The atbegshi package

Heiko Oberdiek*

2019/12/05 v1.19

Abstract

This package is a modern reimplementation of package everyshi without the burden of compatibility. It makes use of \textit{\&}\texttt{T\textit{E}X}'s if available. Both \texttt{T\textit{E}X} and plain \texttt{T\textit{E}X} 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 \textit{\&}\texttt{T\textit{E}X} ......................... 7
    2.3.2 Without \textit{\&}\texttt{T\textit{E}X} .................... 7
    2.3.3 \texttt{\textbackslash lastkern} method ................ 8
  2.4 Output ............................................. 9
  2.5 Separate box register .............................. 9
  2.6 Summary .......................................... 10
    2.6.1 With \textit{\&}\texttt{T\textit{E}X} ......................... 10
    2.6.2 Without \textit{\&}\texttt{T\textit{E}X}, traditional way ....... 10
    2.6.3 \texttt{\textbackslash lastkern} method ............... 11

3 Implementation 11
  3.1 Reload check and package identification ............ 11
  3.2 Catcodes .......................................... 13
  3.3 Preparations .................................... 13
  3.4 Additions to the shipout box ........................ 18
  3.5 Positioning ...................................... 20
  3.6 Patches .......................................... 21
    3.6.1 Package \texttt{crop} ............................ 22
    3.6.2 Package everyshi ............................... 23
    3.6.3 Class \texttt{memoir} ............................ 24

*Please report any issues at \url{https://github.com/ho-tex/atbegshi/issues}
4 Installation
4.1 Download ............................................ 27
4.2 Bundle installation ................................. 27
4.3 Package installation ................................. 28
4.4 Refresh file name databases ....................... 28
4.5 Some details for the interested ................. 28

5 History .............................................. 29
[2007/04/17 v1.0] ........................................ 29
[2007/04/18 v1.1] ........................................ 29
[2007/04/19 v1.2] ........................................ 29
[2007/04/26 v1.3] ........................................ 29
[2007/04/27 v1.4] ........................................ 29
[2007/06/06 v1.5] ........................................ 29
[2007/09/09 v1.6] ........................................ 29
[2008/07/18 v1.7] ........................................ 29
[2008/07/19 v1.8] ........................................ 29
[2008/07/31 v1.9] ........................................ 29
[2009/12/02 v1.10] ..................................... 30
[2010/03/01 v1.11] ..................................... 30
[2010/03/25 v1.12] ..................................... 30
[2010/08/18 v1.13] ..................................... 30
[2010/12/02 v1.14] ..................................... 30
[2011/01/30 v1.15] ..................................... 30
[2011/05/05 v1.16] ..................................... 30
[2016/05/16 v1.17] ..................................... 30
[2016/06/09 v1.18] ..................................... 30
[2019/12/05 v1.19] ..................................... 30

6 Index .................................................. 30

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:

\AtBeginShipout \{⟨code⟩\}
\AtBeginShipoutBox

Execute the ⟨code⟩ 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 L\TeX calls \shipout in \@outputpage (part of its output routine), the meaning of \protect is \noexpand. L\TeX sets \protect to the appropriate \@typeset@protect in the box that is shipped out. This is too late for the hooks, they are called earlier in the redefined \shipout. Therefore package atbegshi sets \protect to \@typeset@protect before it calls the hooks. (In \EveryShipout of package everyshi the user is responsible for the correct setting of \protect.)

\AtBeginShipoutNext{⟨code⟩}

This reimplements package everyshi’s \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:

\newcommand{\MyShipoutHook}{%  
\AtBeginShipout{\MyShipoutHook}\ndef\MyShipoutHook{%  ... do something with next page ...  
\ndef\MyShipoutHook{%
}

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

\AtBeginShipoutFirst{⟨code⟩}

This reimplements L\TeX’s \AtBeginDvi. This hook is usually used for \special commands that include PostScript header files. The ⟨code⟩ is directly executed in a \vbox that is put at the beginning of the output page. Dealing with the output box \AtBeginShipoutBox is not necessary and not permitted here.

\AtBeginShipoutDiscard

This macro notifies package 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 \deadcycles is cleared to zero like an ordinary \shipout would do.

\AtBeginShipoutInit

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

\AtBeginShipoutAddToBox{⟨stuff⟩}  \AtBeginShipoutAddToBoxForeground{⟨stuff⟩}

A quite common use case is the addition of \special or other whatsits to the page output box. Macro \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 \AtBeginShipoutBox. Macro \AtBeginShipoutAddToBox puts the ⟨stuff⟩ in
the background, the other macro `\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 `\AtBeginShipoutBox` is preserved. Also the box nesting level for the original contents of `\AtBeginShipoutBox` remains, for example, to avoid trouble with links across pages in case of pdfTEX.

```
\AtBeginShipoutUpperLeft {{background material}}
```

This is a macro that puts material in the background of box `\AtBeginShipoutBox`. The ⟨background material⟩ is set in an `\hbox`, the reference point is the upper left corner of the output page. In case of pdfTEX 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 ⟨background material⟩ is set inside a `picture` environment:

```
\begin{picture}(0,0)
\setlength{\unitlength}{1pt}\
\hbox{\begin{picture}(\textwidth,\textheight)\put(0,0){⟨background material⟩}\end{picture}}
\end{picture}
```

```
\AtBeginShipoutUpperLeftForeground {{foreground material}}
```

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

```
\AtBeginShipoutOriginalShipout ⟨box⟩
```

It stores the meaning of `\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 `\shipout` is not redefined before the package loading or the box dimensions are not changed by the redefined `\shipout`, these macros contain the dimensions of the shipout box. These values can be remembered by `\label` and `\ref`. For example, this is done by the package module `zref-pagelayout` of project `zref`. The dimensions of the shipout page can be used in some T\TeX{} engines (pdfT\TeX{} in PDF mode, XeT\TeX{}) to calculate the media size of the shipout page if `\pdfpagewidth` and `\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)
Package picture makes life a little easier, because we can now also use length specifications in picture’s commands.

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

\AtBeginShipout{%
\AtBeginShipoutUpperLeft{%
\put(0.5\paperwidth,-0.5\paperheight){\circle{10}}%
}%
%
}%
\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}}%
}%
%
}%
Only on this page we add a red cross.
\newpage
This page has the circle only.
\par
\vspace{\fill}
The next page will be discarded.
\newpage
\AtBeginShipoutNext{%
\AtBeginShipoutDiscard
%
}%
This page is discarded.
\newpage
The last page.
\end{document}

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.

\AtBeginShipout{%
\AtBeginShipoutUpperLeft{%
\put(0.5\paperwidth,-0.5\paperheight){\circle{10}}%
}%
%
}%
\documentclass{minimal}
\usepackage{atbegshi}
\usepackage[dvips, paperwidth=630bp, paperheight=810bp]{geometry}
\AtBeginShipout{%
\setbox\AtBeginShipoutBox=hbox{%
\special{pdf: put @thispage <</TrimBox[9 9 621 801]>>}%
}
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 \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 \hbox, \vbox, or \vtop, e.g. \shipout\hbox{Hello World}.

- **Indirect boxes**: \hbox or \copy references a box register by number. The box register contains the contents of the box.

  Note: \box also clears the box register globally.

Then we have to differentiate between void and empty boxes:

- **Void**: Initially or after \box there is no box in the box register. In this cases the box register is not empty, but *void*.

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

### 2.2 \afterassignment

We want to overload \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\hbox{}
\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 \afterassignment. It takes a macro name and executes it just after the assignment.

Now we can redefine \shipout. The box specification that follows \shipout is caught by \setbox. This is an assignment to a box register. \afterassignment notifies \TeX, that we want to call \@test right after the assignment:
We have seen different box specifications. Indirect boxes are easy to understand:

\shipout \box0 \Rightarrow \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 \@test after the first of such an assignment or after the box specification if the box lacks of assignments. Therefore TEX puts \@test right at the beginning of the box specification, e.g:

\shipout \hbox{Hello World} \Rightarrow \setbox\mybox=\hbox{\@test Hello World}

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:

\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 complicate 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 choosen 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 `4911sp-\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 `\shipout` does not output void boxes. All we have to do is to ensure that our box `\mybox` is always void except for the phase when the overloaded `\shipout` is executed. And secondly we must keep this semantics of `\shipout` for the void case in our macros, namely `\@output`.

```latex
\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
```

The nasty case is `\shipout`\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 `\aftergroup`. Otherwise output is called some time later when the surrounding group closes. But `\mybox` is void outside the execution phase of the redefined `\shipout`. Also `\@output` checks for a void box and cancels the page output. The disadvantage remains that the hook in `\@output` is called for a page that will not be output.

### 2.3.3 \lastkern method

At the beginning of a new box, there is no `\kern`, the contents of the box is still empty and `\lastkern` returns 0 pt. This can be used to distinguish between direct and indirect boxes: We execute `\setbox` in a box with a preceding non-zero kern. After an indirect box, `\lastkern` sees this kern, otherwise it returns 0 pt.
We have two `\setbox` commands. The first creates a controlled context box where we can safely insert a `\kern`. We get rid of this temporarily used context box by putting the local `\setbox` in a group. After the group we want to have our shipout box in `\mybox`. Therefore we use a global assignment here.

### 2.4 Output

With or without \TeX we ensure the original behaviour of `\shipout` that void boxes do not generate output pages.

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

```latex
\@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
```

### 2.5 Separate box register

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

```latex
\shipout\copy<num> \shipout\box<num>
```

We manipulate the box by the hook and without \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.

*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 `\shipout` at all. In fact `\shipout` does not change any
box register. This is usually done by a call of \box, but the output routine can
do it later after invoking of \shipout.

2.6 Summary

2.6.1 With $\varepsilon$-\TeX

Putting the pieces together we get for $\varepsilon$-\TeX:

\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
    \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
\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
\end{verbatim}
\@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

2.6.3 \texttt{lastkern} method

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

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

\shipout :=
\begingroup
\setbox\mybox=\hbox\bgroup
\kern1pt
\afterassignment\@test
\setbox\mybox=

\@test :=
\ifdim\lastkern=0pt
  \expandafter\aftergroup
\fi
\egroup
\endgroup
\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

3 Implementation

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

56 (*package)

3.1 Reload check and package identification

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

57 \begingroup\catcode61\catcode48\catcode32=10\relax%
58 \catcode13=5 % ^^M
\endlinechar=13 %
catcode35=6 % #
catcode39=12 % ,
catcode44=12 % ,
catcode45=12 % -
catcode46=12 % .
catcode58=12 % :
catcode64=11 % @
catcode123=1 % {
catcode125=2 % }
\expandafter\let\expandafter\x\csname ver@atbegshi.sty\endcsname
\ifx\x\relax % plain-TeX, first loading
\else
\def\empty{}
\ifx\x\empty % LaTeX, first loading,
% variable is initialized, but \ProvidesPackage not yet seen
\else
\expandafter\ifx\csname PackageInfo\endcsname\relax
\def\x#1#2#3[#4]{%
  \immediate\write-1{Package #1 Info: #2.}%
  }%
\else
\def\x#1#2[#3]{\PackageInfo{#1}{#2, stopped}}%
\fi
\x{atbegshi}{The package is already loaded}%
\aftergroup\endinput
\fi
\fi
\endgroup%
Package identification:
\begingroup\catcode61\catcode48\catcode32=10\relax%
catcode13=5 % ^M
\endlinechar=13 %
catcode35=6 % #
catcode39=12 % ,
catcode40=12 % (,
catcode41=12 % )
catcode44=12 % ,
catcode45=12 % -
catcode46=12 % .
catcode47=12 % /
catcode58=12 % :
catcode64=11 % @
catcode91=12 % [
catcode93=12 % ]
catcode123=1 % {
catcode125=2 % }
\expandafter\ifx\csname ProvidesPackage\endcsname\relax
\def\x#1#2#3[#4]{\endgroup
  \immediate\write-1{Package: #3 #4}%
  \xdef#1{#4}%
  }%
\else
\def\x#1#2[#3]{\endgroup
  \ifx#1\@undefined
    \xdef#1{#3}%
  \fi
  \fi
\endgroup%
3.2 Catcodes

\begingroup\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 % ^^M
\endlinechar=13 %
\catcode123=1 % {
\catcode125=2 % }
\catcode64=11 % @
\def\x{\endgroup
\expandafter\edef\csname AtBegShi@AtEnd\endcsname{%
\endlinechar=\the\endlinechar\relax
\catcode13=\the\catcode13\relax
\catcode32=\the\catcode32\relax
\catcode35=\the\catcode35\relax
\catcode61=\the\catcode61\relax
\catcode64=\the\catcode64\relax
\catcode123=\the\catcode123\relax
\catcode125=\the\catcode125\relax
)}%
\x\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 % ^^M
\endlinechar=13 %
\catcode35=6 % #
\catcode123=1 % {
\catcode125=2 % }
\def\TMP@EnsureCode#1#2{%
\edef\AtBegShi@AtEnd{\AtBegShi@AtEnd
\noexpand\endinput}
}\endgroup
3.3 Preparations

\begingroup\expandafter\expandafter\expandafter\endgroup
\def\AtBegShi@Shipout{%
  X \begingroup
  X \setbox\AtBeginShipoutBox=\hbox\bgroup
  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
  \AtBegShi@Output
  \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
    \AtBeginShipoutOriginalShipout\box\AtBeginShipoutBox
  \fi
}

\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 
}
\AtBegShi@FirstDisabled
\long\def\AtBegShi@FirstDisabled#1{\%
\PackageWarning{atbegshi}{First page is already shipped out, ignoring\MessageBreak
\string\AtBeginShipoutFirst 
}%
}

16
\AtBeginShi@AddTo
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname g@addto@macro\endcsname\relax
\long\def\AtBeginShi@AddTo#1#2{%
\begingroup
\toks\ltx@zero\expandafter{#1#2}%
\xdef#1{\the\toks\ltx@zero}%
\endgroup
)%
\else
\let\AtBeginShi@AddTo\g@addto@macro
\fi
\AtBeginShi@AddHook
\long\def\AtBeginShi@AddHook#1#2{%
\AtBeginShi@AddTo#1{\AtBeginShi@Item{#2}}%
}\AtBeginShi@Item
\long\def\AtBeginShi@Item#1{%
\ifAtBeginShi@Discarded
\else
#1%
\fi
\else
\ifvoid\AtBeginShipoutBox
\@PackageWarning{atbegshi}{%Shipout box was voided by hook,\MessageBreak
ignoring shipout box%}
}(%
\AtBeginShipoutDiscard
\fi
\fi
\fi
\AtBeginShipoutInit
\AtBeginShi@CheckDefinable\AtBeginShipoutInit
\def\AtBeginShipoutInit{%
\ltx@IfUndefined{newbox}{%
\@PackageError{atbegshi}{%string \AtBeginShipoutInit\space failed,MessageBreak
because of missing \expandafter\string\csname newbox\endcsname
}@ehc
}{%
\csname newbox\endcsname\AtBeginShipoutBox
\AtBeginShi@CheckDefinable\AtBeginShipoutOriginalShipout
\global\let\AtBeginShipoutOriginalShipout\shipout
\global\let\shipout\AtBeginShi@Shipout
}%
\gdef\AtBeginShipoutInit{}%
}\fi
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname AtBeginDocument\endcsname\relax
\AtBeginShipoutInit
\else
\AtBeginDocument{\AtBeginShipoutInit}%
\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@restorebox{%  
        \vfuzz=\the\vfuzz\relax  
        \vbadness=\the\vbadness\relax  
        \dimen\ltx@zero=\the\dimen\ltx@zero\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  
      }%  
      \AtBegShi@restorebox  
      \else  
        \AtBegShi@restore  
      \fi  
    \fi  
  \fi  
}
\AtBeginShipoutAddToBoxForeground
\def\AtBeginShipoutAddToBoxForeground#1{%
\ifhbox\AtBeginShipoutBox
  \edef\AtBegShi@restore{%
    \hfuzz=1073741823sp\relax
    \hbadness=2147483647\relax
  }%
  \hfuzz=1073741823sp\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=0pt\relax
    \ht\ltx@zero=0pt\relax
    \dp\ltx@zero=0pt\relax
    \raise\ht\AtBeginShipoutBox\copy\ltx@zero
    \kern-\wd\AtBeginShipoutBox
  }%
}\AtBeginShipoutBox
\else
  \edef\AtBegShi@restore{%
    \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
      \ht\AtBeginShipoutBox=\the\ht\AtBeginShipoutBox\relax
      \dp\AtBeginShipoutBox=\the\dp\AtBeginShipoutBox\relax
    }%
    \wd\ltx@zero=0pt\relax
    \ht\ltx@zero=0pt\relax
    \dp\ltx@zero=0pt\relax
    \raise\ht\AtBeginShipoutBox\copy\ltx@zero
    \kern-\wd\AtBeginShipoutBox
  }%
}\AtBeginShi@restore
\fi
}
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{iftex}[2019/11/07]%
\else
\RequirePackage{iftex}[2019/11/07]%
\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
0%
\fi
\expandafter\ifx\csname endpicture\endcsname\relax 1%
\else
0%
\fi
\fi
\endgroup
\def\AtBegShi@BeginPicture{\%
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@BeginPicture{%}
\begingroup
\picture(0,0)\relax
\endgroup
\def\AtBegShi@EndPicture{%
\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 \texttt{\shipout}. \LaTeX{} is now supposed to use \epsilon\TeX{}. Thus we do not patch, without \LaTeX{} and \epsilon\TeX{}.
\def\AtBegShi@AbortIfUndefined#1{%
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname#1\endcsname\relax
\expandafter\AtBegShi@AtEnd
\fi
}
3.6.1 Package crop

Fix of method and box.

\def\AtBegShi@PatchCrop{%
\begingroup
  \def\AtBegShi@Crop@shipout{%
    \afterassignment\CROP@ship
    \setbox\@cclv=%
  }%
  \def\AtBegShi@Crop@ship{%
    \ifvoid\@cclv
      \expandafter\aftergroup
      \fi
      \CROP@@ship
    \fi
  }%
  \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
                    \expandafter\endgroup
                    \edef\AtBegShi@GroupLevel{\number\currentgrouplevel}%
                    \afterassignment\CROP@ship
                    \setbox\AtBeginShipoutBox=%
                  \fi
                \fi
              \fi
            \fi
          \fi
        \fi
      \fi
    \fi
  \fi
\endgroup
}
3.6.2 Package everyshi

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

\def\AtBegShi@PatchEveryshi{%
\begingroup
\long\def\AtBegShi@Everyshi@shipout{%
\afterassignment\@EveryShipout@Test
\global\setbox\@cclv=\box\AtBeginShipoutBox
}\%}
\long\def\AtBegShi@Everyshi@Test{%
\ifvoid\@cclv\relax
\@EveryShipout@Output
\else
\@EveryShipout@Output
\fi
}\%
\PackageInfoNoLine{atbegshi}{Package ‘crop’ patched}%
\begingroup
\fi
\fi
\fi
\endgroup
\let\AtBegShi@PatchCrop\relax
}%
\begin{document}{\AtBegShi@PatchCrop}%
\ifpackageloaded{crop}{%
\AtBegShi@PatchCrop
}%
\AtBeginDocument{\AtBegShi@PatchCrop}%
}%

\def\AtBegShi@PatchEveryshi{%
\begingroup
\long\def\AtBegShi@Everyshi@shipout{%
\afterassignment\@EveryShipout@Test
\global\setbox\@cclv=\box\AtBeginShipoutBox
}\%}
\long\def\AtBegShi@Everyshi@Test{%
\ifvoid\@cclv\relax
\@EveryShipout@Output
\else
\@EveryShipout@Output
\fi
}\%
\PackageInfoNoLine{atbegshi}{Package ‘crop’ patched}%
\begingroup
\fi
\fi
\fi
\endgroup
\let\AtBegShi@PatchCrop\relax
%
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 \unvbox\@cclv }
}\def\AtBegShi@Memoir@shipiiC{2008/08/07 v1.6180339a
\ifvoid\@cclv \mem@oldshipout\box\@cclv \else \mem@oldshipout\vbox{\trimmarks \unvbox\@cclv }
}\def\AtBegShi@Memoir@shipiiD{2011/03/06 v3.6j
\ifvoid\@cclv \mem@oldshipout\box\@cclv \else \mem@oldshipout\vbox{\trimmarks \unvbox\@cclv }
}\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
\AtBegShi@GetBoxSize\AtBeginShipoutBox \expandafter\mem@oldshipout
\expandafter\box
\expandafter\AtBeginShipoutBox \fi
\AtBegShi@GetBoxSize\AtBeginShipoutBox \expandafter\mem@oldshipout
\expandafter\box
\expandafter\AtBeginShipoutBox \fi
\AtBegShi@GetBoxSize\AtBeginShipoutBox \expandafter\mem@oldshipout
\expandafter\box
\expandafter\AtBeginShipoutBox \fi
\AtBegShi@GetBoxSize\AtBeginShipoutBox \expandafter\mem@oldshipout
\expandafter\box
\expandafter\AtBeginShipoutBox \fi
\AtBegShi@GetBoxSize\AtBeginShipoutBox \expandafter\mem@oldshipout
\expandafter\box
\expandafter\AtBeginShipoutBox \fi
\AtBegShi@GetBoxSize\AtBeginShipoutBox \expandafter\mem@oldshipout
\expandafter\box
\expandafter\AtBeginShipoutBox \fi
\AtBegShi@GetBoxSize\AtBeginShipoutBox \expandafter\mem@oldshipout
\expandafter\box
\expandafter\AtBeginShipoutBox \fi

4 Installation

4.1 Download

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


Bundle. All the packages of the bundle ‘atbegshi’ 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/atbegshi.tds.zip

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

4.2 Bundle installation

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

\begin{verbatim}
\fi
\fi\fi\fi\fi\fi

\if\AtBegShi@found\relax
\else
\expandafter\endgroup
\expandafter\def\AtBegShi@found{%
\edefAtBegShi@GroupLevel\number\currentgrouplevel%
\afterassignment\mem@shipi
\setbox\AtBeginShipoutBox=%
}
\def\mem@shipi{%
\ifnum\AtBegShi@GroupLevel=\currentgrouplevel
\else
\expandafter\aftergroup
\fi
\mem@shipii
}
\let\mem@shipii\AtBegShi@Memoir@PatchX
\@PackageInfoNoLine{atbegshi}{Class ‘memoir’ patched}%
\begingroup
\fi
\fi
\fi
\endgroup
\let\AtBegShi@PatchMemoir\relax
\@ifclassloaded{memoir}{%
\AtBegShi@PatchMemoir
}{}%
\ifclassloaded{memoir}{}
\AtBeginDocument{\AtBegShi@PatchMemoir}%
\AtBeginShi@AtEnd%
unzip atbegshi.tds.zip -d ~/texmf

4.3 Package installation

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

```
tex atbegshi.dtx
```

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

```
atbegshi.sty → tex/generic/atbegshi/atbegshi.sty
atbegshi.pdf → doc/latex/atbegshi/atbegshi.pdf
atbegshi-example1.tex → doc/latex/atbegshi/atbegshi-example1.tex
atbegshi-example2.tex → doc/latex/atbegshi/atbegshi-example2.tex
atbegshi.dtx → source/latex/atbegshi/atbegshi.dtx
```

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

4.4 Refresh file name databases

If your \TeX{} distribution (\TeX{} Live, \mikTex{}, ...) relies on file name databases, you must refresh these. For example, \TeX{} Live users run texhash or mktexlsr.

4.5 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 \pdfTeX:

```
pdflatex atbegshi.dtx
makeindex -s gind.ist atbegshi.idx
pdflatex atbegshi.dtx
makeindex -s gind.ist atbegshi.idx
pdflatex atbegshi.dtx
```
5 History

[2007/04/17 v1.0]
• First version.

[2007/04/18 v1.1]
• New method based on \lastkern is used if \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 \evershi 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 \pdfhorigin in new Lua\TeX{}.

[2019/12/05 v1.19]
  • Documentation updates.
  • iftex package.

6 Index

Numbers written in italic refer to the page where the corresponding entry is des-
cribed; numbers underlined refer to the code line of the definition; plain numbers
refer to the code lines where the entry is used.

Symbols
\@EveryShipout\@Org\@Shipout ..... 30