Abstract

This package is a modern reimplementation of package 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 crop .................................. 20
      3.6.2 Package everyshi ................................ 22
      3.6.3 Class memoir .................................. 23

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

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

Package `atbegshi` redefines \texttt{\textbackslash 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 \texttt{\textbackslash AtBeginShipout}

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

3. A hook that is only executed for the first page, see \texttt{\textbackslash 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{⟨code⟩}
\AtBeginShipoutBox
\End
\end{verbatim}

Execute the \texttt{⟨code⟩} for every page. The page contents is held in box register \texttt{\textbackslash AtBeginShipoutBox} and may be modified. Use \texttt{\textbackslash AtBeginShipoutDiscard} if you want to discard the page.

\textit{Note:} Package \texttt{everyshi} uses box register 255. With package \texttt{atbegshi} you must use \texttt{\textbackslash AtBeginShipoutBox} instead.
If \LaTeX{} calls \texttt{\textbackslash shipout} in \texttt{\textbackslash outputpage} (part of its output routine), the meaning of \texttt{\protect} is \texttt{\noexpand}. \LaTeX{} sets \texttt{\protect} to the appropriate \texttt{\@typeset\protect} in the box that is shipped out. This is too late for the hooks, they are called earlier in the redefined \texttt{\shipout}. Therefore package \texttt{atbegshi} sets \texttt{\protect} to \texttt{\@typeset\protect} before it calls the hooks. (In \texttt{\textbackslash EveryShipout} of package \texttt{everyshi} the user is responsible for the correct setting of \texttt{\protect}.)

\begin{verbatim}
\AtBeginShipoutNext {⟨code⟩}
\end{verbatim}

This reimplements package \texttt{everyshi}'s \texttt{\textbackslash AtNextShipout}. The ⟨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}{}\
\end{verbatim}

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

\begin{verbatim}
\AtBeginShipoutFirst {⟨code⟩}
\end{verbatim}

This reimplements \LaTeX{}'s \texttt{\textbackslash AtBeginDvi}. This hook is usually used for \texttt{\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.

\begin{verbatim}
\AtBeginShipoutDiscard
\end{verbatim}

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{\shipout} would do.

\begin{verbatim}
\AtBeginShipoutInit
\end{verbatim}

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

\begin{verbatim}
\AtBeginShipoutAddToBox {⟨stuff⟩}\\
\AtBeginShipoutAddToBoxForeground {⟨stuff⟩}
\end{verbatim}

A quite common use case is the addition of \texttt{\special} or other whatstos to the page output box. Macro \texttt{\AtBeginShipoutAddToBox} puts ⟨stuff⟩ in a box with zeroed dimensions. The box with the ⟨stuff⟩ is put in the upper left corner of the shipout box \texttt{\AtBeginShipoutBox}. Macro \texttt{\AtBeginShipoutAddToBox} puts the ⟨stuff⟩ in the background, the other 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 ⟨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 \texttt{pdf\LaTeX}. 
\AtBeginShipoutUpperLeft \{\textit{background material}\}\}

This is a macro that puts material in the background of box \AtBeginShipoutBox. The \textit{background material} is set in an \texttt{hbox}, the reference point is the upper left corner of the output page. In case of pdf\TeX in PDF mode, the settings of \texttt{pdfhorigin} and \texttt{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 \textit{background material} is set inside a \texttt{picture} environment:

\begin{verbatim}
\begin{picture}(0,0)
  \setlength{\unitlength}{1pt}\
  \put(0.5\paperwidth,-0.5\paperheight){\circle{10}}
\end{picture}
\end{verbatim}

\AtBeginShipoutUpperLeftForeground \{\textit{foreground material}\}\}

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

\AtBeginShipoutOriginalShipout \{\textit{box}\}\}

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, X\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.

\begin{verbatim}
\documentclass[a4paper]{article}
\usepackage{color}
\usepackage{atbegshi}
\usepackage{picture}
\AtBeginShipout{\AtBeginShipoutUpperLeft{\put(0.5\paperwidth,-0.5\paperheight){\circle{10}}}}
\end{verbatim}

Package \texttt{picture} makes life a little easier, because we can now also use length specifications in \texttt{picture}'s commands.

\begin{verbatim}
\AtBeginShipout{\AtBeginShipoutUpperLeft{\put(0.5\paperwidth,-0.5\paperheight){\circle{10}}}}
\end{verbatim}

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.

\begin{verbatim}
\AtBeginShipout{}
\AtBeginShipoutUpperLeft{\put(0.5\paperwidth,-0.5\paperheight){\circle{10}}}
\end{verbatim}
\begin{document}
\section{Hello World}
\newpage
\AtBeginShipoutNext{\AtBeginShipoutUpperLeft{\color{red}\put(0,-0.5\paperheight){\line(1,0){\paperwidth}}\put(0.5\paperwidth, 0){\line(0,-1){\paperheight}}}}
\onlyonthispage{We add a red cross.}
\newpage
\Thispagehas{The circle only.}
\vspace{\fill}
\thenextpage{will be discarded.}
\newpage
\AtBeginShipoutNext{\AtBeginShipoutDiscard}
\thispageis{discarded.}
\newpage
\thelastpage{}\end{document}

\section{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.

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

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

2.1 $\texttt{\textbackslash shipout}$

The TeX primitive command $\texttt{\textbackslash shipout}$ 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. $\texttt{\textbackslash shipout}\hbox{Hello World}$.

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 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 not void.

2.2 $\texttt{\textbackslash afterassignment}$

We want to overload $\texttt{\textbackslash shipout}$ 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:

$\texttt{\textbackslash shipout}\texttt{\null}$
$\texttt{\textbackslash shipout}\texttt{\vbox{...}}$
$\texttt{\textbackslash shipout}\texttt{\vtop}\texttt{\bgroup ...\egroup}$
$\texttt{\textbackslash shipout}\texttt{\box255}$

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

$\texttt{\textbackslash shipout}\texttt{\hbox{\bgroup}}$
$\texttt{\textbackslash shipout}\texttt{\vbox{\egroup}}$

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

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

$\texttt{\textbackslash shipout := \textbackslash afterassignment\@test}$
$\texttt{\setbox\mybox=}$

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

$\texttt{\textbackslash shipout}\texttt{\box0} \Rightarrow \texttt{\setbox\mybox=\box0 \@test}$

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:

$\texttt{\textbackslash shipout}\texttt{\hbox{Hello World}}$
$\Rightarrow \texttt{\setbox\mybox=\hbox{\@test Hello World}}$
2.3 Test for direct or indirect boxes

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

2.3.1 With $\varepsilon$-\LaTeX

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

\begin{verbatim}
\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
\end{verbatim}

2.3.2 Without $\varepsilon$-\LaTeX

Life becomes complicated without $\varepsilon$-\LaTeX. We cannot ask the group level. However, if we are inside a direct box, the box register \texttt{\mybox} is not yet changed by \texttt{\setbox}. Thus we need a special initial value and compare it in \texttt{@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: \texttt{\ifhbox, \ifvbox}
- Dimensions: \texttt{\wd, \ht, \dp}
- Voidness: \texttt{\ifvoid}

Unhappily all these qualities even combined are not sufficient for constructing an initial box value, because \texttt{\shipout} can be called with a box that is accidentally 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 \texttt{4911sp-\maxdimen}.
- A void box.

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

- \texttt{\shipout} is used in a group that perhaps closes some pages later. A bad place for \texttt{@output}.
- Without a surrounding group \texttt{\aftergroup} effectively kills its argument.
In the first case of a box with special dimensions we can even lose 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}
\shipout :=
\% 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{\shipout\box\voidbox} 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}
\shipout :=
\begingroup
\setbox\mybox=\hbox\bgroup
\kern 1pt
\afterassignment\shipout@test
\global\setbox\mybox=
\@test :=
\ifdim\lastkern=0pt
\% direct box
\aftergroup\egroup
\aftergroup\endgroup
\aftergroup\@output
\else
\egroup
\endgroup
\@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 \texttt{\$-\LaTeX} we ensure the original behaviour of \texttt{\stringshipout} 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{\stringmybox}. The following case that outputs the same page twice shows that we are not free in the use of the box register:

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

We manipulate the box by the hook and without \texttt{\$-\LaTeX} 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{\stringshipout} at all. In fact \texttt{\stringshipout} does not change any box register. This is usually done by a call of \texttt{\stringbox}, but the output routine can do it later after invoking of \texttt{\stringshipout}.

2.6 Summary

2.6.1 With \texttt{\$-\LaTeX} X

Putting the pieces together we get for \texttt{\$-\LaTeX}:

\begin{verbatim}
\newbox\mybox \\
\let\original@shipout\shipout \\
\stringshipout := \\
edef\saved@grouplevel{\number\currentgrouplevel} \\
\afterassignment\@test \\
\setbox\mybox= \\
\@test := \\
\ifnum\saved@grouplevel<\currentgrouplevel \\
\expandafter\aftergroup \\
\fi \\
\@output \\
\@output := \\
@ifvoid\mybox \\
% cancel output of void box \\
\else \\
\@hook \\
@ifvoid\mybox
\end{verbatim}
2.6.2 Without $\varepsilon$-TeX, traditional way

And for $\TeX$ without $\varepsilon$-TeX:
\begin{verbatim}
\newbox\mybox
\begingroup
 \setbox\mybox=\box\mybox % ensure \mybox is void
\endgroup
\let\original@shipout\shipout
\shipout :=
 % trick to get a void box \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
\else
 \@hook
 \ifvoid\mybox
 % user code in \@hook could have voided the box
\else
 \original@shipout\box\mybox
\fi
\fi
\fi
\end{verbatim}

2.6.3 $\lastkern$ method

And for $\TeX$ without $\varepsilon$-TeX using the $\lastkern$ method:
\begin{verbatim}
\newbox\mybox
\let\original@shipout\shipout
\shipout :=
 \begingroup
 \setbox\mybox=\hbox\bgroup
 \kern1pt
 \afterassignment\@test
\setbox\mybox=
\@test :=
\ifdim\lastkern=0pt
\expandafter\aftergroup
\fi
\@output
\@output :=
\end{verbatim}
3 Implementation

Package \texttt{atbegshi} uses \texttt{\LaTeX}'s \texttt{\currentgrouplevel}, if it is available. Otherwise the \texttt{\lastkern} method is used.

\section{Reload check and package identification}

Reload check, especially if the package is not used with \texttt{\LaTeX}.

Package identification:
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
}
\TMP@RequirePackage{infwarerr}[2007/09/09]%
\TMP@RequirePackage{ltxcmds}[2010/03/01]%
\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
\else
\ltx@newif\ifAtBegShi@Discarded
\AtBeginShipoutDiscard
\AtBegShi@CheckDefinable\AtBeginShipoutDiscard
\def\AtBeginShipoutDiscard{%
\@PackageError{atbegshi}{%}
\string#1\space is already defined%
}\@ehd
\fi
\fi
\ifAtBegShi@Discarded
\ltx@newif\ifAtBegShi@Discarded
\AtBeginShipoutDiscard
\AtBegShi@CheckDefinable\AtBeginShipoutDiscard
\def\AtBeginShipoutDiscard{%
\@PackageError{atbegshi}{%}
\string#1\space is already defined%
}\@ehd
\fi
\fi

\AtBeginShipoutDiscard
\def\AtBeginShipoutDiscard{%
\@PackageError{atbegshi}{%}
\string#1\space is already defined%
}\@ehd
\fi
\fi
\ifAtBegShi@Discarded
\ltx@newif\ifAtBegShi@Discarded
\AtBeginShipoutDiscard
\AtBegShi@CheckDefinable\AtBeginShipoutDiscard
\def\AtBeginShipoutDiscard{%
\@PackageError{atbegshi}{%}
\string#1\space is already defined%
}\@ehd
\fi
\fi

\AtBeginShipoutDiscard
\def\AtBeginShipoutDiscard{%
\@PackageError{atbegshi}{%}
\string#1\space is already defined%
}\@ehd
\fi
\fi

\AtBeginShipoutDiscard
\def\AtBeginShipoutDiscard{%
\@PackageError{atbegshi}{%}
\string#1\space is already defined%
}\@ehd
\fi
\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{%
\begingroup
\setbox\AtBeginShipoutBox=\hbox\bgroup
\kern\p@
\edef\AtBegShi@GroupLevel{\number\currentgrouplevel}%
\afterassignment\AtBegShi@Test
\global
\setbox\AtBeginShipoutBox=%
}
\AtBegShi@Test
\def\AtBegShi@Test{%
\ifdim\lastkern=0pt %
  \ifnum\AtBegShi@GroupLevel<\currentgrouplevel
    \aftergroup
  \fi
  \AtBegShi@Output
\fi
\AtBegShi@Output
\def\AtBegShi@Output{%
\egroup
\endgroup
\ifvoid\AtBeginShipoutBox
  \@PackageWarning{atbegshi}{Ignoring void shipout box}%
\else
  \let\AtBegShi@OrgProtect\protect
  \csname set@typeset@protect\endcsname
  \global\AtBegShi@Discardedfalse
  \AtBegShi@Hook
  \ifAtBegShi@Discarded
    \@PackageInfoNoLine{atbegshi}{Shipout page discarded}%
  \global\AtBegShi@Discardedfalse
  \begingroup
  \setbox\AtBeginShipoutBox\box\AtBeginShipoutBox
  \let\protect\AtBegShi@OrgProtect
  \else
    \AtBegShi@First
    \if\AtBegShi@OrgProtect
    \AtBegShi@GetBoxSize\AtBeginShipoutBox
    \ltx@ifundefined{AtNextShipout}{%
    \global\AtBegShi@Discardedfalse
    \begingroup
    \setbox\AtBeginShipoutBox\box\AtBeginShipoutBox
    \endgroup
    \let\protect\AtBegShi@OrgProtect
  \else
    %
    \AtNextShipout{\AtBegShi@GetBoxSize@cclv}%
  )%
  \AtBeginShipoutOriginalShipout\box\AtBeginShipoutBox
\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
}\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
\AtBegShi@FirstDisabled
\long\def\AtBegShi@FirstDisabled#1{%
  \PackageWarning{atbegshi}{First page is already shipped out, ignoring}%
  \string\AtBeginShipoutFirst
  \MessageBreak
  \string\AtBeginShipoutFirst
  \MessageBreak
  \string\AtBeginShipoutFirst
}\%
\AtBegShi@AddTo
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname g@addto@macro\endcsname\relax
\long\def\AtBegShi@AddTo#1#2{%
  \begingroup
  \toks\ltx@zero\expandafter{#1#2}
  \xdef#1{\the\toks\ltx@zero}
  \endgroup
}\%
\else
  \let\AtBegShi@AddTo\g@addto@macro
\fi
\AtBegShi@AddHook
\long\def\AtBegShi@AddHook#1#2{%
  \AtBegShi@AddTo#1{\AtBegShi@Item{#2}}
}\%
\AtBegShi@Item
\long\def\AtBegShi@Item#1{%
  \ifAtBegShi@Discarded
    #1%
  \else
    \ifvoid\AtBeginShipoutBox
      \PackageWarning{atbegshi}{Shipout box was voided by hook, ignoring shipout box}%
    \else
      \AtBeginShipoutDiscard
    \fi
  \fi
}\%
\fi
\fi
\fi
\AtBeginShipoutInit
\AtBegSh@CheckDefinable\AtBeginShipoutInit
\def\AtBeginShipoutInit{%
  \ltex@IfUndefined{newbox}{%
    \PackageError{atbegshi}{\string\AtBeginShipoutInit\space failed\MessageBreak
    because of missing \expandafter\string\cname newbox\endcsname}
  }{%}
  \AtBeginShipoutInit
}\%
\AtBegSh@CheckDefinable\AtBeginShipoutOriginalShipout
\global\let\AtBeginShipoutOriginalShipout
\global\let\shipout\AtBegShi@Shipout
\global\let\AtBeginShipoutBox\AtBeginShipoutBox
\AtBegSh@CheckDefinable\AtBeginShipoutInit
\gdef\AtBeginShipoutInit{%
\begingroup\expandafter\expandafter\expandafter\endgroup

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

\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{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
1%
\else
0%
\fi
\fi
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
}\else
\endgroup
\def\AtBegShi@BeginPicture{%
\setbox\ltx@zero=\hbox\bgroup
\begingroup
\endgroup
\expandafter\ifx\csname picture\endcsname\relax
1%
\else
1%
\else
0%
\fi
\fi
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\endpicture
\endgroup
\def\AtBegShi@BeginPicture{%
\begroup
3.6 Patches

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

3.6.1 Package \texttt{crop}

Fix of method and box.
\setbox\@cclv=%
\def\AtBegShi@Crop@ship{%
\ifvoid\@cclv
\expandafter\aftergroup
\fi
\CROP@ship}
\def\AtBegShi@Crop@shiplist{%
\lineskip\z@
\lineskiplimit\z@
\baselineskip\z@
\CROP@kernel
\box\@cclv
}
\def\AtBegShi@Crop@@ship{%
\CROP@shipout\vbox{%
\CROP@shiplist
}
}
\ifx\AtBegShi@Crop@ship\CROP@ship
\ifx\AtBegShi@Crop@shiplist\CROP@shiplist
\ifx\AtBegShi@Crop@@ship\CROP@@ship
\let\AtBegShi@found\relax
\ifx\shipout\AtBegShi@Crop@shipout
\def\AtBegShi@found{\shipout}%
\else
\ifx\AtBeginShipoutOriginalShipout\AtBegShi@Crop@shipout
\def\AtBegShi@found{\AtBeginShipoutOriginalShipout}%
\else
\ifx\@EveryShipout@Org@Shipout\AtBegShi@Crop@shipout
\def\AtBegShi@found{\@EveryShipout@Org@Shipout}%
\else
\ifx\GPTorg@shipout\AtBegShi@Crop@shipout
\def\AtBegShi@found{\GPTorg@shipout}%
\else
\ifx\THBorg@shipout\AtBegShi@Crop@shipout
\def\AtBegShi@found{\THBorg@shipout}%
\else
\ifx\mem@oldshipout\AtBegShi@Crop@shipout
\def\AtBegShi@found{\mem@oldshipout}%
\else
\fi
\fi
\fi
\fi
\fi
\fi
\expandafter\endgroup
\expandafter\def\AtBegShi@found{%
\edef\AtBegShi@GroupLevel{\number\currentgrouplevel}%
\afterassignment\CROP@ship
\setbox\AtBeginShipoutBox=%
}
\def\CROP@ship{%
\ifnum\AtBegShi@GroupLevel=\currentgrouplevel%
\afterassignment\CROP@ship
\setbox\AtBeginShipoutBox=%
}
\def\CROP@shiplist{%
\lineskip 0pt\relax
\lineskiplimit 0pt\relax
\baselineskip 0pt\relax
\CROP@kernel
\box\AtBeginShipoutBox
}
\def\CROP@@ship{%
\ifvoid\AtBeginShipoutBox%
\else
\expandafter\endgroup
\expandafter\def\AtBegShi@GroupLevel{\number\currentgrouplevel}%
\else
\expandafter\aftergroup
\fi
\CROP@ship%
\def\CROP@ship{%
\lineskip 0pt\relax
\lineskiplimit 0pt\relax
\baselineskip 0pt\relax
\CROP@kernel
\box\AtBeginShipoutBox
}
\def\CROP@ship{%
\ifvoid\AtBeginShipoutBox%
\else
\expandafter\endgroup
\expandafter\def\AtBegShi@GroupLevel{\number\currentgrouplevel}%
\else
\expandafter\aftergroup
\fi
}\CROP@ship%
3.6.2 Package everyshi

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

```latex
\def\AtBeginShipout\Everyshi{}\begingroup
\long\def\AtBegin\Everyshi\out\{}\afterassignment\@EveryShipout\test
\global\setbox\@cclv= %
\long\def\AtBegin\Everyshi\test\{}\ifvoid\@cclv\relax
\aftergroup\@EveryShipout\output
\else
\@EveryShipout\output
\fi
\ifx\AtBegin\Everyshi\test\@EveryShipout\test
\let\AtBegin\Everyshi\out\relax
\if\AtBegin\Everyshi\out\%\else\if\everyshi\AtBegin\Everyshi\out\%\else\if\GPTorg\AtBegin\Everyshi\out\%\else\if\THBorg\AtBegin\Everyshi\out\%\else\if\mem\AtBegin\Everyshi\out\%\else\if\@EveryShipout\out\%\else\if\everyshi\AtBegin\Everyshi\out\%\else\if\GPTorg\AtBegin\Everyshi\out\%\else\if\THBorg\AtBegin\Everyshi\out\%\else\if\mem\AtBegin\Everyshi\out\%\else\endgroup
\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi
```
3.6.3 Class memoir

Fix of method and box.

```latex
\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{\mem@oldshipout\vbox{\trimmarks
\ifvoid\@cclv\else\mem@oldshipout\vbox{\trimmarks
\ifvoid\@cclv\else\mem@oldshipout\vbox{\trimmarks}}}}%
\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
    \fi
  \fi
}\%
4 Test

4.1 Catcode checks for loading

\catcode\{=1 %
\catcode\}=2 %
\catcode\#=6 %
\catcode\@=11 %
\expandafter\ifx\csname count@\endcsname\relax
\countdef\count@=255 %
\fi
\expandafter\ifx\csname @gobble\endcsname\relax
\long\def\@gobble#1{}%
\fi
\expandafter\ifx\csname @firstofone\endcsname\relax
\long\def\@firstofone#1{#1}%
\fi
\expandafter\ifx\csname loop\endcsname\relax
\else
\expandafter\@gobble
\fi
{%
\def\loop#1\repeat{%
\def\body{#1}%
\iterate
}%
\def\iterate{%
\body
\let\next\iterate
\else
\let\next\relax
\fi
\next
}%
\let\repeat=\fi
}%
\def\RestoreCatcodes{}
\count@=0 %
\loop
\edef\RestoreCatcodes{%
\RestoreCatcodes
\catcode\the\count@=\the\catcode\count@\relax
}%
\ifnum\count@<255 %
\advance\count@1 %
\repeat
\def\RangeCatcodeInvalid#1#2{%
\count@=#1\relax
\loop
\catcode\count@=15 %
\ifnum\count@<#2\relax
\advance\count@1 %
\repeat
\def\RangeCatcodeCheck#1#2#3{%
\count@=#1\relax
\loop
\ifnum\count@=#2\relax
\advance\count@1 %
\repeat
\def\RangeCatcodeCheck#1#2#3{%
\count@=#1\relax
\loop
\ifnum\count@=#2\relax
\advance\count@1 %
\repeat
\AtBeginShipoutFirst{%
\msg{** Inside \string\AtBeginShipoutFirst}%
Hello World%
}
\testmsg{\string\null}{1}
\shipout\null
\AtBeginShipoutFirst{%
This is too late%
}
\testmsg{void}{}
\shipout\void
\AtBeginShipoutFirst{%
This is too late%
}
\testmsg{void}{2}
\shipout\void
\testmsg{\string\copy255 (not void)}{2}
\setbox255\hbox{\vrule height 10bp width 10bp}
\shipout\copy255 
\testmsg{\string\copy255 (again)}{3}
\shipout\copy255 
\testmsg{\string\box255}{4}
\shipout\box255 
\testmsg{\string\box255 (again)}{}
\shipout\box255 
\testmsg{\string\hbox}{5}
\shipout\hbox{\vrule height 5bp width 20bp}
\testmsg{\string\vbox}{6}
\shipout\vbox{\hrule height 20bp width 5bp}
\testmsg{\string\null, voided by hook}{}
\def\VoidBox{%
\begingroup
\setbox\AtBeginShipoutBox=\box\AtBeginShipoutBox
\endgroup
}
\AtBeginShipout\VoidBox
\shipout\null
\def\VoidBox{}
\testmsg{output routine}{9}
Hello World
\testmsg{\null\space(discarded)}{}
\AtBeginShipout{%\msg{* Inside \string\AtBeginShipout: DISCARD}\}%\AtBeginShipoutDiscard
\AtBeginShipout{\null}
\shipout\null
\end
⟨/\test2⟩
\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{%\null\vbox{\null\vss\hrule\hbox{\vrule\box\AtBeginShipoutBox\vrule}\hrule}\null}
\usepackage{eso-pic}
\makeatletter
\ifundefined{@EveryShipout@Init}{{\typeout{Test skipped}}\@@end}{}
\AtEveryShipout@Init\let\@EveryShipout@Init\relax\makeatother
\AddToShipoutPicture{%\hspace{.52\paperwidth}\colorbox{cyan}{\rule{0mm}{\paperheight}\hspace{.48\paperwidth}}}%\usepackage{eso-pic}\makeatletter\ifdefined{@EveryShipout@Init}{}\let{@EveryShipout@Init}@end\relax\makeatother\AddToShipoutPicture{%\hspace{.52\paperwidth}\colorbox{cyan}{\rule{0mm}{\paperheight}}}%\usepackage{eso-pic}
\makeatletter\ifdefined{@EveryShipout@Init}{}\let{@EveryShipout@Init}@end\relax\makeatother\AddToShipoutPicture{%\hspace{.52\paperwidth}\colorbox{cyan}}%
\begin{document}
\shipout\null
\shipout\box{\csname voidb@x\endcsname}
\section{Hello World}
\end{document}
⟨/\test3⟩

Newer versions of class \textit{memoir} emulate package \textit{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}
⟨/\test3⟩
5 Installation

5.1 Download

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


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: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):

\begin{verbatim}
unzip oberdiek.tds.zip -d ~/texmf
\end{verbatim}

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):

\begin{verbatim}
chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/
\end{verbatim}

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:

\begin{verbatim}
tex atbegshi.dtx
\end{verbatim}

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

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

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\ 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{mktexlar}.

\footnote{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:

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

Do not forget to quote the argument according to the demands of your shell.

Generating the documentation. You can use both the .dtx or the .drv to generate the documentation. The process can be configured by the configuration file ltxdoc.cfg. For instance, put this line into this file, if you want to have A4 as paper format:

\PassOptionsToClass{a4paper}{article}

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

pdflatex atbegshi.dtx
makeindex -s gind.ist atbegshi.idx
pdflatex atbegshi.dtx
makeindex -s gind.ist atbegshi.idx
pdflatex atbegshi.dtx

6 Catalogue

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

```
<?xml version='1.0' encoding='us-ascii'?>
<!DOCTYPE entry SYSTEM 'catalogue.dtd'>
<entry datestamp='$Date$' modifier='$Author$' id='atbegshi'>
  <name>atbegshi</name>
  <caption>Execute stuff at \shipout 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
  \xref{everyshi}, providing various commands
  to be executed before a \tt\shipout\tt command. It makes use of
  e-\TeX\&\#x2019;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 \xref{oberdiek}\bundle.
  </description>
  <documentation
```
7 History

[2007/04/17 v1.0]
• First version.

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

[2007/04/19 v1.2]
• \texttt{AtBeginShipoutEarly} removed for simplification reasons.
• Forgotten definition of \texttt{AtBegShi@Info} added.
• Patches for packages \texttt{crop} and \texttt{everyshi} and class \texttt{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]
• \texttt{AtBeginShipoutUpperLeft} added.
• Example added.
• Fix in second test file for newer version of \texttt{memoir}.

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

[2008/07/18 v1.7]
• Documentation of \texttt{AtBeginShipoutUpperLeft} fixed and extended.

[2008/07/19 v1.8]
• \texttt{AtBeginShipoutUpperLeftForeground} added.

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

[2009/12/02 v1.10]
• \texttt{AtBeginShipoutOriginalShipout} added.
• Test file fixed.
• Compatibility with ini-Tex except for \newbox.

\AtBeginShipoutNext can now be used inside \AtBeginShipoutNext.

• Fixes for \AtBegShi@CheckDefinable.

• Remove the warning because of void box if the hook calls .

• Already loaded package files are not input in plain TEx.

• \AtBeginShipoutAddToBox, \AtBeginShipoutAddToBoxForeground added.

• \AtBeginShipoutBoxWidth, \AtBeginShipoutBoxHeight, \AtBeginShipoutBoxDepth added.

• Updates for patches of class memoir.

• Documentation updates.

• 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.

Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>882</td>
</tr>
<tr>
<td>%</td>
<td>958</td>
</tr>
<tr>
<td>@</td>
<td>883, 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>@EveryShipout@Output</td>
<td>670, 672, 716</td>
</tr>
<tr>
<td>@EveryShipout@Shipout</td>
<td>692, 693</td>
</tr>
<tr>
<td>@EveryShipout@Test</td>
<td>665, 675, 702, 705</td>
</tr>
<tr>
<td>@PackageError</td>
<td>194, 338</td>
</tr>
<tr>
<td>@PackageInfoNoLine</td>
<td>248, 648, 719, 864</td>
</tr>
<tr>
<td>@PackageWarning</td>
<td>238, 300, 326</td>
</tr>
<tr>
<td>@cclv</td>
<td>260, 577, 580, 590, 666, 669, 735,</td>
</tr>
<tr>
<td>@ehc</td>
<td>341</td>
</tr>
<tr>
<td>@ehd</td>
<td>196</td>
</tr>
<tr>
<td>@firstofone</td>
<td>891, 894</td>
</tr>
<tr>
<td>@gobble</td>
<td>888, 896</td>
</tr>
<tr>
<td>@ifclassloaded</td>
<td>872</td>
</tr>
<tr>
<td>@ifdefined</td>
<td>201</td>
</tr>
<tr>
<td>@ifpackageloaded</td>
<td>657, 726</td>
</tr>
<tr>
<td>@ifundefined</td>
<td>1099</td>
</tr>
<tr>
<td>@undefined</td>
<td>113, 188, 491, 493</td>
</tr>
<tr>
<td>\</td>
<td>957</td>
</tr>
<tr>
<td>{</td>
<td>880</td>
</tr>
<tr>
<td>}</td>
<td>881</td>
</tr>
</tbody>
</table>