The **atbegshi** package

Heiko Oberdiek
<heiko.oberdiek at googlemail.com>

2011/10/05 v1.16

Abstract

This package is a modern reimplementation of package everyshi without the burden of compatibility. It makes use of ε-TeX's if available. Both LaTeX and plain TeX are supported.

Contents

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 `\shipout` in `\@outputpage` (part of its output routine), the meaning of `\protect` is `\noexpand`. LaTeX 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}
\gdef\MyShipoutHook{%
   ... do something with next page ...
  \gdef\MyShipoutHook{}%
 }
```

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

`\AtBeginShipoutFirst {⟨code⟩}`

This reimplements LaTeX'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}%
  ⟨background material⟩
\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 (pdfTeX in PDF mode, XƎTeX) 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}
```

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}}%
 9   }%
10 }
11 \begin{document}
12 \section{Hello World}
13 \newpage
14 \AtBeginShipoutNext{%
```

```
15    \AtBeginShipoutUpperLeft{%
16      \color{red}%
17      \put(0,-0.5\paperheight){\line(1,0){\paperwidth}}%
18      \put(0.5\paperwidth, 0){\line(0,-1){\paperheight}}%
19    }%
20  }
21  Only on this page we add a red cross.
22  \newpage
23  This page has the circle only.
24  \par
25  \vspace{\fill}
26  The next page will be discarded.
27  \newpage
28  \AtBeginShipoutNext{%
29    \AtBeginShipoutDiscard
30  }
31  This page is discarded.
32  \newpage
33  The last page.
34  \end{document}

35  ⟨/example1⟩
```

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.

```
36  ⟨*example2⟩
37  \documentclass{minimal}
38  \usepackage{atbegshi}
39  \usepackage[
40    dvipdfm,
41    paperwidth=630bp,
42    paperheight=810bp
43  ]{geometry}
44  \AtBeginShipout{%
45    \setbox\AtBeginShipoutBox=\hbox{%
46      \special{pdf: put @thispage <</TrimBox[9 9 621 801]>>}%
47      \box\AtBeginShipoutBox
48    }%
49  }
50  \begin{document}
51    First page
52    \newpage
53    Second page
54  \end{document}
55  ⟨/example2⟩
```

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: `\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 reliable 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 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 catched by `\setbox`. This is an assignment to a box register. `\afterassignment` notifies 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:

$\shipout\box0 \Rightarrow \setbox\mybox=\box0$ ⟨`\@test`⟩

However direct boxes can have arbitrary contents with lots of other assignments. It would be quite inpredictable 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 ε-TeX

With the ε-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 ε-TeX

Life becomes complicate without ε-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`.

```
\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 $0\,\mathrm{pt}$. 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 $0\,\mathrm{pt}$.

```
\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 `\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 ε-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.

```
\@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:

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

We manipulate the box by the hook and without ε-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 -TeX

Putting the pieces together we get for ε-TeX:

```
\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
```

2.6.2   Without -TeX, traditional way

And for TeX without ε-TeX:

```
\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
```

2.6.3   `\lastkern` method

And for TeX without ε-TeX using the `\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
  \@output

\@output :=
  \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 atbegshi uses ε-TeX's \currentgrouplevel, if it is available. Otherwise the \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
59   \endlinechar=13 %
60   \catcode35=6 % #
61   \catcode39=12 % '
62   \catcode44=12 % ,
63   \catcode45=12 % -
64   \catcode46=12 % .
65   \catcode58=12 % :
66   \catcode64=11 % @
67   \catcode123=1 % {
68   \catcode125=2 % }
69   \expandafter\let\expandafter\x\csname ver@atbegshi.sty\endcsname
70   \ifx\x\relax % plain-TeX, first loading
71   \else
72     \def\empty{}%
73     \ifx\x\empty % LaTeX, first loading,
74       % variable is initialized, but \ProvidesPackage not yet seen
75     \else
76       \expandafter\ifx\csname PackageInfo\endcsname\relax
77         \def\x#1#2{%
78           \immediate\write-1{Package #1 Info: #2.}%
79         }%
80       \else
81         \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
82       \fi
83       \x{atbegshi}{The package is already loaded}%
84       \aftergroup\endinput
85     \fi
86   \fi
87 \endgroup%
```

Package identification:

```
88 \begingroup\catcode61\catcode48\catcode32=10\relax%
89   \catcode13=5 % ^^M
90   \endlinechar=13 %
91   \catcode35=6 % #
92   \catcode39=12 % '
93   \catcode40=12 % (
94   \catcode41=12 % )
95   \catcode44=12 % ,
96   \catcode45=12 % -
97   \catcode46=12 % .
98   \catcode47=12 % /
99   \catcode58=12 % :
100   \catcode64=11 % @
```

```
101  \catcode91=12 % [
102  \catcode93=12 % ]
103  \catcode123=1 % {
104  \catcode125=2 % }
105  \expandafter\ifx\csname ProvidesPackage\endcsname\relax
106    \def\x#1#2#3[#4]{\endgroup
107      \immediate\write-1{Package: #3 #4}%
108      \xdef#1{#4}%
109    }%
110  \else
111    \def\x#1#2[#3]{\endgroup
112      #2[{#3}]%
113      \ifx#1\@undefined
114        \xdef#1{#3}%
115      \fi
116      \ifx#1\relax
117        \xdef#1{#3}%
118      \fi
119    }%
120  \fi
121 \expandafter\x\csname ver@atbegshi.sty\endcsname
122 \ProvidesPackage{atbegshi}%
123   [2011/10/05 v1.16 At begin shipout hook (HO)]%
```

3.2 Catcodes

```
124 \begingroup\catcode61\catcode48\catcode32=10\relax%
125   \catcode13=5 % ^^M
126   \endlinechar=13 %
127   \catcode123=1 % {
128   \catcode125=2 % }
129   \catcode64=11 % @
130   \def\x{\endgroup
131     \expandafter\edef\csname AtBegShi@AtEnd\endcsname{%
132       \endlinechar=\the\endlinechar\relax
133       \catcode13=\the\catcode13\relax
134       \catcode32=\the\catcode32\relax
135       \catcode35=\the\catcode35\relax
136       \catcode61=\the\catcode61\relax
137       \catcode64=\the\catcode64\relax
138       \catcode123=\the\catcode123\relax
139       \catcode125=\the\catcode125\relax
140     }%
141   }%
142 \x\catcode61\catcode48\catcode32=10\relax%
143 \catcode13=5 % ^^M
144 \endlinechar=13 %
145 \catcode35=6 % #
146 \catcode64=11 % @
147 \catcode123=1 % {
148 \catcode125=2 % }
149 \def\TMP@EnsureCode#1#2{%
150   \edef\AtBegShi@AtEnd{%
151     \AtBegShi@AtEnd
152     \catcode#1=\the\catcode#1\relax
153   }%
154   \catcode#1=#2\relax
155 }
156 \TMP@EnsureCode{40}{12}% (
157 \TMP@EnsureCode{41}{12}% )
158 \TMP@EnsureCode{44}{12}% ,
159 \TMP@EnsureCode{45}{12}% -
```

```
160 \TMP@EnsureCode{47}{12}% /
161 \TMP@EnsureCode{46}{12}% .
162 \TMP@EnsureCode{58}{12}% :
163 \TMP@EnsureCode{91}{12}% [
164 \TMP@EnsureCode{93}{12}% ]
165 \TMP@EnsureCode{94}{7}% ^(superscript)
166 \TMP@EnsureCode{96}{12}% `
167 \edef\AtBegShi@AtEnd{\AtBegShi@AtEnd\noexpand\endinput}
```

3.3 Preparations

```
168 \begingroup\expandafter\expandafter\expandafter\endgroup
169 \expandafter\ifx\csname RequirePackage\endcsname\relax
170   \def\TMP@RequirePackage#1[#2]{%
171     \begingroup\expandafter\expandafter\expandafter\endgroup
172     \expandafter\ifx\csname ver@#1.sty\endcsname\relax
173       \input #1.sty\relax
174     \fi
175   }%
176   \TMP@RequirePackage{infwarerr}[2007/09/09]%
177   \TMP@RequirePackage{ltxcmds}[2010/03/01]%
178 \else
179   \RequirePackage{infwarerr}[2007/09/09]%
180   \RequirePackage{ltxcmds}[2010/03/01]%
181 \fi
```

\AtBegShi@CheckDefinable

```
182 \begingroup\expandafter\expandafter\expandafter\endgroup
183 \expandafter\ifx\csname @ifdefinable\endcsname\relax
184   \def\AtBegShi@CheckDefinable#1{%
185     \ifcase\ifx#1\relax
186             \ltx@one
187           \else
188             \ifx#1\@undefined
189               \ltx@one
190             \else
191               \ltx@zero
192             \fi
193         \fi
194     \@PackageError{atbegshi}{%
195       \string#1\space is already defined%
196     }\@ehd
197   \fi
198 }%
199 \else
200   \def\AtBegShi@CheckDefinable#1{%
201     \@ifdefinable{#1}{}%
202   }%
203 \fi
```

\ifAtBegShi@Discarded

```
204 \ltx@newif\ifAtBegShi@Discarded
```

\AtBeginShipoutDiscard

```
205 \AtBegShi@CheckDefinable\AtBeginShipoutDiscard
206 \def\AtBeginShipoutDiscard{%
207   \deadcycles=\ltx@zero
208   \global\AtBegShi@Discardedtrue
209 }
```

```
210 \begingroup\expandafter\expandafter\expandafter\endgroup
211 \expandafter\ifx\csname currentgrouplevel\endcsname\relax
212   \catcode`X=9  % ignore
213   \catcode`E=14 % comment
```

13

```
214 \else
215   \catcode`X=14 % comment
216   \catcode`E=9  % ignore
217 \fi
```

\AtBegShi@Shipout

```
218 \def\AtBegShi@Shipout{%
219 X \begingroup
220 X \setbox\AtBeginShipoutBox=\hbox\bgroup
221 X \kern\p@
222 E \edef\AtBegShi@GroupLevel{\number\currentgrouplevel}%
223   \afterassignment\AtBegShi@Test
224 X \global
225   \setbox\AtBeginShipoutBox=%
226 }
```

\AtBegShi@Test

```
227 \def\AtBegShi@Test{%
228 X \ifdim\lastkern=0pt %
229 E \ifnum\AtBegShi@GroupLevel<\currentgrouplevel
230     \expandafter\aftergroup
231   \fi
232   \AtBegShi@Output
233 }
```

\AtBegShi@Output

```
234 \def\AtBegShi@Output{%
235 X \egroup
236 X \endgroup
237   \ifvoid\AtBeginShipoutBox
238     \@PackageWarning{atbegshi}{Ignoring void shipout box}%
239   \else
240     \let\AtBegShi@OrgProtect\protect
241     \csname set@typeset@protect\endcsname
242     \global\AtBegShi@Discardedfalse
243     \AtBegShi@Hook
244     \expandafter\gdef\expandafter\AtBegShi@HookNext
245     \expandafter{\expandafter}%
246     \AtBegShi@HookNext
247     \ifAtBegShi@Discarded
248       \@PackageInfoNoLine{atbegshi}{Shipout page discarded}%
249       \global\AtBegShi@Discardedfalse
250       \begingroup
251         \setbox\AtBeginShipoutBox\box\AtBeginShipoutBox
252       \endgroup
253       \let\protect\AtBegShi@OrgProtect
254     \else
255       \AtBegShi@First
256       \let\protect\AtBegShi@OrgProtect
257       \AtBegShi@GetBoxSize\AtBeginShipoutBox
258       \ltx@ifundefined{AtNextShipout}{%
259       }{%
260         \AtNextShipout{\AtBegShi@GetBoxSize\@cclv}%
261       }%
262       \AtBeginShipoutOriginalShipout\box\AtBeginShipoutBox
263     \fi
264   \fi
265 }
```

\AtBegShi@GetBoxSize

```
266 \def\AtBegShi@GetBoxSize#1{%
267   \xdef\AtBeginShipoutBoxWidth{\the\wd#1}%
```

```
268    \xdef\AtBeginShipoutBoxHeight{\the\ht#1}%
269    \xdef\AtBeginShipoutBoxDepth{\the\dp#1}%
270 }
```

`\AtBeginShipoutBoxWidth`

```
271 \def\AtBeginShipoutBoxWidth{0pt}
```

`\AtBeginShipoutBoxHeight`

```
272 \def\AtBeginShipoutBoxHeight{0pt}
```

`\AtBeginShipoutBoxDepth`

```
273 \def\AtBeginShipoutBoxDepth{0pt}

274 \catcode`\X=11 %
275 \catcode`\E=11 %
```

`\AtBegShi@First`

```
276 \def\AtBegShi@First{%
277    \ifx\AtBegShi@HookFirst\ltx@empty
278    \else
279      \AtBeginShipoutAddToBox{\AtBegShi@HookFirst}%
280    \fi
281    \global\let\AtBegShi@First\ltx@empty
282    \global\let\AtBeginShipoutFirst\AtBegShi@FirstDisabled
283 }
```

`\AtBegShi@Hook`

```
284 \gdef\AtBegShi@Hook{}
```

`\AtBegShi@HookNext`

```
285 \gdef\AtBegShi@HookNext{}
```

`\AtBegShi@HookFirst`

```
286 \gdef\AtBegShi@HookFirst{}
```

`\AtBeginShipout`

```
287 \AtBegShi@CheckDefinable\AtBeginShipout
288 \def\AtBeginShipout{%
289    \AtBegShi@AddHook\AtBegShi@Hook
290 }
```

`\AtBeginShipoutNext`

```
291 \AtBegShi@CheckDefinable\AtBeginShipoutNext
292 \def\AtBeginShipoutNext{%
293    \AtBegShi@AddHook\AtBegShi@HookNext
294 }
```

`\AtBeginShipoutFirst`

```
295 \AtBegShi@CheckDefinable\AtBeginShipoutFirst
296 \def\AtBeginShipoutFirst{%
297    \AtBegShi@AddTo\AtBegShi@HookFirst
298 }
```

`\AtBegShi@FirstDisabled`

```
299 \long\def\AtBegShi@FirstDisabled#1{%
300    \@PackageWarning{atbegshi}{%
301      First page is already shipped out, ignoring\MessageBreak
302      \string\AtBeginShipoutFirst
303    }%
304 }
```

`\AtBegShi@AddTo`

```
305 \begingroup\expandafter\expandafter\expandafter\endgroup
306 \expandafter\ifx\csname g@addto@macro\endcsname\relax
307   \long\def\AtBegShi@AddTo#1#2{%
308     \begingroup
309       \toks\ltx@zero\expandafter{#1#2}%
310       \xdef#1{\the\toks\ltx@zero}%
311     \endgroup
312   }%
313 \else
314   \let\AtBegShi@AddTo\g@addto@macro
315 \fi
```

`\AtBegShi@AddHook`

```
316 \long\def\AtBegShi@AddHook#1#2{%
317   \AtBegShi@AddTo#1{\AtBegShi@Item{#2}}%
318 }
```

`\AtBegShi@Item`

```
319 \long\def\AtBegShi@Item#1{%
320   \ifAtBegShi@Discarded
321   \else
322     #1%
323     \ifAtBegShi@Discarded
324     \else
325       \ifvoid\AtBeginShipoutBox
326         \@PackageWarning{atbegshi}{%
327           Shipout box was voided by hook,\MessageBreak
328           ignoring shipout box%
329         }%
330         \AtBeginShipoutDiscard
331       \fi
332     \fi
333   \fi
334 }
```

`\AtBeginShipoutInit`

```
335 \AtBegShi@CheckDefinable\AtBeginShipoutInit
336 \def\AtBeginShipoutInit{%
337   \ltx@IfUndefined{newbox}{%
338     \@PackageError{atbegshi}{%
339       \string\AtBeginShipoutInit\space failed\MessageBreak
340       because of missing \expandafter\string\csname newbox\endcsname
341     }\@ehc
342   }{%
343     \csname newbox\endcsname\AtBeginShipoutBox
344     \AtBegShi@CheckDefinable\AtBeginShipoutOriginalShipout
345     \global\let\AtBeginShipoutOriginalShipout\shipout
346     \global\let\shipout\AtBegShi@Shipout
347   }%
348   \gdef\AtBeginShipoutInit{}%
349 }
```

```
350 \begingroup\expandafter\expandafter\expandafter\endgroup
351 \expandafter\ifx\csname AtBeginDocument\endcsname\relax
352   \AtBeginShipoutInit
353 \else
354   \AtBeginDocument{\AtBeginShipoutInit}%
355 \fi
```

3.4 Additions to the shipout box

\AtBeginShipoutAddToBox

```
356 \def\AtBeginShipoutAddToBox#1{%
357   \ifhbox\AtBeginShipoutBox
358     \edef\AtBegShi@restore{%
359       \hfuzz=\the\hfuzz\relax
360       \hbadness=\the\hbadness\relax
361     }%
362     \hfuzz=1073741823sp\relax
363     \hbadness=2147483647\relax
364     \setbox\AtBeginShipoutBox=\hbox to \wd\AtBeginShipoutBox{%
365       \setbox\ltx@zero=\hbox{%
366         \begingroup
367           \AtBegShi@restore
368           #1%
369         \endgroup
370       }%
371       \wd\ltx@zero=0pt\relax
372       \ht\ltx@zero=0pt\relax
373       \dp\ltx@zero=0pt\relax
374       \raise\ht\AtBeginShipoutBox\copy\ltx@zero
375       \unhcopy\AtBeginShipoutBox
376     }%
377     \AtBegShi@restore
378   \else
379     \ifvbox\AtBeginShipoutBox
380       \edef\AtBegShi@restore{%
381         \vfuzz=\the\vfuzz\relax
382         \vbadness=\the\vbadness\relax
383         \dimen\ltx@zero=\the\dimen\ltx@zero\relax
384       }%
385       \edef\AtBegShi@restorebox{%
386         \ht\AtBeginShipoutBox=\the\ht\AtBeginShipoutBox\relax
387         \dp\AtBeginShipoutBox=\the\dp\AtBeginShipoutBox\relax
388       }%
389       \vfuzz=1073741823sp\relax
390       \vbadness=2147483647\relax
391       \dimen\ltx@zero=\ht\AtBeginShipoutBox
392       \advance\dimen\ltx@zero by \dp\AtBeginShipoutBox
393       \setbox\AtBeginShipoutBox=\vbox to \dimen\ltx@zero{%
394         \setbox\ltx@zero=\hbox{%
395           \begingroup
396             \AtBegShi@restore
397             #1%
398           \endgroup
399         }%
400         \wd\ltx@zero=0pt\relax
401         \ht\ltx@zero=0pt\relax
402         \dp\ltx@zero=0pt\relax
403         \baselineskip=0pt\relax
404         \lineskip=0pt\relax
405         \lineskiplimit=0pt\relax
406         \copy\ltx@zero
407         \unvbox\AtBeginShipoutBox
408         \kern0pt%
409       }%
410       \AtBegShi@restore
411       \AtBegShi@restorebox
412     \fi
413   \fi
414 }
```

```
415 \def\AtBeginShipoutAddToBoxForeground#1{%
416   \ifhbox\AtBeginShipoutBox
417     \edef\AtBegShi@restore{%
418       \hfuzz=\the\hfuzz\relax
419       \hbadness=\the\hbadness\relax
420     }%
421     \hfuzz=1073741823sp\relax
422     \hbadness=2147483647\relax
423     \setbox\AtBeginShipoutBox=\hbox to \wd\AtBeginShipoutBox{%
424       \unhcopy\AtBeginShipoutBox
425       \kern-\wd\AtBeginShipoutBox
426       \setbox\ltx@zero=\hbox{%
427         \begingroup
428           \AtBegShi@restore
429           #1%
430         \endgroup
431       }%
432       \wd\ltx@zero=0pt\relax
433       \ht\ltx@zero=0pt\relax
434       \dp\ltx@zero=0pt\relax
435       \raise\ht\AtBeginShipoutBox\copy\ltx@zero
436       \kern\wd\AtBeginShipoutBox
437     }%
438     \AtBegShi@restore
439   \else
440     \ifvbox\AtBeginShipoutBox
441       \edef\AtBegShi@restore{%
442         \vfuzz=\the\vfuzz\relax
443         \vbadness=\the\vbadness\relax
444         \dimen\ltx@zero=\the\dimen\ltx@zero\relax
445       }%
446       \edef\AtBegShi@restorebox{%
447         \ht\AtBeginShipoutBox=\the\ht\AtBeginShipoutBox\relax
448         \dp\AtBeginShipoutBox=\the\dp\AtBeginShipoutBox\relax
449       }%
450       \vfuzz=1073741823sp\relax
451       \vbadness=2147483647\relax
452       \dimen\ltx@zero=\ht\AtBeginShipoutBox
453       \advance\dimen\ltx@zero by \dp\AtBeginShipoutBox
454       \setbox\AtBeginShipoutBox=\vbox to \dimen\ltx@zero{%
455         \setbox\ltx@zero=\hbox{%
456           \begingroup
457             \AtBegShi@restore
458             #1%
459           \endgroup
460         }%
461         \wd\ltx@zero=0pt\relax
462         \ht\ltx@zero=0pt\relax
463         \dp\ltx@zero=0pt\relax
464         \baselineskip=0pt\relax
465         \lineskip=0pt\relax
466         \lineskiplimit=0pt\relax
467         \unvbox\AtBeginShipoutBox
468         \kern-\dimen\ltx@zero
469         \copy\ltx@zero
470         \kern\dimen\ltx@zero
471       }%
472       \AtBegShi@restore
473       \AtBegShi@restorebox
474     \fi
475   \fi
```

```
476 }
```

3.5 Positioning

```
477 \begingroup\expandafter\expandafter\expandafter\endgroup
478 \expandafter\ifx\csname RequirePackage\endcsname\relax
479   \def\TMP@RequirePackage#1[#2]{%
480     \begingroup\expandafter\expandafter\expandafter\endgroup
481     \expandafter\ifx\csname ver@#1.sty\endcsname\relax
482       \input #1.sty\relax
483     \fi
484   }%
485   \TMP@RequirePackage{ifpdf}[2011/01/30]%
486 \else
487   \RequirePackage{ifpdf}[2011/01/30]%
488 \fi

489 \ifpdf
490   \def\AtBegShi@horigin{\pdfhorigin}%
491   \def\AtBegShi@vorigin{\pdfvorigin}%
492 \else
493   \def\AtBegShi@horigin{72.27pt}%
494   \def\AtBegShi@vorigin{72.27pt}%
495 \fi

496 \begingroup
497 \ifcase
498     \expandafter\ifx\csname picture\endcsname\relax
499       1%
500     \else
501       \expandafter\ifx\csname endpicture\endcsname\relax
502         1%
503       \else
504         0%
505       \fi
506     \fi
507   \endgroup
508   \def\AtBegShi@@BeginPicture{%
509     \begingroup
510     \picture(0,0)\relax
511     \begingroup\expandafter\expandafter\expandafter\endgroup
512     \expandafter\ifx\csname unitlength\endcsname\relax
513     \else
514       \unitlength=1pt\relax
515     \fi
516     \ignorespaces
517   }%
518   \def\AtBegShi@@EndPicture{%
519     \endpicture
520     \endgroup
521   }%
522 \else
523   \endgroup
524   \def\AtBegShi@@BeginPicture{%
525     \setbox\ltx@zero=\hbox\bgroup
526     \begingroup
527     \ignorespaces
528   }%
529   \def\AtBegShi@@EndPicture{%
530     \endgroup
531     \egroup
532     \ht\ltx@zero=0pt\relax
533     \dp\ltx@zero=0pt\relax
534     \copy\ltx@zero
```

```
535    }%
536 \fi
```

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

```
537 \def\AtBeginShipoutUpperLeft#1{%
538   \AtBeginShipoutAddToBox{%
539     \kern-\AtBegShi@horigin\relax
540     \vbox to 0pt{%
541       \kern-\AtBegShi@vorigin\relax
542       \AtBegShi@BeginPicture
543       #1%
544       \AtBegShi@EndPicture
545       \vss
546     }%
547   }%
548 }
```

ginShipoutUpperLeftForeground

```
549 \def\AtBeginShipoutUpperLeftForeground#1{%
550   \AtBeginShipoutAddToBoxForeground{%
551     \kern-\AtBegShi@horigin\relax
552     \vbox to 0pt{%
553       \kern-\AtBegShi@vorigin\relax
554       \AtBegShi@BeginPicture
555       #1%
556       \AtBegShi@EndPicture
557       \vss
558     }%
559   }%
560 }
```

3.6 Patches

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

```
561 \def\AtBegShi@AbortIfUndefined#1{%
562   \begingroup\expandafter\expandafter\expandafter\endgroup
563   \expandafter\ifx\csname#1\endcsname\relax
564     \expandafter\AtBegShi@AtEnd
565   \fi
566 }
567 \AtBegShi@AbortIfUndefined{currentgrouplevel}%
568 \AtBegShi@AbortIfUndefined{AtBeginDocument}%
569 \AtBegShi@AbortIfUndefined{@ifpackageloaded}%
570 \AtBegShi@AbortIfUndefined{@ifclassloaded}%
```

3.6.1 Package **crop**

Fix of method and box.

```
571 \def\AtBegShi@PatchCrop{%
572   \begingroup
573     \def\AtBegShi@Crop@shipout{%
574       \afterassignment\CROP@ship
575       \setbox\@cclv=%
576     }%
577     \def\AtBegShi@Crop@ship{%
578       \ifvoid\@cclv
579         \expandafter\aftergroup
580       \fi
581       \CROP@@ship
582     }%
```

```
583    \def\AtBegShi@Crop@shiplist{%
584      \lineskip\z@
585      \lineskiplimit\z@
586      \baselineskip\z@
587      \CROP@kernel
588      \box\@cclv
589    }%
590    \def\AtBegShi@Crop@@ship{%
591      \CROP@shipout\vbox{%
592        \CROP@shiplist
593      }%
594    }%
595    \ifx\AtBegShi@Crop@ship\CROP@ship
596      \ifx\AtBegShi@Crop@shiplist\CROP@shiplist
597        \ifx\AtBegShi@Crop@@ship\CROP@@ship
598          \let\AtBegShi@found\relax
599          \ifx\shipout\AtBegShi@Crop@shipout
600            \def\AtBegShi@found{\shipout}%
601          \else\ifx\AtBeginShipoutOriginalShipout\AtBegShi@Crop@shipout
602            \def\AtBegShi@found{\AtBeginShipoutOriginalShipout}%
603          \else\ifx\@EveryShipout@Org@Shipout\AtBegShi@Crop@shipout
604            \def\AtBegShi@found{\@EveryShipout@Org@Shipout}%
605          \else\ifx\GPTorg@shipout\AtBegShi@Crop@shipout
606            \def\AtBegShi@found{\GPTorg@shipout}%
607          \else\ifx\THBorg@shipout\AtBegShi@Crop@shipout
608            \def\AtBegShi@found{\THBorg@shipout}%
609          \else\ifx\mem@oldshipout\AtBegShi@Crop@shipout
610            \def\AtBegShi@found{\mem@oldshipout}%
611          \fi\fi\fi\fi\fi\fi
612          \ifx\AtBegShi@found\relax
613          \else
614            \expandafter\endgroup
615            \expandafter\def\AtBegShi@found{%
616              \edef\AtBegShi@GroupLevel{\number\currentgrouplevel}%
617              \afterassignment\CROP@ship
618              \setbox\AtBeginShipoutBox=%
619            }%
620            \def\CROP@ship{%
621              \ifnum\AtBegShi@GroupLevel=\currentgrouplevel
622              \else
623                \expandafter\aftergroup
624              \fi
625              \CROP@@ship
626            }%
627            \def\CROP@shiplist{%
628              \lineskip 0pt\relax
629              \lineskiplimit 0pt\relax
630              \baselineskip 0pt\relax
631              \CROP@kernel
632              \box\AtBeginShipoutBox
633            }%
634            \def\CROP@@ship{%
635              \ifvoid\AtBeginShipoutBox
636              \else
637                \setbox\AtBeginShipoutBox=\vbox{%
638                  \CROP@shiplist
639                }%
640                \AtBegShi@GetBoxSize\AtBeginShipoutBox
641                \expandafter\CROP@shipout
642                \expandafter\box
643                \expandafter\AtBeginShipoutBox
644              \fi
```

```
645            }%
646            \@PackageInfoNoLine{atbegshi}{Package `crop' patched}%
647            \begingroup
648          \fi
649        \fi
650      \fi
651    \fi
652    \endgroup
653    \let\AtBegShi@PatchCrop\relax
654 }
655 \@ifpackageloaded{crop}{%
656    \AtBegShi@PatchCrop
657 }{%
658    \AtBeginDocument{\AtBegShi@PatchCrop}%
659 }
```

3.6.2 Package **everyshi**

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

```
660 \def\AtBegShi@PatchEveryshi{%
661    \begingroup
662      \long\def\AtBegShi@Everyshi@shipout{%
663        \afterassignment\@EveryShipout@Test
664        \global\setbox\@cclv= %
665      }%
666      \long\def\AtBegShi@Everyshi@Test{%
667        \ifvoid\@cclv\relax
668          \aftergroup\@EveryShipout@Output
669        \else
670          \@EveryShipout@Output
671        \fi
672      }%
673      \ifx\AtBegShi@Everyshi@Test\@EveryShipout@Test
674        \let\AtBegShi@found\relax
675        \ifx\shipout\AtBegShi@Everyshi@shipout
676          \def\AtBegShi@found{\shipout}%
677        \else\ifx\AtBeginShipoutOriginalShipout\AtBegShi@Everyshi@shipout
678          \def\AtBegShi@found{\AtBeginShipoutOriginalShipout}%
679        \else\ifx\CROP@shipout\AtBegShi@Everyshi@shipout
680          \def\AtBegShi@found{\CROP@shipout}%
681        \else\ifx\GPTorg@shipout\AtBegShi@Everyshi@shipout
682          \def\AtBegShi@found{\GPTorg@shipout}%
683        \else\ifx\THBorg@shipout\AtBegShi@Everyshi@shipout
684          \def\AtBegShi@found{\THBorg@shipout}%
685        \else\ifx\mem@oldshipout\AtBegShi@Everyshi@shipout
686          \def\AtBegShi@found{\mem@oldshipout}%
687        \else
688          \expandafter\ifx\csname @EveryShipout@Org@Shipout\endcsname
689                          \relax
690            \ifx\@EveryShipout@Shipout\AtBegShi@Everyshi@shipout
691              \def\AtBegShi@found{\@EveryShipout@Shipout}%
692            \fi
693          \fi
694        \fi\fi\fi\fi\fi\fi
695        \ifx\AtBegShi@found\relax
696        \else
697          \expandafter\endgroup
698          \expandafter\def\AtBegShi@found{%
699            \edef\AtBegShi@GroupLevel{\number\currentgrouplevel}%
700            \afterassignment\@EveryShipout@Test
701            \setbox\AtBeginShipoutBox=%
702          }%
```

```
703        \def\@EveryShipout@Test{%
704          \ifnum\AtBegShi@GroupLevel=\currentgrouplevel
705          \else
706            \expandafter\aftergroup
707          \fi
708          \AtBegShi@Everyshi@Output
709        }%
710        \def\AtBegShi@Everyshi@Output{%
711          \ifvoid\AtBeginShipoutBox
712          \else
713            \global\setbox\ltx@cclv\box\AtBeginShipoutBox
714            \expandafter\@EveryShipout@Output
715          \fi
716        }%
717        \@PackageInfoNoLine{atbegshi}{Package `everyshi' patched}%
718        \begingroup
719      \fi
720    \fi
721    \endgroup
722    \let\AtBegShi@PatchEveryshi\relax
723 }
724 \@ifpackageloaded{everyshi}{%
725   \AtBegShi@PatchEveryshi
726 }{%
727   \AtBeginDocument{\AtBegShi@PatchEveryshi}%
728 }
```

3.6.3 Class **memoir**

Fix of method and box.

```
729 \def\AtBegShi@PatchMemoir{%
730   \begingroup
731     \def\AtBegShi@Memoir@shipout{%
732       \afterassignment\mem@shipi
733       \setbox\@cclv=%
734     }%
735     \def\AtBegShi@Memoir@shipi{%
736       \ifvoid\@cclv
737         \expandafter\aftergroup
738       \fi
739       \mem@shipii
740     }%
741     \def\AtBegShi@Memoir@shipiiA{%
742       \mem@oldshipout\vbox{%
743         \trimmarks
744         \unvbox\@cclv
745       }%
746     }%
747     \def\AtBegShi@Memoir@shipiiB{%
748       \ifvoid\@cclv
749         \mem@oldshipout\box\@cclv
750       \else
751         \mem@oldshipout\vbox{%
752           \trimmarks
753           \unvbox\@cclv
754         }%
755       \fi
756     }%
757     \def\AtBegShi@Memoir@PatchAB{%
758       \ifvoid\AtBeginShipoutBox
759       \else
760         \setbox\AtBeginShipoutBox=\vbox{%
```

```
761        \trimmarks
762        \ifvbox\AtBeginShipoutBox
763          \unvbox\AtBeginShipoutBox
764        \else
765          \box\AtBeginShipoutBox
766        \fi
767      }%
768      \AtBegShi@GetBoxSize\AtBeginShipoutBox
769      \expandafter\mem@oldshipout
770      \expandafter\box
771      \expandafter\AtBeginShipoutBox
772    \fi
773  }%
774  \def\AtBegShi@Memoir@shipiiC{% 2008/08/07 v1.6180339a
775    \ifvoid\@cclv
776      \mem@oldshipout\box\@cclv
777    \else
778      \ifshowtrims
779        \mem@oldshipout\vbox{\trimmarks\unvbox\@cclv}%
780      \else
781        \mem@oldshipout\box\@cclv
782      \fi
783    \fi
784  }%
785  \def\AtBegShi@Memoir@shipiiD{% 2011/03/06 v3.6j
786    \ifvoid\@cclv
787      \mem@oldshipout\box\@cclv
788    \else
789      \ifshowtrims
790        \mem@oldshipout\vbox{%
791          \trimmarks
792          \nointerlineskip
793          \box\@cclv
794        }%
795      \else
796        \mem@oldshipout\box\@cclv
797      \fi
798    \fi
799  }%
800  \def\AtBegShi@Memoir@PatchCD{%
801    \ifvoid\AtBeginShipoutBox
802    \else
803      \ifshowtrims
804        \setbox\AtBeginShipoutBox=\vbox{%
805          \trimmarks
806          \nointerlineskip
807          \box\AtBeginShipoutBox
808        }%
809      \fi
810      \AtBegShi@GetBoxSize\AtBeginShipoutBox
811      \expandafter\mem@oldshipout
812      \expandafter\box
813      \expandafter\AtBeginShipoutBox
814    \fi
815  }%
816  \ifx\AtBegShi@Memoir@shipi\mem@shipi
817    \let\AtBegShi@found\ltx@one
818    \ifx\AtBegShi@Memoir@shipiiA\mem@shipii
819      \let\AtBegShi@found\ltx@zero
820      \global\let\AtBegShi@Memoir@PatchX\AtBegShi@Memoir@PatchAB
821    \else\ifx\AtBegShi@Memoir@shipiiB\mem@shipii
822      \let\AtBegShi@found\ltx@zero
```

```
823        \global\let\AtBegShi@Memoir@PatchX\AtBegShi@Memoir@PatchAB
824      \else\ifx\AtBegShi@Memoir@shipiiC\mem@shipii
825        \let\AtBegShi@found\ltx@zero
826        \global\let\AtBegShi@Memoir@PatchX\AtBegShi@Memoir@PatchCD
827      \else\ifx\AtBegShi@Memoir@shipiiD\mem@shipii
828        \let\AtBegShi@found\ltx@zero
829        \global\let\AtBegShi@Memoir@PatchX\AtBegShi@Memoir@PatchCD
830      \fi\fi\fi\fi
831      \ifcase\AtBegShi@found
832        \let\AtBegShi@found\relax
833        \ifx\shipout\AtBegShi@Memoir@shipout
834          \def\AtBegShi@found{\shipout}%
835        \else\ifx\AtBeginShipoutOriginalShipout\AtBegShi@Memoir@shipout
836          \def\AtBegShi@found{\AtBeginShipoutOriginalShipout}%
837        \else\ifx\CROP@shipout\AtBegShi@Memoir@shipout
838          \def\AtBegShi@found{\CROP@shipout}%
839        \else\ifx\GPTorg@shipout\AtBegShi@Memoir@shipout
840          \def\AtBegShi@found{\GPTorg@shipout}%
841        \else\ifx\THBorg@shipout\AtBegShi@Memoir@shipout
842          \def\AtBegShi@found{\THBorg@shipout}%
843        \else\ifx\@EveryShipout@Org@Shipout\AtBegShi@Memoir@shipout
844          \def\AtBegShi@found{\@EveryShipout@Org@Shipout}%
845        \fi\fi\fi\fi\fi\fi
846        \ifx\AtBegShi@found\relax
847        \else
848          \expandafter\endgroup
849          \expandafter\def\AtBegShi@found{%
850            \edef\AtBegShi@GroupLevel{\number\currentgrouplevel}%
851            \afterassignment\mem@shipi
852            \setbox\AtBeginShipoutBox=%
853          }%
854          \def\mem@shipi{%
855            \ifnum\AtBegShi@GroupLevel=\currentgrouplevel
856            \else
857              \expandafter\aftergroup
858            \fi
859            \mem@shipii
860          }%
861          \let\mem@shipii\AtBegShi@Memoir@PatchX
862          \@PackageInfoNoLine{atbegshi}{Class `memoir' patched}%
863          \begingroup
864        \fi
865      \fi
866    \fi
867  \endgroup
868  \let\AtBegShi@PatchMemoir\relax
869 }
870 \@ifclassloaded{memoir}{%
871   \AtBegShi@PatchMemoir
872 }{%
873   \AtBeginDocument{\AtBegShi@PatchMemoir}%
874 }
875 \AtBegShi@AtEnd%
876 ⟨/package⟩
```

4 Test

4.1 Catcode checks for loading

```
877 ⟨*test1⟩
878 \catcode`\{=1 %
```

```
879 \catcode`\}=2 %
880 \catcode`\#=6 %
881 \catcode`\@=11 %
882 \expandafter\ifx\csname count@\endcsname\relax
883   \countdef\count@=255 %
884 \fi
885 \expandafter\ifx\csname @gobble\endcsname\relax
886   \long\def\@gobble#1{}%
887 \fi
888 \expandafter\ifx\csname @firstofone\endcsname\relax
889   \long\def\@firstofone#1{#1}%
890 \fi
891 \expandafter\ifx\csname loop\endcsname\relax
892   \expandafter\@firstofone
893 \else
894   \expandafter\@gobble
895 \fi
896 {%
897   \def\loop#1\repeat{%
898     \def\body{#1}%
899     \iterate
900   }%
901   \def\iterate{%
902     \body
903       \let\next\iterate
904     \else
905       \let\next\relax
906     \fi
907     \next
908   }%
909   \let\repeat=\fi
910 }%
911 \def\RestoreCatcodes{}
912 \count@=0 %
913 \loop
914   \edef\RestoreCatcodes{%
915     \RestoreCatcodes
916     \catcode\the\count@=\the\catcode\count@\relax
917   }%
918 \ifnum\count@<255 %
919   \advance\count@ 1 %
920 \repeat
921
922 \def\RangeCatcodeInvalid#1#2{%
923   \count@=#1\relax
924   \loop
925     \catcode\count@=15 %
926   \ifnum\count@<#2\relax
927     \advance\count@ 1 %
928   \repeat
929 }
930 \def\RangeCatcodeCheck#1#2#3{%
931   \count@=#1\relax
932   \loop
933     \ifnum#3=\catcode\count@
934     \else
935       \errmessage{%
936         Character \the\count@\space
937         with wrong catcode \the\catcode\count@\space
938         instead of \number#3%
939       }%
940     \fi
```

```
941   \ifnum\count@<#2\relax
942     \advance\count@ 1 %
943   \repeat
944 }
945 \def\space{ }
946 \expandafter\ifx\csname LoadCommand\endcsname\relax
947   \def\LoadCommand{\input atbegshi.sty\relax}%
948 \fi
949 \def\Test{%
950   \RangeCatcodeInvalid{0}{47}%
951   \RangeCatcodeInvalid{58}{64}%
952   \RangeCatcodeInvalid{91}{96}%
953   \RangeCatcodeInvalid{123}{255}%
954   \catcode`\@=12 %
955   \catcode`\\=0 %
956   \catcode`\%=14 %
957   \LoadCommand
958   \RangeCatcodeCheck{0}{36}{15}%
959   \RangeCatcodeCheck{37}{37}{14}%
960   \RangeCatcodeCheck{38}{47}{15}%
961   \RangeCatcodeCheck{48}{57}{12}%
962   \RangeCatcodeCheck{58}{63}{15}%
963   \RangeCatcodeCheck{64}{64}{12}%
964   \RangeCatcodeCheck{65}{90}{11}%
965   \RangeCatcodeCheck{91}{91}{15}%
966   \RangeCatcodeCheck{92}{92}{0}%
967   \RangeCatcodeCheck{93}{96}{15}%
968   \RangeCatcodeCheck{97}{122}{11}%
969   \RangeCatcodeCheck{123}{255}{15}%
970   \RestoreCatcodes
971 }
972 \Test
973 \csname @@end\endcsname
974 \end
975 ⟨/test1⟩
976 ⟨*test2⟩
977 \input atbegshi.sty\relax
978 \def\msg#{\immediate\write16}
979 \msg{File: atbegshi-test2.tex 2011/10/05 v1.16 Test file for plain-TeX}
980 \def\testmsg#1#2{%
981   \msg{}%
982   \msg{*** Test with box (#1), expected page output [#2]}% hash-ok
983 }
984
985 \newbox\voidbox
986 \def\void{\box\voidbox}
987 \begingroup
988   \setbox\voidbox=\void
989 \endgroup
990
991 \count0=0\relax
992 \AtBeginShipout{%
993   \global\advance\count0 by 1\relax
994   \msg{* Inside \string\AtBeginShipout: [\the\count0]}%
995 }
996
997 \AtBeginShipoutFirst{%
998   \msg{* Inside \string\AtBeginShipoutFirst}%
999   Hello World%
1000 }
1001
1002 \testmsg{\string\null}{1}
```

27

```
1003 \shipout\null
1004
1005 \AtBeginShipoutFirst{%
1006   This is too late%
1007 }
1008
1009 \testmsg{void}{}
1010 \shipout\void
1011
1012 \testmsg{\string\copy255 (not void)}{2}
1013 \setbox255\hbox{\vrule height 10bp width 10bp}
1014 \shipout\copy255 %
1015
1016 \testmsg{\string\copy255 (again)}{3}
1017 \shipout\copy255 %
1018
1019 \testmsg{\string\box255}{4}
1020 \shipout\box255 %
1021
1022 \testmsg{\string\box255 (again)}{}
1023 \shipout\box255 %
1024
1025 \testmsg{\string\hbox}{5}
1026 \shipout\hbox{\vrule height 5bp width 20bp}
1027
1028 \testmsg{\string\vbox}{6}
1029 \shipout\vbox{\hrule height 20bp width 5bp}
1030
1031 \testmsg{\string\null, voided by hook}{}
1032 \def\VoidBox{%
1033   \begingroup
1034     \setbox\AtBeginShipoutBox=\box\AtBeginShipoutBox
1035   \endgroup
1036 }
1037 \AtBeginShipout{\VoidBox}
1038 \shipout\null
1039 \def\VoidBox{}
1040
1041 \msg{*** \string\begingroup}
1042 \begingroup
1043   \testmsg{void}{}%
1044   \shipout\void
1045 \msg{*** \string\endgroup}
1046 \endgroup
1047
1048 \msg{*** \string\begingroup}
1049 \begingroup
1050   \testmsg{void}{}%
1051   \shipout\void
1052   \testmsg{\string\null}{8}%
1053   \shipout\null
1054 \msg{*** \string\endgroup}
1055 \endgroup
1056
1057 \testmsg{output routine}{9}
1058 Hello World
1059 \vfill
1060 \eject
1061
1062 \testmsg{\string\null\space(discarded)}{}
1063 \AtBeginShipout{%
1064   \msg{* Inside \string\AtBeginShipout: DISCARD}%
```

```
1065     \AtBeginShipoutDiscard
1066 }
1067 \shipout\null
1068
1069 \end
1070 ⟨/test2⟩

1071 ⟨*test3⟩
1072 \NeedsTeXFormat{LaTeX2e}
1073 \ProvidesFile{atbegshi-test3.tex}[2011/10/05 v1.16 Test file for LaTeX]
1074 \RequirePackage{color}
1075 \pagecolor{yellow}
1076 \documentclass[a5paper,showtrims]{memoir}
1077 \usepackage{atbegshi}
1078 \AtBeginShipout{%
1079   \setbox\AtBeginShipoutBox=\vbox{%
1080     \vbox to 0pt{%
1081       \kern-1.5in %
1082       \hbox to 0pt{%
1083         \kern-1.5in %
1084         \color{blue}%
1085         \rule{1in}{1in}%
1086         \hss
1087       }%
1088       \vss
1089     }%
1090     \hrule
1091     \hbox{\vrule\box\AtBeginShipoutBox\vrule}%
1092     \hrule
1093   }%
1094 }
1095 \usepackage{eso-pic}
1096 \makeatletter
1097 \@ifundefined{@EveryShipout@Init}{%
1098   \typeout{Test skipped}%
1099   \@@end
1100 }{}
1101 \@EveryShipout@Init
1102 \let\@EveryShipout@Init\relax
1103 \makeatother
1104 \AddToShipoutPicture{%
1105   \hspace{.52\paperwidth}%
1106   \colorbox{cyan}{%
1107     \rule{0mm}{\paperheight}%
1108     \hspace{.48\paperwidth}%
1109   }%
1110 }
```

Newer versions of class memoir emulate package crop and prevents its loading.
This is undone in next line for this test file.

```
1111 \expandafter\let\csname ver@crop.sty\endcsname\relax
1112 \usepackage[color=red,cross,a4,center]{crop}
1113 \begin{document}
1114 \shipout\null
1115 \shipout\box\csname voidb@x\endcsname
1116 \section{Hello World}
1117 \end{document}
1118 ⟨/test3⟩
```

5 Installation

5.1 Download

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

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

`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.

`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 `texmf` in their name are usually organized this way.

5.2 Bundle installation

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

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

Script installation. Check the directory `TDS:scripts/oberdiek/` for scripts that need further installation steps. Package `attachfile2` comes with the Perl script `pdfatfi.pl` that should be installed in such a way that it can be called as `pdfatfi`. Example (linux):

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

5.3 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/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
```

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`.

5.4 Refresh file name databases

If your TeX distribution (teTeX, mikTeX, …) relies on file name databases, you must refresh these. For example, teTeX users run `texhash` or `mktexlsr`.

---

[1] `ftp://ftp.ctan.org/tex-archive/`

5.5 Some details for the interested

Attached source. The PDF documentation on CTAN also includes the `.dtx` source file. It can be extracted by AcrobatReader 6 or higher. Another option is `pdftk`, e.g. unpack the file into the current directory:

```
pdftk atbegshi.pdf unpack_files output .
```

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 pdfLaTeX:

```
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`.

```
1119 ⟨*catalogue⟩
1120 <?xml version='1.0' encoding='us-ascii'?>
1121 <!DOCTYPE entry SYSTEM 'catalogue.dtd'>
1122 <entry datestamp='$Date$' modifier='$Author$' id='atbegshi'>
1123   <name>atbegshi</name>
1124   <caption>Execute stuff at \shipout time.</caption>
1125   <authorref id='auth:oberdiek'/>
1126   <copyright owner='Heiko Oberdiek' year='2007-2011'/>
1127   <license type='lppl1.3'/>
1128   <version number='1.16'/>
1129   <description>
1130     This package is a modern reimplementation of package
1131     <xref refid='everyshi'>everyshi</xref>, providing various commands
1132     to be executed before a <tt>\shipout</tt> command.  It makes use of
1133     e-TeX&#x2019;s facilities if they are available.  The package may
1134     be used either with LaTeX or with plain TeX.
1135     <p/>
1136     The package is part of the <xref refid='oberdiek'>oberdiek</xref> bundle.
1137   </description>
1138   <documentation details='Package documentation'
1139       href='ctan:/macros/latex/contrib/oberdiek/atbegshi.pdf'/>
```

```
1140    <ctan file='true' path='/macros/latex/contrib/oberdiek/atbegshi.dtx'/>
1141    <miktex location='oberdiek'/>
1142    <texlive location='oberdiek'/>
1143    <install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip'/>
1144 </entry>
1145 ⟨/catalogue⟩
```

7 History

[2007/04/17 v1.0]

- First version.

[2007/04/18 v1.1]

- New method based on `\lastkern` is used if ε-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.

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.

34

35

36