The \texttt{atbegshi} package

Heiko Oberdiek
\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

*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:

\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 \LaTeX{} calls \texttt{\shipout} in \texttt{@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{\EveryShipout} of package \texttt{everyshi} the user is responsible for the correct setting of \texttt{\protect}.)

\texttt{\AtBeginShipoutNext \{\langle code\rangle\}}

This reimplements package \texttt{everyshi}'s \texttt{\AtNextShipout}. The \texttt{\langle code\rangle} 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 ... 
gdef\MyShipoutHook{% }
}
\end{verbatim}

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

\texttt{\AtBeginShipoutFirst \{\langle code\rangle\}}

This reimplements \LaTeX{}'s \texttt{\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.

\texttt{\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{\deadcycles} is cleared to zero like an ordinary \texttt{\shipout} would do.

\texttt{\AtBeginShipoutInit}

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}.

\texttt{\AtBeginShipoutAddToBox \{\langle stuff\rangle\}}
\texttt{\AtBeginShipoutAddToBoxForeground \{\langle stuff\rangle\}}

A quite common use case is the addition of \texttt{\special} or other whatsits to the page output box. Macro \texttt{\AtBeginShipoutAddToBox} puts \texttt{\langle stuff\rangle} in a box with zeroed dimensions. The box with the \texttt{\langle stuff\rangle} is put in the upper left corner of the shipout box \texttt{\AtBeginShipoutBox}. Macro \texttt{\AtBeginShipoutAddToBox} puts the \texttt{\langle stuff\rangle} 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 \texttt{\langle stuff\rangle} 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 {⟨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}\langle background material\rangle\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 \TeX engines (pdf\TeX in PDF mode, \Xe\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)
2 \documentclass[a4paper]{article}
3 \usepackage{color}
4 \usepackage{atbegshi}
5 \usepackage{picture}
6 \AtBeginShipout{\%
7 \AtBeginShipoutUpperLeft{\%
8 put(0.5\paperwidth,-0.5\paperheight){\circle{10}}\%

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. \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\paperwidth,-0.5\paperheight){\circle{10}}\%
\section{Hello World}

Only on this page we add a red cross.

This page has the circle only.

The next page will be discarded.

This page is discarded.

The last page.

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.

\begin{document}
First page
\newpage
Second page
\end{document}
2 Method of \shipout overloading

2.1 \shipout

The Te\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: \box 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:

\shipout\null
\shipout\vbox{...}
\shipout\vtop\bgroup ...\egroup
\shipout\box255

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

\shipout\hbox\bgroup}
\shipout\vbox{\egroup

Happily Te\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 Te\TeX, that we want to call \@test right after the assignment:

\shipout :=
\afterassignment\@test
\setbox\mybox=

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

\shipout\box0 ⇒ \setbox\mybox=\box0 \@test

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

\shipout\hbox{Hello World}
⇒ \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:

\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 \(\varepsilon\)-\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 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 `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 lose 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.

\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\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 \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.

\shipout :=
\begingroup
\setbox\mybox=\hbox\bgROUP
\kern1pt
\afterassignment\shipout@test
\global\setbox\mybox=
\@test :=
\ifdim\lastkern=0pt
\% direct box
\aftergroup\egroup
\aftergroup\endgroup
\aftergroup\@output
\else
\egroup
\endgroup
\@output
\fi

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 \kern 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 \(\varepsilon\)-\TeX we ensure the original behaviour of \texttt{\textbackslash shipout} that void boxes do not generate output pages.

Now we can place the hook \texttt{\textbackslash @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

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  \\
\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
\@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 $\texttt{lastkern}$ method

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

\begin{verbatim}
\newbox\mybox
\let\original\shipout\shipout

\shipout :=
\begingroup
  \setbox\mybox=hbox\bgroup
  \kern1pt
\endgroup
\afterassignment\@test
\setbox\mybox=
\@test :=
\ifdim\lastkern=0pt
\expandafter\aftergroup\fi
\@output
\@output :=
\end{verbatim}
3 Implementation

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

3.1 Reload check and package identification

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

Package identification:

```latex
\begingroup\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 \endlinechar=13 %
\catcode35=6 %
\catcode39=12 %
\catcode44=12 %
\catcode45=12 %
\catcode46=12 %
\catcode58=12 %
\catcode64=11 %
\expandafter\let\expandafter\x\csname ver\atbegshi.sty\endcsname
\endgroup%
```
3.2 Catcodes
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
\}
\else
\def\AtBegShi@CheckDefinable#1{%
\@ifdefinable{#1}{}%
\fi
\fi
\def\AtBeginShipoutDiscard{%
\AtBegShi@CheckDefinable\AtBeginShipoutDiscard
\def\AtBeginShipoutDiscard{%
\@ifdefinable{#1}{% 
\string#1\space is already defined}%
\@ehd
\}
\else
\def\AtBeginShipoutDiscard{%
\@ifdefinable{#1}{% 
\string#1\space is already defined}%
\@ehd
\}
\fi

\ifAtBegShi@Discarded
\ltx@newif\ifAtBegShi@Discarded
\AtBeginShipoutDiscard
\def\AtBeginShipoutDiscard{%

\end{document}
\defdeadcycles=\@xzero
\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
\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
\expandafter\gdef\expandafter\AtBegShi@HookNext\expandafter{\expandafter}
\AtBegShi@HookNext\ifAtBegShi@Discarded
@PackageWarningNoLine{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}{}{\AtNextShipout\@cclv}%
\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}{}{\AtNextShipout\@cclv}%
}(%
\AtNextShipout\AtBegShi@GetBoxSize\@cclv\)%
})%
\AtBeginShipoutOriginalShipout\box\AtBeginShipoutBox
\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
\AtBeginShipoutInit
335 \def\AtBeginShipoutInit\@CheckDefinable\AtBeginShipoutInit
336 \def\AtBeginShipoutInit{%
337 \ltx@if\IfUndefined\newbox\}@\PackageError\atbegshi{%
338 }\@ehc
339 \string\AtBeginShipoutInit\space failed\MessageBreak
340 because of missing \expandafter\string\cname newbox\endcsname
341 }\@ehc
342 }%
343 \cname newbox\endcsname\AtBeginShipoutBox
344 \AtBeginShipoutInit\@CheckDefinable\AtBeginShipoutOriginalShipout
345 \global\let\AtBeginShipoutOriginalShipout\shipout
346 \global\let\shipout\AtBeginShip\@Shipout
347 }%
348 \gdef\AtBeginShipoutInit{}%
349 }
350 \begingroup\expandafter\expandafter\expandafter\endgroup
\AtBeginShipoutAddToBox

3.4 Additions to the shipout box

\def\AtBeginShipoutAddToBox#1{\% 356  \ifhbox\AtBeginShipoutBox 357  \edef\AtBegShi@restore{\% 358  \hfuzz=\the\hfuzz\relax 359  \hbadness=\the\hbadness\relax 360  }\% 361  \hfuzz=1073741823sp\relax 362  \hbadness=2147483647\relax 363  \setbox\AtBeginShipoutBox=\hbox to \wd\AtBeginShipoutBox{\% 364  \setbox\ltx@zero=\hbox{\% 365  \begingroup\AtBegShi@restore\% 366  \#1\% 367  \endgroup 368  }\% 369  \wd\ltx@zero=0pt\relax 370  \ht\ltx@zero=0pt\relax 371  \dp\ltx@zero=0pt\relax 372  \raise\ht\AtBeginShipoutBox\copy\ltx@zero 373  \unhcopy\AtBeginShipoutBox 374  }\% 375  \AtBegShi@restore 376  \else\% 377  \ifvbox\AtBeginShipoutBox 378  \edef\AtBegShi@restore{\% 379  \vfuzz=\the\vfuzz\relax 380  \vbadness=\the\vbadness\relax 381  \dimen\ltx@zero=\the\dimen\ltx@zero\relax 382  }\% 383  \edef\AtBegShi@restorebox{\% 384  \ht\AtBeginShipoutBox=\the\ht\AtBeginShipoutBox\relax 385  }\% 386  \advance\dimen\ltx@zero by \dp\AtBeginShipoutBox 387  \setbox\AtBeginShipoutBox=\vbox to \dimen\ltx@zero{\% 388  \setbox\ltx@zero=\hbox{\% 389  \begingroup\AtBegShi@restore\% 390  \#1\% 391  \endgroup 392  }\% 393  \wd\ltx@zero=0pt\relax 394  \ht\ltx@zero=0pt\relax 395  \dp\ltx@zero=0pt\relax 396  \baselineskip=0pt\relax 397  \lineskip=0pt\relax 398  \lineskiplimit=0pt\relax 399  \copy\ltx@zero 400  \unvbox\AtBeginShipoutBox 401  }\% 402  \kern0pt\%
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
\expandafter\ifx\csname endpicture\endcsname\relax
1%
\else
0%
\fi
\fi
\endgroup
\def\AtBegShi@BeginPicture{%
\begingroup
\picture(0,0)\relax
\endgroup
\ignorespaces}
\def\AtBegShi@EndPicture{%
\begin{verbatim}
% \def\AtBegShi@EndPicture{% 
% }\def\AtBegShi@AbortIfUndefined#1{% 
% \begingroup\expandafter\expandafter\expandafter\endgroup 
% \expandafter\ifx\csname#1\endcsname\relax 
% \expandafter\AtBegShi@AtEnd 
% \fi 
% }\AtBegShi@AbortIfUndefined{currentgrouplevel}% \AtBegShi@AbortIfUndefined{AtBeginDocument}% \AtBegShi@AbortIfUndefined{@ifpackageloaded}% \AtBegShi@AbortIfUndefined{@ifclassloaded}% 
\end{verbatim}

3.6 Patches

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

3.6.1 Package \texttt{crop}

Fix of method and box.

\begin{verbatim}
% \def\AtBegShi@PatchCrop{% 
% }\begingroup 
% \def\AtBegShi@Crop@shipout{% 
% \afterassignment\CROP@shipout 
% \edef\CROP@shipout{
% \begingroup\expandafter\expandafter\expandafter\endgroup 
% \expandafter\ifx\csname\textbackslash shipout\endcsname\relax 
% \expandafter\AtBegShi@AtEnd 
% \fi 
% }\AtBegShi@PatchCrop% 
\end{verbatim}

3.6.2 Package everyshi

Fix of method. Use of box 255 is not changed.
3.6.3 Class \texttt{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
      \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
      }% 
    }% 
  }% 
}\AtBeginDocument{\AtBegShi@PatchEveryshi}%
```

23
\unvbox\@cclv
\fi
}\%
\def\AtBegShi@Memoir@PatchAB{%
\ifvoid\AtBeginShipoutBox
\else
\setbox\AtBeginShipoutBox=\vbox{\trimmarks
\ifvbox\AtBeginShipoutBox
\unvbox\AtBeginShipoutBox
\else
\box\AtBeginShipoutBox
\fi
}\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
}\%
\fi
\fi
\AtBegShi@GetBoxSize\AtBeginShipoutBox
\expandafter\mem@oldshipout\box\AtBeginShipoutBox
\expandafter\mem@oldshipout\
\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
}\%
\fi
\fi
\AtBegShi@GetBoxSize\AtBeginShipoutBox
\expandafter\mem@oldshipout\box\AtBeginShipoutBox
\expandafter\mem@oldshipout\
\fi
\fi
}24
\ifx\AtBegShi@Memoir@shipi\mem@shipi
\let\AtBegShi@found\ltx@one
\ifx\AtBegShi@Memoir@shipiiA\mem@shipi
\let\AtBegShi@found\ltx@zero
\global\let\AtBegShi@Memoir@PatchX\AtBegShi@Memoir@PatchAB
\else\ifx\AtBegShi@Memoir@shipiiB\mem@shipi
\let\AtBegShi@found\ltx@zero
\global\let\AtBegShi@Memoir@PatchX\AtBegShi@Memoir@PatchAB
\else\ifx\AtBegShi@Memoir@shipiiC\mem@shipi
\let\AtBegShi@found\ltx@zero
\global\let\AtBegShi@Memoir@PatchX\AtBegShi@Memoir@PatchCD
\else\ifx\AtBegShi@Memoir@shipiiD\mem@shipi
\let\AtBegShi@found\ltx@zero
\global\let\AtBegShi@Memoir@PatchX\AtBegShi@Memoir@PatchCD
\fi\fi\fi\fi
\ifcase\AtBegShi@found
\let\AtBegShi@found\relax
\ifx\shipout\AtBegShi@Memoir@shipout
\def\AtBegShi@found{\shipout}%
\else\ifx\AtBeginShipoutOriginalShipout\AtBegShi@Memoir@shipout
\def\AtBegShi@found{\AtBeginShipoutOriginalShipout}%
\else\ifx\CROP@shipout\AtBegShi@Memoir@shipout
\def\AtBegShi@found{\CROP@shipout}%
\else\ifx\GPTorg@shipout\AtBegShi@Memoir@shipout
\def\AtBegShi@found{\GPTorg@shipout}%
\else\ifx\THBorg@shipout\AtBegShi@Memoir@shipout
\def\AtBegShi@found{\THBorg@shipout}%
\else\ifx\@EveryShipout@Org@Shipout\AtBegShi@Memoir@shipout
\def\AtBegShi@found{\@EveryShipout@Org@Shipout}%
\fi\fi\fi\fi\fi\fi
\ifx\AtBegShi@found\relax
\else
\expandafter\endgroup
\expandafter\def\AtBegShi@found{%
\edef\AtBegShi@GroupLevel{\number\currentgrouplevel}%
\afterassignment\mem@shipi
\setbox\AtBeginShipoutBox=%
}%
\def\mem@shipi{%
\ifnum\AtBegShi@GroupLevel=\currentgrouplevel
\else
\expandafter\aftergroup
\mem@shipii
\}%
\let\mem@shipii\AtBegShi@Memoir@PatchX
\@PackageInfoNoLine(atbegshi){Class `memoir' patched}%
\begingroup
\fi
\fi
\endgroup
\let\AtBegShi@PatchMemoir\relax
\ifclassloaded{memoir}{%
\AtBegShi@PatchMemoir
\AtBeginDocument{\AtBegShi@PatchMemoir}%
}{}
\AtBegShi@AtEnd%
\PackageInfo{atbegshi}{Class `memoir' patched}
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#3=\count@ \\
\else
\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
\space
\ifcsname 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*/
\input atbegshi.sty\relax
\def\msg#1#2{\msg{File: atbegshi-test2.tex 2016/06/09 v1.18 Test file for plain-\TeX}\
\def\testmsg#1#2{\msg{}\msg{*** Test with box (#1), expected page output [#2]}% hash-ok\}
\msg{\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}%
\AtBeginShipoutFirst{%
\msg{* Inside \string\AtBeginShipoutFirst}%
Hello World%
}
\testmsg{\string\null}{1}
\shipout\null
\AtBeginShipoutFirst{%
This is too late%
}
\testmsg{}{}
\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}{8}
\def\VoidBox{%
\begingroup
\setbox\AtBeginShipoutBox=\box\AtBeginShipoutBox
\endgroup
}
\AtBeginShipoutBox=%
\endgroup
\AtBeginShipoutBox=%
\def\VoidBox{}
\testmsg{output routine}{9}
Hello World
5 Installation

5.1 Download

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

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

\texttt{CTAN:macros/latex/contrib/oberdiek/atbegshi.pdf} Documentation.

Bundle. All the packages of the bundle ‘oberdiek’ are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

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

\textit{TDS} refers to the standard “A Directory Structure for \TeX\ 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):

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

If you have a \texttt{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.

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{mktexlsr}.

\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 \texttt{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:

\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}
<?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}{everyshi}, providing various commands
to be executed before a \texttt{\shipout} 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}{oberdiek} 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>
</catalogue>
\end{verbatim}
7 History

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

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

[2007/04/19 v1.2]
- `\AtBeginShipoutEarly` removed for simplification reasons.
- Forgotten definition of `\AtBegShi@Info` 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.

Symbols

\# ........................................ 882
\% ........................................... 958
\@ ........................................... 883, 956
\@end .................................... 1101
\@EveryShipout@Init ............. 1103, 1104
\@EveryShipout@Org@Shipout ....
...................... 605, 606, 845, 846
\@EveryShipout@Output .. 670, 672, 716
\@EveryShipout@Shipout ....... 692, 693
\@EveryShipout@Test 665, 675, 702, 705
\@PackageError ............... 194, 338
\@PackageInfoNoLine 248, 648, 719, 864
\@PackageWarning ......... 238, 300, 326
\@cclv ......................... 260, 577, 580, 590, 666, 669, 735,
\@ehc ............................... 341
\@ehd ................................ 196
\@firstofone .................. 891, 894
\@gobble ....................... 888, 896
\@ifclassloaded ................ 872
\@ifundefined .................. 1099
\@undefined .................... 113, 188, 491, 493
\\ ........................................ 957
\{ ........................................ 880
\} ........................................ 881