttt{pdfatfi.pl} that should be installed in such a way that it can be called as \texttt{pdfatfi}.

Example (linux):

```
chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/
```

5.3 Package installation

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

```
tex atbegshi.dtx
```

\textit{TDS}. Now the different files must be moved into the different directories in your installation TDS tree (also known as \texttt{texmf} tree):

- \texttt{atbegshi.sty} $\rightarrow$ \texttt{tex/generic/oberdiek/atbegshi.sty}
- \texttt{atbegshi.pdf} $\rightarrow$ \texttt{doc/latex/oberdiek/atbegshi.pdf}
- \texttt{atbegshi-example1.tex} $\rightarrow$ \texttt{doc/latex/oberdiek/atbegshi-example1.tex}
- \texttt{atbegshi-example2.tex} $\rightarrow$ \texttt{doc/latex/oberdiek/atbegshi-example2.tex}
- \texttt{test/atbegshi-test1.tex} $\rightarrow$ \texttt{doc/latex/oberdiek/test/atbegshi-test1.tex}
- \texttt{test/atbegshi-test2.tex} $\rightarrow$ \texttt{doc/latex/oberdiek/test/atbegshi-test2.tex}
- \texttt{test/atbegshi-test3.tex} $\rightarrow$ \texttt{doc/latex/oberdiek/test/atbegshi-test3.tex}
- \texttt{atbegshi.dtx} $\rightarrow$ \texttt{source/latex/oberdiek/atbegshi.dtx}

If you have a \texttt{docstrip.cfg} that configures and enables \texttt{docstrip}’s TDS installing feature, then some files can already be in the right place, see the documentation of \texttt{docstrip}.

5.4 Refresh file name databases

If your \TeX_X distribution (\texttt{te\TeX}, \texttt{mik\TeX}, …) relies on file name databases, you must refresh these. For example, \texttt{te\TeX} users run \texttt{texhash} or \texttt{mktexlatex}.

\footnote{\url{http://ctan.org/pkg/atbegshi}}
5.5 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:

\begin{verbatim}
latex \let\install=y\input{atbegshi.dtx}
\end{verbatim}

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:

\begin{verbatim}
\PassOptionsToClass{a4paper}{article}
\end{verbatim}

An example follows how to generate the documentation with pdf\LaTeX:

\begin{verbatim}
pdflatex atbegshi.dtx
makeindex -s gind.ist atbegshi.idx
pdflatex atbegshi.dtx
makeindex -s gind.ist atbegshi.idx
pdflatex atbegshi.dtx
\end{verbatim}

6 Catalogue

The following XML file can be used as source for the \TeX{} Catalogue. The elements \texttt{caption} and \texttt{description} are imported from the original XML file from the Catalogue. The name of the XML file in the Catalogue is \texttt{atbegshi.xml}.

\begin{verbatim}
<entry datestamp='$Date$' modifier='$Author$' id='atbegshi'>
  <name>atbegshi</name>
  <caption>Execute stuff at \textout time.</caption>
  <authorref id='auth:oberdiek'/>
  <copyright owner='Heiko Oberdiek' year='2007-2011'/>
  <license type='lppl1.3'/>
  <version number='1.18'/>
  <description>
    This package is a modern reimplementation of package
    \texttt{everyshi}, providing various commands to be executed before a \texttt{\textout} command. It makes use of \TeX's facilities if they are available. The package may be used either with \LaTeX{} or with plain TeX.
    \p
    The package is part of the \texttt{oberdiek}/\texttt{atbegshi} bundle.
  </description>
  <documentation details='Package documentation' href='ctan:/macros/latex/contrib/oberdiek/atbegshi.pdf'/>
  <ctan file='true' path='/macros/latex/contrib/oberdiek/atbegshi.dtx'/>
  <miktex location='oberdiek'/>
  <texlive location='oberdiek'/>
  <install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip'/>
</entry>
\end{verbatim}
7 History

[2007/04/17 v1.0]
- First version.

[2007/04/18 v1.1]
- New method based on `\lastkern` is used if `\e-\TeX` is missing.
- `\AtBeginShipoutDiscard` also resets `\deadcycles`.

[2007/04/19 v1.2]
- `\AtBeginShipoutEarly` removed for simplification reasons.
- Forgotten definition of `\AtBeginShipoutInfo` added.
- Patches for packages `crop` and `everyshi` and class `memoir` added.

[2007/04/26 v1.3]
- Use of package `infwarerr`.
- Catcode section after generic header.

[2007/04/27 v1.4]
- Small optimizations.

[2007/06/06 v1.5]
- `\AtBeginShipoutUpperLeft` added.
- Example added.
- Fix in second test file for newer version of `memoir`.

[2007/09/09 v1.6]
- Catcode section rewritten.

[2008/07/18 v1.7]
- Documentation of `\AtBeginShipoutUpperLeft` fixed and extended.

[2008/07/19 v1.8]
- `\AtBeginShipoutUpperLeftForeground` added.

[2008/07/31 v1.9]
- Second example (TrimBox for dvipdfmx) added.
- No changes in package code.

[2009/12/02 v1.10]
- `\AtBeginShipoutOriginalShipout` added.
- Test file fixed.
[2010/03/01 v1.11]
- Compatibility with ini-TeX except for `\newbox`.

[2010/03/25 v1.12]
- `\AtBeginShipoutNext` can now be used inside `\AtBeginShipoutNext`.

[2010/08/18 v1.13]
- Fixes for `\AtBegShi@CheckDefinable`.

[2010/12/02 v1.14]
- Remove the warning because of void box if the hook calls `. `

[2011/01/30 v1.15]
- Already loaded package files are not input in plain TeX.

[2011/10/05 v1.16]
- `\AtBeginShipoutAddToBox`, `\AtBeginShipoutAddToBoxForeground` added.
- `\AtBeginShipoutBoxWidth`, `\AtBeginShipoutBoxHeight`, `\AtBeginShipoutBoxDepth` added.
- Updates for patches of class memoir.

[2016/05/16 v1.17]
- Documentation updates.

[2016/06/09 v1.18]
- Update for `\pdfhorign` in new LuaTeX.

8 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.

<table>
<thead>
<tr>
<th>Symbols</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>882</td>
</tr>
<tr>
<td>%</td>
<td>958</td>
</tr>
<tr>
<td>@</td>
<td>833, 956</td>
</tr>
<tr>
<td>@end</td>
<td>1101</td>
</tr>
<tr>
<td>@EveryShipout@Init</td>
<td>1103, 1104</td>
</tr>
<tr>
<td>@EveryShipout@Org@Shipout</td>
<td>605, 606, 845, 846</td>
</tr>
<tr>
<td>@PackageError</td>
<td>665, 675, 702, 705</td>
</tr>
<tr>
<td>@PackageWarning</td>
<td>238, 300, 326</td>
</tr>
<tr>
<td>@cclv</td>
<td>258, 590, 666, 669, 735</td>
</tr>
<tr>
<td>{</td>
<td>880</td>
</tr>
<tr>
<td>}</td>
<td>881</td>
</tr>
</tbody>
</table>

33
AtBegShi@FirstDisabled 282, 299
AtBegShi@Everyshi@Output 668, 675
AtBegShi@FirstDisabled 282, 299
AtBegShi@Everyshi@shipout 710, 712
AtBegShi@Everyshi@shipout 664, 677, 679, 681, 683, 685, 687, 692
AtBegShi@Everyshi@Test 668, 675
AtBegShi@First 255, 276
AtBegShi@FirstDisabled 282, 299
AtBegShi@GetBoxSize 257, 260, 266, 642, 770, 812
AtBegShi@GroupLevel 222, 229, 618, 623, 701, 706, 852, 857
AtBegShi@Hook 243, 284, 289
AtBegShi@HookFirst 277, 279, 286, 297
AtBegShi@HookNext 244, 246, 285, 293
AtBegShi@horigin 490, 495, 541, 553
AtBegShi@Item 317, 319
AtBegShi@Memoir@PatchAB 759, 822, 825
AtBegShi@Memoir@PatchCD 802, 828, 831
AtBegShi@Memoir@PatchX 822, 825, 828, 831, 863
AtBegShi@Memoir@ship 737, 818
AtBegShi@Memoir@shipiiA 743, 820
AtBegShi@Memoir@shipiiB 749, 823
AtBegShi@Memoir@shipiiC 776, 826
AtBegShi@Memoir@shipiiD 787, 829
AtBegShi@Memoir@shipout 733, 835, 839, 841, 843, 845
AtBegShi@OrgProtect 240, 253, 256
AtBegShi@Output 232, 234
AtBegShi@PatchCrop 573, 655, 658, 660
AtBegShi@PatchEveryshi 662, 724, 727, 729
AtBegShi@PatchMemoir 731, 870, 873, 875
AtBegShi@restore 358, 367, 377, 380, 396, 410, 417, 428, 438, 441, 457, 472
AtBegShi@restороbox 385, 411, 446, 473
AtBegShi@Shipout 218, 346
AtBegShi@Text 223, 227
AtBegShi@vorigin 492, 496, 543, 555
AtNextShipout 260
\baselineskip 403, 464, 588, 632
\begin 11, 50, 1115
\body 900, 904
\box 47, 251, 262, 590, 634, 644, 715, 751, 767, 772, 778, 783, 789, 795, 798, 809, 814, 988, 1021, 1022, 1024, 1025, 1036, 1093, 1117
\catcode 57, 58, 60, 61, 62, 63, 64, 65, 66, 67, 68, 88, 89, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 124, 125, 127, 128, 129, 133, 134, 135, 136, 137, 138, 139, 142, 143, 145, 146, 147, 148, 152, 154, 212, 213, 215, 216,