

The atbegshi package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2007/06/06 v1.5

Abstract

This package is a modern reimplementaion of package `everyshi` without the burden of compatibility. It makes use of ε -TeX's if available. Both L^AT_EX and plain-T_EX are supported.

Contents

1	Documentation	2
1.1	Example	3
2	Method of <code>\shipout</code> overloading	4
2.1	<code>\shipout</code>	4
2.2	<code>\afterassignment</code>	4
2.3	Test for direct or indirect boxes	5
2.3.1	With ε -TeX	5
2.3.2	Without ε -TeX	5
2.3.3	<code>\lastkern</code> method	6
2.4	Output	7
2.5	Separate box register	7
2.6	Summary	7
2.6.1	With ε -TeX	7
2.6.2	Without ε -TeX, traditional way	8
2.6.3	<code>\lastkern</code> method	9
3	Implementation	9
3.1	Reload check and package identification	9
3.2	Catcodes	10
3.3	Preparations	11
3.4	Positioning	14
3.5	Patches	15
3.5.1	Package <code>crop</code>	15
3.5.2	Package <code>everyshi</code>	17
3.5.3	Class <code>memoir</code>	18
4	Test	20
5	Installation	22
5.1	Download	22
5.2	Bundle installation	22
5.3	Package installation	23
5.4	Refresh file name databases	23
5.5	Some details for the interested	23

6 History	24
[2007/04/17 v1.0]	24
[2007/04/18 v1.1]	24
[2007/04/19 v1.2]	24
[2007/04/26 v1.3]	24
[2007/04/27 v1.4]	24
[2007/06/06 v1.5]	24
7 Index	24

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⟩}`

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 L^AT_EX'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`.

`\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 pdf_TE_X in PDF mode, the settings of `\pdfhorigin` and `\pdfvorigin` are respected.

For L^AT_EX users the *<background material>* is set inside a `picture` environment:

```
\begin{picture}(0,0)
  \setlength{\unitlength}{1pt}%
  <background material>
\end{picture}
```

1.1 Example

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

```
1 <*example>
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 \end{example}

```

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 caught 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 ⇒ \setbox\mybox=\box0 \@test
```

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

```
\shipout\hbox{Hello World}
⇒ \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 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 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 0pt. 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 0pt.

```
\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 ε -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 ε -T_EX, traditional way

And for T_EX without ε -T_EX:

```

\newbox\mybox
\begin{group}
  \setbox\mybox=\box\mybox % ensure \mybox is void
\end{group}
\let\original@shipout\shipout

\shipout :=
  % trick to get a void box \mybox
  \begin{group}
    \setbox\mybox=\box\mybox
  \end{group}
  \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 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ without $\varepsilon\text{-}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 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 $\varepsilon\text{-}\mathrm{T}_{\mathrm{E}}\mathrm{X}$'s `\currentgrouplevel`, if it is available. Otherwise the `\lastkern` method is used.

```
36 <*package>
```

3.1 Reload check and package identification

Reload check, especially if the package is not used with $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$.

```
37 \begingroup
38 \catcode44 12 % ,
39 \catcode45 12 % -
40 \catcode46 12 % .
41 \catcode58 12 % :
42 \catcode64 11 % @
43 \expandafter\let\expandafter\x\csname ver@atbegshi.sty\endcsname
44 \ifcase 0%
45   \ifx\x\relax % plain
46   \else
47     \ifx\x\empty % LaTeX
48     \else
49       1%
50     \fi
51   \fi
52 \else
53   \expandafter\ifx\csname PackageInfo\endcsname\relax
54     \def\x#1#2{%
```

```

55     \immediate\write-1{Package #1 Info: #2.}%
56   }%
57   \else
58     \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
59   \fi
60   \x{atbegshi}{The package is already loaded}%
61   \endgroup
62   \expandafter\endinput
63 \fi
64 \endgroup

```

Package identification:

```

65 \begingroup
66   \catcode40 12 % (
67   \catcode41 12 % )
68   \catcode44 12 % ,
69   \catcode45 12 % -
70   \catcode46 12 % .
71   \catcode47 12 % /
72   \catcode58 12 % :
73   \catcode64 11 % @
74   \expandafter\ifx\csname ProvidesPackage\endcsname\relax
75     \def\x#1#2#3[#4]{\endgroup
76       \immediate\write-1{Package: #3 #4}%
77       \xdef#1{#4}%
78     }%
79   \else
80     \def\x#1#2[#3]{\endgroup
81       #2[{#3}]%
82       \ifx#1\relax
83         \xdef#1{#3}%
84       \fi
85     }%
86   \fi
87   \expandafter\x\csname ver@atbegshi.sty\endcsname
88   \ProvidesPackage{atbegshi}%
89   [2007/06/06 v1.5 At begin shipout hook (H0)]

```

3.2 Catcodes

```

90 \expandafter\edef\csname AtBegShi@End\endcsname{%
91   \catcode40 \the\catcode40\relax % (
92   \catcode41 \the\catcode41\relax % )
93   \catcode44 \the\catcode44\relax % ,
94   \catcode45 \the\catcode45\relax % -
95   \catcode46 \the\catcode46\relax % .
96   \catcode58 \the\catcode58\relax % :
97   \catcode61 \the\catcode61\relax % =
98   \catcode64 \the\catcode64\relax % @
99   \catcode94 \the\catcode94\relax % ^
100  \catcode96 \the\catcode96\relax % '
101  \noexpand\endinput
102 }
103 \catcode40 12 % (
104 \catcode41 12 % )
105 \catcode44 12 % ,
106 \catcode45 12 % -
107 \catcode46 12 % .
108 \catcode58 12 % :
109 \catcode61 12 % =
110 \catcode64 11 % @
111 \catcode94 7 % ^
112 \catcode96 12 % '

```

3.3 Preparations

```

113 \begingroup\expandafter\expandafter\expandafter\endgroup
114 \expandafter\ifx\csname RequirePackage\endcsname\relax
115   \input infwarerr.sty\relax
116 \else
117   \RequirePackage{infwarerr}%
118 \fi

\AtBegShi@CheckDefinable

119 \begingroup\expandafter\expandafter\expandafter\endgroup
120 \expandafter\ifx\csname @ifdefinable\endcsname\relax
121   \def\AtBegShi@CheckDefinable#1{%
122     \ifcase\ifx#1\relax
123       \@ne
124     \else
125       \ifx#1\@undefined
126         \@ne
127       \else
128         \z@
129       \fi
130     \fi
131     \errmessage{%
132       Package atbegshi: \string#1\space
133       is already defined%
134     }%
135   \endgroup
136 \fi
137 }%
138 \else
139   \def\AtBegShi@CheckDefinable#1{%
140     \@ifdefinable{#1}{}%
141   }%
142 \fi

143 \newif\ifAtBegShi@Discarded

\AtBeginShipoutDiscard

144 \AtBegShi@CheckDefinable\AtBeginShipoutDiscard
145 \def\AtBeginShipoutDiscard{%
146   \deadcycles=\z@
147   \global\AtBegShi@Discardedtrue
148 }

149 \begingroup\expandafter\expandafter\expandafter\endgroup
150 \expandafter\ifx\csname currentgrouplevel\endcsname\relax
151   \catcode'X=9 % ignore
152   \catcode'E=14 % comment
153 \else
154   \catcode'X=14 % comment
155   \catcode'E=9 % ignore
156 \fi

\AtBegShi@Shipout

157 \def\AtBegShi@Shipout{%
158 X \begingroup
159 X \setbox\AtBeginShipoutBox=\hbox\bgroup
160 X \kern\p@
161 E \edef\AtBegShi@GroupLevel{\number\currentgrouplevel}%
162 \afterassignment\AtBegShi@Test
163 X \global
164 \setbox\AtBeginShipoutBox=%
165 }
```

\AtBegShi@Test

```
166 \def\AtBegShi@Test{%
167 X \ifdim\lastkern=\z@
168 E \ifnum\AtBegShi@GroupLevel<\currentgrouplevel
169 \expandafter\aftergroup
170 \fi
171 \AtBegShi@Output
172 }
```

\AtBegShi@Output

```
173 \def\AtBegShi@Output{%
174 X \egroup
175 X \endgroup
176 \ifvoid\AtBeginShipoutBox
177 \@PackageWarning{atbegshi}{Ignoring void shipout box}%
178 \else
179 \let\AtBegShi@OrgProtect\protect
180 \csname set@typeset@protect\endcsname
181 \global\AtBegShi@Discardedfalse
182 \AtBegShi@Hook
183 \AtBegShi@HookNext
184 \gdef\AtBegShi@HookNext{}%
185 \ifAtBegShi@Discarded
186 \@PackageInfoNoLine{atbegshi}{Shipout page discarded}%
187 \global\AtBegShi@Discardedfalse
188 \begingroup
189 \setbox\AtBeginShipoutBox\box\AtBeginShipoutBox
190 \endgroup
191 \let\protect\AtBegShi@OrgProtect
192 \else
193 \AtBegShi@First
194 \let\protect\AtBegShi@OrgProtect
195 \AtBegShi@OrgShipout\box\AtBeginShipoutBox
196 \fi
197 \fi
198 }

199 \catcode'\X=11 %
200 \catcode'\E=11 %
```

\AtBegShi@First

```
201 \def\AtBegShi@First{%
202 \begingroup
203 \def\@empty{}%
204 \ifx\AtBegShi@HookFirst\@empty
205 \else
206 \setbox\z@=\vbox{%
207 \begingroup
208 \AtBegShi@HookFirst
209 \endgroup
210 }%
211 \wd\z@=\z@
212 \ht\z@=\z@
213 \dp\z@=\z@
214 \global\setbox\AtBeginShipoutBox=\vbox{%
215 \baselineskip\z@skip
216 \lineskip\z@skip
217 \lineskiplimit\z@
218 \copy\z@
219 \copy\AtBeginShipoutBox
220 }%
221 \fi
```

```

222     \global\let\AtBegShi@First\@empty
223     \global\let\AtBeginShipoutFirst\AtBegShi@FirstDisabled
224 \endgroup
225 }

\AtBegShi@Hook
226 \gdef\AtBegShi@Hook{}

\AtBegShi@HookNext
227 \gdef\AtBegShi@HookNext{}

\AtBegShi@HookFirst
228 \gdef\AtBegShi@HookFirst{}

\AtBeginShipout
229 \AtBegShi@CheckDefinable\AtBeginShipout
230 \def\AtBeginShipout{%
231     \AtBegShi@AddHook\AtBegShi@Hook
232 }

\AtBeginShipoutNext
233 \AtBegShi@CheckDefinable\AtBeginShipoutNext
234 \def\AtBeginShipoutNext{%
235     \AtBegShi@AddHook\AtBegShi@HookNext
236 }

\AtBeginShipoutFirst
237 \AtBegShi@CheckDefinable\AtBeginShipoutFirst
238 \def\AtBeginShipoutFirst{%
239     \AtBegShi@AddTo\AtBegShi@HookFirst
240 }

\AtBegShi@FirstDisabled
241 \long\def\AtBegShi@FirstDisabled#1{%
242     \@PackageWarning{atbegshi}{%
243         First page is already shipped out, ignoring\MessageBreak
244         \string\AtBeginShipoutFirst
245     }%
246 }

\AtBegShi@AddTo
247 \begingroup\expandafter\expandafter\expandafter\endgroup
248 \expandafter\ifx\csname g@addto@macro\endcsname\relax
249     \long\def\AtBegShi@AddTo#1#2{%
250         \begingroup
251             \toks\z@\expandafter{#1#2}%
252             \xdef#1{\the\toks\z@}%
253         \endgroup
254     }%
255 \else
256     \let\AtBegShi@AddTo\g@addto@macro
257 \fi

\AtBegShi@AddHook
258 \long\def\AtBegShi@AddHook#1#2{%
259     \AtBegShi@AddTo#1{\AtBegShi@Item{#2}}%
260 }

```

\AtBegShi@Item

```
261 \long\def\AtBegShi@Item#1{%
262   \ifAtBegShi@Discarded
263   \else
264     #1%
265     \ifvoid\AtBeginShipoutBox
266       \@PackageWarning{atbegshi}{%
267         Shipout box was voided by hook,\MessageBreak
268         ignoring shipout box%
269       }%
270     \AtBeginShipoutDiscard
271   \fi
272 \fi
273 }
```

\AtBeginShipoutInit

```
274 \AtBegShi@CheckDefinable\AtBeginShipoutInit
275 \def\AtBeginShipoutInit{%
276   \csname newbox\endcsname\AtBeginShipoutBox
277   \AtBegShi@CheckDefinable\AtBegShi@OrgShipout
278   \global\let\AtBegShi@OrgShipout\shipout
279   \global\let\shipout\AtBegShi@Shipout
280   \gdef\AtBeginShipoutInit{}%
281 }

282 \begingroup\expandafter\expandafter\expandafter\endgroup
283 \expandafter\ifx\csname AtBeginDocument\endcsname\relax
284   \AtBeginShipoutInit
285 \else
286   \AtBeginDocument{\AtBeginShipoutInit}%
287 \fi
```

3.4 Positioning

```
288 \begingroup\expandafter\expandafter\expandafter\endgroup
289 \expandafter\ifx\csname RequirePackage\endcsname\relax
290   \input ifpdf.sty\relax
291 \else
292   \RequirePackage{ifpdf}\relax
293 \fi

294 \ifpdf
295   \def\AtBegShi@horigin{\pdfhorigin}%
296   \def\AtBegShi@vorigin{\pdfvorigin}%
297 \else
298   \def\AtBegShi@horigin{72.27pt}%
299   \def\AtBegShi@vorigin{72.27pt}%
300 \fi

301 \begingroup
302 \ifcase
303   \expandafter\ifx\csname picture\endcsname\relax
304     1%
305   \else
306     \expandafter\ifx\csname endpicture\endcsname\relax
307       1%
308     \else
309       0%
310     \fi
311   \fi
312 \endgroup
313 \def\AtBegShi@BeginPicture{%
314   \begingroup
```

```

315     \picture(0,0)\relax
316     \begingroup\expandafter\expandafter\expandafter\endgroup
317     \expandafter\ifx\csname unitlength\endcsname\relax
318     \else
319         \unitlength=1pt\relax
320     \fi
321     \ignorespaces
322 }%
323 \def\AtBegShi@EndPicture{%
324     \endpicture
325     \endgroup
326 }%
327 \else
328     \endgroup
329     \def\AtBegShi@BeginPicture{%
330         \setbox0=\hbox\bgroup
331         \begingroup
332         \ignorespaces
333     }%
334     \def\AtBegShi@EndPicture{%
335         \endgroup
336         \egroup
337         \ht0=0pt\relax
338         \dp0=0pt\relax
339         \copy0 %
340     }%
341 \fi

342 \def\AtBeginShipoutUpperLeft#1{%
343     \global\setbox\AtBeginShipoutBox=\hbox{%
344         \rlap{%
345             \kern-\AtBegShi@horigin\relax
346             \vbox to 0pt{%
347                 \kern-\AtBegShi@vorigin\relax
348                 \kern-\ht\AtBeginShipoutBox
349                 \AtBegShi@BeginPicture
350                 #1%
351                 \AtBegShi@EndPicture
352                 \vss
353             }%
354         }%
355     \box\AtBeginShipoutBox
356 }%
357 }

```

3.5 Patches

Patches for L^AT_EX packages that redefine `\shipout`. L^AT_EX is now supposed to use ε -T_EX. Thus we do not patch, without L^AT_EX and ε -T_EX.

```

358 \def\AtBegShi@AbortIfUndefined#1{%
359     \begingroup\expandafter\expandafter\expandafter\endgroup
360     \expandafter\ifx\csname#1\endcsname\relax
361         \expandafter\AtBegShi@End
362     \fi
363 }
364 \AtBegShi@AbortIfUndefined{currentgrouplevel}
365 \AtBegShi@AbortIfUndefined{AtBeginDocument}
366 \AtBegShi@AbortIfUndefined{@ifpackageloaded}
367 \AtBegShi@AbortIfUndefined{@ifclassloaded}

```

3.5.1 Package crop

Fix of method and box.

```

368 \def\AtBegShi@PatchCrop{%
369   \begingroup
370   \def\AtBegShi@Crop@shipout{%
371     \afterassignment\CROP@ship
372     \setbox\@cclv=%
373   }%
374   \def\AtBegShi@Crop@ship{%
375     \ifvoid\@cclv
376       \expandafter\aftergroup
377       \fi
378     \CROP@@ship
379   }%
380   \def\AtBegShi@Crop@shiplist{%
381     \lineskip\z@
382     \lineskiplimit\z@
383     \baselineskip\z@
384     \CROP@kernel
385     \box\@cclv
386   }%
387   \def\AtBegShi@Crop@@ship{%
388     \CROP@shipout\vbox{%
389       \CROP@shiplist
390     }%
391   }%
392   \ifx\AtBegShi@Crop@ship\CROP@ship
393     \ifx\AtBegShi@Crop@shiplist\CROP@shiplist
394       \ifx\AtBegShi@Crop@@ship\CROP@@ship
395         \let\AtBegShi@found\relax
396         \ifx\shipout\AtBegShi@Crop@shipout
397           \def\AtBegShi@found{\shipout}%
398         \else\ifx\AtBegShi@OrgShipout\AtBegShi@Crop@shipout
399           \def\AtBegShi@found{\AtBegShi@OrgShipout}%
400         \else\ifx\@EveryShipout@Org@Shipout\AtBegShi@Crop@shipout
401           \def\AtBegShi@found{\@EveryShipout@Org@Shipout}%
402         \else\ifx\GPTorg@shipout\AtBegShi@Crop@shipout
403           \def\AtBegShi@found{\GPTorg@shipout}%
404         \else\ifx\THBorg@shipout\AtBegShi@Crop@shipout
405           \def\AtBegShi@found{\THBorg@shipout}%
406         \else\ifx\mem@oldshipout\AtBegShi@Crop@shipout
407           \def\AtBegShi@found{\mem@oldshipout}%
408         \fi\fi\fi\fi\fi\fi
409         \ifx\AtBegShi@found\relax
410           \else
411             \expandafter\endgroup
412             \expandafter\def\AtBegShi@found{%
413               \edef\AtBegShi@GroupLevel{\number\currentgrouplevel}%
414               \afterassignment\CROP@ship
415               \setbox\AtBeginShipoutBox=%
416             }%
417             \def\CROP@ship{%
418               \ifnum\AtBegShi@GroupLevel=\currentgrouplevel
419               \else
420                 \expandafter\aftergroup
421                 \fi
422               \CROP@@ship
423             }%
424             \def\CROP@shiplist{%
425               \lineskip\z@
426               \lineskiplimit\z@
427               \baselineskip\z@
428               \CROP@kernel
429               \box\AtBeginShipoutBox

```



```

430         }%
431         \def\CROP@ship{%
432             \ifvoid\AtBeginShipoutBox
433             \else
434                 \setbox\AtBeginShipoutBox=\vbox{%
435                     \CROP@shiplist
436                 }%
437                 \expandafter\CROP@shipout
438                 \expandafter\box
439                 \expandafter\AtBeginShipoutBox
440             \fi
441         }%
442         \@PackageInfoNoLine{atbegshi}{Package 'crop' patched}%
443         \begingroup
444         \fi
445     \fi
446 \fi
447 \fi
448 \endgroup
449 \let\AtBegShi@PatchCrop\relax
450 }
451 \ifpackageloaded{crop}{%
452     \AtBegShi@PatchCrop
453 }{%
454     \AtBeginDocument{\AtBegShi@PatchCrop}%
455 }

```

3.5.2 Package everyshi

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

```

456 \def\AtBegShi@PatchEveryshi{%
457     \begingroup
458     \long\def\AtBegShi@Everyshi@shipout{%
459         \afterassignment\@EveryShipout@Test
460         \global\setbox\@cclv= %
461     }%
462     \long\def\AtBegShi@Everyshi@Test{%
463         \ifvoid\@cclv\relax
464             \aftergroup\@EveryShipout@Output
465         \else
466             \@EveryShipout@Output
467         \fi
468     }%
469     \ifx\AtBegShi@Everyshi@Test\@EveryShipout@Test
470         \let\AtBegShi@found\relax
471         \ifx\shipout\AtBegShi@Everyshi@shipout
472             \def\AtBegShi@found{\shipout}%
473         \else\ifx\AtBegShi@OrgShipout\AtBegShi@Everyshi@shipout
474             \def\AtBegShi@found{\AtBegShi@OrgShipout}%
475         \else\ifx\CROP@shipout\AtBegShi@Everyshi@shipout
476             \def\AtBegShi@found{\CROP@shipout}%
477         \else\ifx\GPTorg@shipout\AtBegShi@Everyshi@shipout
478             \def\AtBegShi@found{\GPTorg@shipout}%
479         \else\ifx\THBorg@shipout\AtBegShi@Everyshi@shipout
480             \def\AtBegShi@found{\THBorg@shipout}%
481         \else\ifx\mem@oldshipout\AtBegShi@Everyshi@shipout
482             \def\AtBegShi@found{\mem@oldshipout}%
483         \else
484             \expandafter\ifx\csname @EveryShipout@Org@Shipout\endcsname
485                 \relax
486             \ifx\@EveryShipout@Shipout\AtBegShi@Everyshi@shipout
487                 \def\AtBegShi@found{\@EveryShipout@Shipout}%

```

```

488         \fi
489     \fi
490     \fi\fi\fi\fi\fi\fi
491     \ifx\AtBegShi@found\relax
492     \else
493         \expandafter\endgroup
494         \expandafter\def\AtBegShi@found{%
495             \edef\AtBegShi@GroupLevel{\number\currentgrouplevel}%
496             \afterassignment\@EveryShipout@Test
497             \setbox\AtBeginShipoutBox=%
498             }%
499         \def\@EveryShipout@Test{%
500             \ifnum\AtBegShi@GroupLevel=\currentgrouplevel
501             \else
502                 \expandafter\aftergroup
503                 \fi
504             \AtBegShi@Everyshi@Output
505             }%
506         \def\AtBegShi@Everyshi@Output{%
507             \ifvoid\AtBeginShipoutBox
508             \else
509                 \global\setbox\@cclv\box\AtBeginShipoutBox
510                 \expandafter\@EveryShipout@Output
511                 \fi
512             }%
513         \@PackageInfoNoLine{atbegshi}{Package 'everyshi' patched}%
514         \begingroup
515         \fi
516     \fi
517 \endgroup
518 \let\AtBegShi@PatchEveryshi\relax
519 }
520 \ifpackageloaded{everyshi}{%
521     \AtBegShi@PatchEveryshi
522 }{%
523     \AtBeginDocument{\AtBegShi@PatchEveryshi}%
524 }

```

3.5.3 Class memoir

Fix of method and box.

```

525 \def\AtBegShi@PatchMemoir{%
526     \begingroup
527     \def\AtBegShi@Memoir@shipout{%
528         \afterassignment\mem@shipi
529         \setbox\@cclv=%
530     }%
531     \def\AtBegShi@Memoir@shipi{%
532         \ifvoid\@cclv
533         \expandafter\aftergroup
534         \fi
535         \mem@shipii
536     }%
537     \def\AtBegShi@Memoir@shipiiA{%
538         \mem@oldshipout\vbox{%
539             \trimmarks
540             \unvbox\@cclv
541         }%
542     }%
543     \def\AtBegShi@Memoir@shipiiB{%
544         \ifvoid\@cclv
545         \mem@oldshipout\box\@cclv

```

```

546 \else
547 \mem@oldshipout\vbox{%
548 \trimmarks
549 \unvbox\@cclv
550 }%
551 \fi
552 }%
553 \ifx\AtBegShi@Memoir@shipi\mem@shipi
554 \ifcase\ifx\AtBegShi@Memoir@shipiiA\mem@shipii
555 \z@
556 \else
557 \ifx\AtBegShi@Memoir@shipiiB\mem@shipii
558 \z@
559 \else
560 \@ne
561 \fi
562 \fi
563 \let\AtBegShi@found\relax
564 \ifx\shipout\AtBegShi@Memoir@shipout
565 \def\AtBegShi@found{\shipout}%
566 \else\ifx\AtBegShi@OrgShipout\AtBegShi@Memoir@shipout
567 \def\AtBegShi@found{\AtBegShi@OrgShipout}%
568 \else\ifx\CROP@shipout\AtBegShi@Memoir@shipout
569 \def\AtBegShi@found{\CROP@shipout}%
570 \else\ifx\GPTorg@shipout\AtBegShi@Memoir@shipout
571 \def\AtBegShi@found{\GPTorg@shipout}%
572 \else\ifx\THBorg@shipout\AtBegShi@Memoir@shipout
573 \def\AtBegShi@found{\THBorg@shipout}%
574 \else\ifx\@EveryShipout@Org@Shipout\AtBegShi@Memoir@shipout
575 \def\AtBegShi@found{\@EveryShipout@Org@Shipout}%
576 \fi\fi\fi\fi\fi\fi
577 \ifx\AtBegShi@found\relax
578 \else
579 \expandafter\endgroup
580 \expandafter\def\AtBegShi@found{%
581 \edef\AtBegShi@GroupLevel{\number\currentgrouplevel}%
582 \afterassignment\mem@shipi
583 \setbox\AtBeginShipoutBox=%
584 }%
585 \def\mem@shipi{%
586 \ifnum\AtBegShi@GroupLevel=\currentgrouplevel
587 \else
588 \expandafter\aftergroup
589 \fi
590 \mem@shipii
591 }%
592 \def\mem@shipii{%
593 \ifvoid\AtBeginShipoutBox
594 \else
595 \setbox\AtBeginShipoutBox=\vbox{%
596 \trimmarks
597 \ifvbox\AtBeginShipoutBox
598 \unvbox\AtBeginShipoutBox
599 \else
600 \box\AtBeginShipoutBox
601 \fi
602 }%
603 \expandafter\mem@oldshipout
604 \expandafter\box
605 \expandafter\AtBeginShipoutBox
606 \fi
607 }%

```

```

608         \@PackageInfoNoLine{atbegshi}{Class 'memoir' patched}%
609         \begingroup
610         \fi
611         \fi
612         \fi
613     \endgroup
614     \let\AtBegShi@PatchMemoir\relax
615 }
616 \@ifclassloaded{memoir}{%
617     \AtBegShi@PatchMemoir
618 }{%
619     \AtBeginDocument{\AtBegShi@PatchMemoir}%
620 }
621 \AtBegShi@End
622 \end{package}

```

4 Test

```

623 (*test1)
624 \input atbegshi.sty\relax
625 \def\msg#{\immediate\write16}
626 \msg{File: atbegshi-test1.tex 2007/06/06 v1.5 Test file for plain-TeX}
627 \def\testmsg#1#2{%
628     \msg{#1}
629     \msg{*** Test with box (#1), expected page output [#2]}}%
630 }
631
632 \newbox\voidbox
633 \def\void{\box\voidbox}
634 \begingroup
635     \setbox\voidbox=\void
636 \endgroup
637
638 \count0=0\relax
639 \AtBeginShipout{%
640     \global\advance\count0 by 1\relax
641     \msg{* Inside \string\AtBeginShipout: [\the\count0]}}%
642 }
643
644 \AtBeginShipoutFirst{%
645     \msg{* Inside \string\AtBeginShipoutFirst}}%
646     Hello World%
647 }
648
649 \testmsg{\string\null}{1}
650 \shipout\null
651
652 \AtBeginShipoutFirst{%
653     This is too late%
654 }
655
656 \testmsg{void}{}
657 \shipout\void
658
659 \testmsg{\string\copy255 (not void)}{2}
660 \setbox255\hbox{\vrule height 10bp width 10bp}
661 \shipout\copy255 %
662
663 \testmsg{\string\copy255 (again)}{3}
664 \shipout\copy255 %
665

```

```

666 \testmsg{\string\box255}{4}
667 \shipout\box255 %
668
669 \testmsg{\string\box255 (again)}{}
670 \shipout\box255 %
671
672 \testmsg{\string\hbox}{5}
673 \shipout\hbox{\vrule height 5bp width 20bp}
674
675 \testmsg{\string\vbox}{6}
676 \shipout\vbox{\hrule height 20bp width 5bp}
677
678 \testmsg{\string\null, voided by hook}{}
679 \def\VoidBox{%
680   \begingroup
681     \setbox\AtBeginShipoutBox=\box\AtBeginShipoutBox
682   \endgroup
683 }
684 \AtBeginShipout{\VoidBox}
685 \shipout\null
686 \def\VoidBox{}
687
688 \msg{*** \string\begingroup}
689 \begingroup
690   \testmsg{void}{}%
691   \shipout\void
692 \msg{*** \string\endgroup}
693 \endgroup
694
695 \msg{*** \string\begingroup}
696 \begingroup
697   \testmsg{void}{}%
698   \shipout\void
699   \testmsg{\string\null}{8}%
700   \shipout\null
701 \msg{*** \string\endgroup}
702 \endgroup
703
704 \testmsg{output routine}{9}
705 Hello World
706 \vfill
707 \eject
708
709 \testmsg{\string\null\space(discarded)}{}
710 \AtBeginShipout{%
711   \msg{* Inside \string\AtBeginShipout: DISCARD}%
712   \AtBeginShipoutDiscard
713 }
714 \shipout\null
715
716 \end
717 </test1>

718 <*test2>
719 \NeedsTeXFormat{LaTeX2e}
720 \ProvidesFile{atbegshi-test2.tex}[2007/06/06 v1.5 Test file for LaTeX]
721 \RequirePackage{color}
722 \pagecolor{yellow}
723 \documentclass[a5paper,showtrims]{memoir}
724 \usepackage{atbegshi}
725 \AtBeginShipout{%
726   \setbox\AtBeginShipoutBox=\vbox{%
727     \vbox to 0pt{%

```

```

728      \kern-1.5in %
729      \hbox to 0pt{%
730      \kern-1.5in %
731      \color{blue}%
732      \rule{1in}{1in}%
733      \hss
734      }%
735      \vss
736      }%
737      \hrule
738      \hbox{\vrule\box\AtBeginShipoutBox\vrule}%
739      \hrule
740      }%
741  }
742  \usepackage{eso-pic}
743  \makeatletter
744  \@EveryShipout@Init
745  \let\@EveryShipout@Init\relax
746  \makeatother
747  \AddToShipoutPicture{%
748      \hspace{.52\paperwidth}%
749      \colorbox{cyan}{%
750          \rule{0mm}{\paperheight}%
751          \hspace{.48\paperwidth}%
752      }%
753  }

```

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

```

754 \expandafter\let\csname ver@crop.sty\endcsname\relax
755 \usepackage[color=red,cross,a4,center]{crop}
756 \begin{document}
757 \shipout\null
758 \shipout\box\csname voidb@x\endcsname
759 \section{Hello World}
760 \end{document}
761 </test2>

```

5 Installation

5.1 Download

Package. This package is available on CTAN¹:

[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:macros/latex/contrib/oberdiek/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
```

¹<http://ftp.ctan.org/tex-archive/>

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-example.tex → doc/latex/oberdiek/atbegshi-example.tex
atbegshi-test1.tex  → doc/latex/oberdiek/atbegshi-test1.tex
atbegshi-test2.tex  → doc/latex/oberdiek/atbegshi-test2.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 (te \TeX , mi \TeX , ...) relies on file name databases, you must refresh these. For example, te \TeX users run `texhash` or `mktextlsr`.

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 pdfL^AT_EX:

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

6 History

[2007/04/17 v1.0]

- First version.

[2007/04/18 v1.1]

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

7 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; numbers in roman refer to the code lines where the entry is used.

Symbols	
<code>\@EveryShipout@Init</code>	744, 745
<code>\@EveryShipout@Org@Shipout</code>	400, 401, 574, 575
<code>\@EveryShipout@Output</code>	464, 466, 510
<code>\@EveryShipout@Shipout</code>	486, 487
<code>\@EveryShipout@Test</code>	459, 469, 496, 499
<code>\@PackageInfoNoLine</code>	186, 442, 513, 608
<code>\@PackageWarning</code>	177, 242, 266
<code>\@ccclv</code>	372, 375, 385, 460, 463, 509, 529, 532, 540, 544, 545, 549
<code>\@empty</code>	203, 204, 222
<code>\@ifclassloaded</code>	616
<code>\@ifdefinable</code>	140
<code>\@ifpackageloaded</code>	451, 520
<code>\@one</code>	123, 126, 560
<code>\@undefined</code>	125

A	
\AddToShipoutPicture	747
\advance	640
\afterassignment	162, 371, 414, 459, 496, 528, 582
\aftergroup	169, 376, 420, 464, 502, 533, 588
\AtBeginDocument ..	286, 454, 523, 619
\AtBeginShipout	2, 6, 229, 639, 641, 684, 710, 711, 725
\AtBeginShipoutBox	159, 164, 176, 189, 195, 214, 219, 265, 276, 343, 348, 355, 415, 429, 432, 434, 439, 497, 507, 509, 583, 593, 595, 597, 598, 600, 605, 681, 726, 738
\AtBeginShipoutDiscard	3, 29, 144, 270, 712
\AtBeginShipoutFirst	3, 223, 237, 244, 644, 645, 652
\AtBeginShipoutInit ..	3, 274, 284, 286
\AtBeginShipoutNext ...	2, 14, 28, 233
\AtBeginShipoutUpperLeft	3, 7, 15, 342
\AtBegShi@AbortIfUndefined	358, 364, 365, 366, 367
\AtBegShi@AddHook	231, 235, 258
\AtBegShi@AddTo	239, 247, 259
\AtBegShi@BeginPicture	313, 329, 349
\AtBegShi@CheckDefinable	119, 144, 229, 233, 237, 274, 277
\AtBegShi@Crop@@ship	387, 394
\AtBegShi@Crop@ship	374, 392
\AtBegShi@Crop@shiplist ...	380, 393
\AtBegShi@Crop@shipout	370, 396, 398, 400, 402, 404, 406
\AtBegShi@Discardedfalse ...	181, 187
\AtBegShi@Discardedtrue	147
\AtBegShi@End	361, 621
\AtBegShi@EndPicture ..	323, 334, 351
\AtBegShi@Everyshi@Output ..	504, 506
\AtBegShi@Everyshi@shipout .	458, 471, 473, 475, 477, 479, 481, 486
\AtBegShi@Everyshi@Test ...	462, 469
\AtBegShi@First	193, 201
\AtBegShi@FirstDisabled ...	223, 241
\AtBegShi@found ...	395, 397, 399, 401, 403, 405, 407, 409, 412, 470, 472, 474, 476, 478, 480, 482, 487, 491, 494, 563, 565, 567, 569, 571, 573, 575, 577, 580
\AtBegShi@GroupLevel	161, 168, 413, 418, 495, 500, 581, 586
\AtBegShi@Hook	182, 226, 231
\AtBegShi@HookFirst	204, 208, 228, 239
\AtBegShi@HookNext .	183, 184, 227, 235
\AtBegShi@horigin	295, 298, 345
\AtBegShi@Item	259, 261
\AtBegShi@Memoir@shipi	531, 553
\AtBegShi@Memoir@shipiiA ...	537, 554
\AtBegShi@Memoir@shipiiB ...	543, 557
\AtBegShi@Memoir@shipout	527, 564, 566, 568, 570, 572, 574
\AtBegShi@OrgProtect ..	179, 191, 194
\AtBegShi@OrgShipout ...	195, 277, 278, 398, 399, 473, 474, 566, 567
\AtBegShi@Output	171, 173
\AtBegShi@PatchCrop	368, 449, 452, 454
\AtBegShi@PatchEveryshi	456, 518, 521, 523
\AtBegShi@PatchMemoir	525, 614, 617, 619
\AtBegShi@Shipout	157, 279
\AtBegShi@Test	162, 166
\AtBegShi@vorigin	296, 299, 347
B	
\baselineskip	215, 383, 427
\begin	11, 756
\box	189, 195, 355, 385, 429, 438, 509, 545, 600, 604, 633, 666, 667, 669, 670, 681, 738, 758
C	
\catcode	38, 39, 40, 41, 42, 66, 67, 68, 69, 70, 71, 72, 73, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 151, 152, 154, 155, 199, 200
\circle	8
\color	16, 731
\colorbox	749
\copy .	218, 219, 339, 659, 661, 663, 664
\count	638, 640, 641
\CROP@@ship	378, 394, 422, 431
\CROP@kernel	384, 428
\CROP@ship	371, 392, 414, 417
\CROP@shiplist	389, 393, 424, 435
\CROP@shipout	388, 437, 475, 476, 568, 569
\csname	43, 53, 74, 87, 90, 114, 120, 150, 180, 248, 276, 283, 289, 303, 306, 317, 360, 484, 754, 758
\currentgrouplevel	161, 168, 413, 418, 495, 500, 581, 586
D	
\deadcycles	146
\documentclass	2, 723
\dp	213, 338
E	
\E	200
\eject	707
\empty	47
\end	34, 716, 760
\endcsname	43, 53, 74, 87, 90, 114, 120, 150, 180, 248, 276, 283, 289, 303, 306, 317, 360, 484, 754, 758
\endinput	62, 101
\endpicture	324
\errmessage	131
F	
\fill	25

G		P	
\g@addto@macro	256	\p@	160
\gdef	184, 226, 227, 228, 280	\PackageInfo	58
\GPTorg@shipout	402, 403, 477, 478, 570, 571	\pagecolor	722
H		\paperheight	8, 17, 18, 750
\hbox	159, 330, 343, 660, 672, 673, 729, 738	\paperwidth	8, 17, 18, 748, 751
\hrule	676, 737, 739	\par	24
\hspace	748, 751	\pdfhorigin	295
\hss	733	\pdfvorigin	296
\ht	212, 337, 348	\picture	315
I		\protect	179, 191, 194
\ifAtBegShi@Discarded	143, 185, 262	\ProvidesFile	720
\ifcase	44, 122, 302, 554	\ProvidesPackage	88
\ifdim	167	\put	8, 17, 18
\ifnum	168, 418, 500, 586	R	
\ifpdf	294	\RequirePackage	117, 292, 721
\ifvbox	597	\rlap	344
\ifvoid	176, 265, 375, 432, 463, 507, 532, 544, 593	\rule	732, 750
\ifx	45, 47, 53, 74, 82, 114, 120, 122, 125, 150, 204, 248, 283, 289, 303, 306, 317, 360, 392, 393, 394, 396, 398, 400, 402, 404, 406, 409, 469, 471, 473, 475, 477, 479, 481, 484, 486, 491, 553, 554, 557, 564, 566, 568, 570, 572, 574, 577	S	
\ignorespaces	321, 332	\section	12, 759
\immediate	55, 76, 625	\setbox	159, 164, 189, 206, 214, 330, 343, 372, 415, 434, 460, 497, 509, 529, 583, 595, 635, 660, 681, 726
\input	115, 290, 624	\shipout	278, 279, 396, 397, 471, 472, 564, 565, 650, 657, 661, 664, 667, 670, 673, 676, 685, 691, 698, 700, 714, 757, 758
K		\space	132, 709
\kern	160, 345, 347, 348, 728, 730	T	
L		\testmsg	627, 649, 656, 659, 663, 666, 669, 672, 675, 678, 690, 697, 699, 704, 709
\lastkern	167	\THBorg@shipout	404, 405, 479, 480, 572, 573
\line	17, 18	\the	91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 252, 641
\lineskip	216, 381, 425	\toks	251, 252
\lineskiplimit	217, 382, 426	\trimmarks	539, 548, 596
M		U	
\makeatletter	743	\unitlength	319
\makeatother	746	\unvbox	540, 549, 598
\mem@oldshipout	406, 407, 481, 482, 538, 545, 547, 603	\usepackage	3, 4, 5, 724, 742, 755
\mem@shipi	528, 553, 582, 585	V	
\mem@shipii	535, 554, 557, 590, 592	\vbox	206, 214, 346, 388, 434, 538, 547, 595, 675, 676, 726, 727
\MessageBreak	243, 267	\vfill	706
\msg	625, 626, 628, 629, 641, 645, 688, 692, 695, 701, 711	\void	633, 635, 657, 691, 698
N		\VoidBox	679, 684, 686
\NeedsTeXFormat	719	\voidbox	632, 633, 635
\newbox	632	\vrule	660, 673, 738
\newif	143	\vspace	25
\newpage	13, 22, 27, 32	\vss	352, 735
\null	649, 650, 678, 685, 699, 700, 709, 714, 757	W	
\number	161, 413, 495, 581	\wd	211
		\write	55, 76, 625
		X	
		\X	199

<code>\x</code>	43, 45, 47, 54, 58, 60, 75, 80, 87	213, 217, 218, 251, 252, 381, 382, 383, 425, 426, 427, 555, 558
Z			
<code>\z@</code>	...	128, 146, 167, 206, 211, 212,	<code>\z@skip</code> 215, 216