

# The `atbegshi` package

Heiko Oberdiek\*

2016/06/09 v1.18

## Abstract

This package is a modern reimplementation of package `everyshi` without the burden of compatibility. It makes use of  $\epsilon$ -TeX's if available. Both L<sup>A</sup>T<sub>E</sub>X and plain T<sub>E</sub>X are supported.

## Contents

<b>1</b>	<b>Documentation</b>	<b>2</b>
1.1	Examples	4
1.1.1	Example: circle in background	4
1.1.2	Example: adding TrimBox for dvipdfmx	5
<b>2</b>	<b>Method of <code>\shipout</code> overloading</b>	<b>6</b>
2.1	<code>\shipout</code>	6
2.2	<code>\afterassignment</code>	6
2.3	Test for direct or indirect boxes	7
2.3.1	With $\epsilon$ -TeX	7
2.3.2	Without $\epsilon$ -TeX	7
2.3.3	<code>\lastkern</code> method	8
2.4	Output	9
2.5	Separate box register	9
2.6	Summary	10
2.6.1	With $\epsilon$ -TeX	10
2.6.2	Without $\epsilon$ -TeX, traditional way	10
2.6.3	<code>\lastkern</code> method	11
<b>3</b>	<b>Implementation</b>	<b>11</b>
3.1	Reload check and package identification	11
3.2	Catcodes	13
3.3	Preparations	13
3.4	Additions to the shipout box	18
3.5	Positioning	20
3.6	Patches	21
3.6.1	Package <code>crop</code>	22
3.6.2	Package <code>everyshi</code>	23
3.6.3	Class <code>memoir</code>	24
<b>4</b>	<b>Test</b>	<b>27</b>
4.1	Catcode checks for loading	27

---

\*Please report any issues at <https://github.com/ho-tex/oberdiek/issues>

<b>5</b>	<b>Installation</b>	<b>31</b>
5.1	Download	31
5.2	Bundle installation	32
5.3	Package installation	32
5.4	Refresh file name databases	32
5.5	Some details for the interested	32
<b>6</b>	<b>History</b>	<b>33</b>
[2007/04/17 v1.0]		33
[2007/04/18 v1.1]		33
[2007/04/19 v1.2]		33
[2007/04/26 v1.3]		33
[2007/04/27 v1.4]		33
[2007/06/06 v1.5]		33
[2007/09/09 v1.6]		33
[2008/07/18 v1.7]		34
[2008/07/19 v1.8]		34
[2008/07/31 v1.9]		34
[2009/12/02 v1.10]		34
[2010/03/01 v1.11]		34
[2010/03/25 v1.12]		34
[2010/08/18 v1.13]		34
[2010/12/02 v1.14]		34
[2011/01/30 v1.15]		34
[2011/10/05 v1.16]		34
[2016/05/16 v1.17]		34
[2016/06/09 v1.18]		34

# 1 Documentation

Package `atbegshi` redefines `\shipout` to insert hooks for user code that is executed before the page is shipped out. The code may modify or even discard the output page. Three hooks are implemented:

1. A hook that is executed for every page, see `\AtBeginShipout`
2. A hook that is executed for the next page only, see `\AtBeginShipoutNext`
3. A hook that is only executed for the first page, see `\AtBeginShipoutFirst`

The hooks are executed in this order. The following three macros provide the user interface for adding code to these hooks:

```
\AtBeginShipout {<code>}
\AtBeginShipoutBox
```

Execute the `<code>` for every page. The page contents is held in box register `\AtBeginShipoutBox` and may be modified. Use `\AtBeginShipoutDiscard` if you want to discard the page.

*Note:* Package `everyshi` uses box register 255. With package `atbegshi` you must use `\AtBeginShipoutBox` instead.

If L<sup>A</sup>T<sub>E</sub>X calls `\shipout` in `\@outputpage` (part of its output routine), the meaning of `\protect` is `\noexpand`. L<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X'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  $\langle stuff \rangle$  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 { $\langle background material \rangle$ }`

This is a macro that puts material in the background of box `\AtBeginShipoutBox`. The  $\langle background material \rangle$  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 L<sup>A</sup>T<sub>E</sub>X users the  $\langle background material \rangle$  is set inside a `picture` environment:

```
\begin{picture}(0,0)
  \setlength{\unitlength}{1pt}%
   $\langle background material \rangle$ 
\end{picture}
```

`\AtBeginShipoutUpperLeftForeground { $\langle foreground material \rangle$ }`

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

`\AtBeginShipoutOriginalShipout  $\langle box \rangle$`

It stores the meaning of `\shipout` at the time this package is loaded.

`\AtBeginShipoutBoxWidth`  
`\AtBeginShipoutBoxHeight`  
`\AtBeginShipoutBoxDepth`

These macros store the dimensions of the output box `\AtBeginShipoutBox` before the original shipout is called. If `\shipout` is not redefined before the package loading or the box dimensions are not changed by the redefined `\shipout`, these macros contain the dimensions of the shipout box. These values can be remembered by `\label` and `\ref`. For example, this is done by the package module `zref-pagelayout` of project `zref`. The dimensions of the shipout page can be used in some T<sub>E</sub>X engines (pdfTeX in PDF mode, X<sub>Y</sub>TeX) to calculate the media size of the shipout page if `\pdfpagewidth` and `\pdfpageheight` are not set.

## 1.1 Examples

### 1.1.1 Example: circle in background

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

```
1  $\langle *example1 \rangle$ 
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{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 \begin{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 \end{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  $\text{\TeX}$  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{}` ( $= \text{\null}$ ) 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  $\text{\TeX}$  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  $\text{\TeX}$ , 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 $\varepsilon$ - $\TeX$

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

```
\shipout :=
\edef\saved@grouplevel{\number\currentgrouplevel}
\afterassignment\@test
\setbox\mybox=

\@test :=
\ifnum\saved@grouplevel=\currentgrouplevel
% case: indirect box, the assignment is completed
\@output
\else
% case: direct box, we are inside the box
\aftergroup\@outbox
\fi
```

### 2.3.2 Without $\varepsilon$ - $\TeX$

Life becomes complicate without  $\varepsilon$ - $\TeX$ . We cannot ask the group level. However, if we are inside a direct box, the box register `\mybox` is not yet changed by `\setbox`. Thus we need a special initial value and compare it in `\@test` with the current value of the box.

What can be used as initial value? Arbitrary box contents cannot be compared.  $\TeX$  only tells us a few properties:

- Box type: `\ifhbox`, `\ifvbox`
- Dimensions: `\wd`, `\ht`, `\dp`
- Voidness: `\ifvoid`

Unhappily all these qualities even combined are not sufficient for constructing an initial box value, because `\shipout` can be called with a box that is accidently just the same as the choosen initial value.

Nevertheless we have two alternatives for an initial value:

- A box of some type with some funny settings that are unlikely to occur in real life, e.g a height of `4911sp-\maxdimen`.
- A void box.

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

- `\shipout` is used in a group that perhaps closes some pages later. A bad place for `\@output`.
- Without a surrounding group `\aftergroup` effectively kills its argument.

In the first case of a box with special dimensions we can even loose the page. However in the case of the void box, this effect is even desired, because the original `\shipout` does not output void boxes. All we have to do is to ensure that our box `\mybox` is always void except for the phase when the overloaded `\shipout` is executed. And secondly we must keep this semantics of `\shipout` for the void case in our macros, namely `\@output`.

```
\shipout :=
% trick to get a void box \mybox
\begingroup
  \setbox\mybox=\box\mybox
\endgroup
\afterassignment\@test
\setbox\mybox=

\@test :=
\ifvoid\mybox
  \aftergroup\@output
\else
  \@output
\fi
```

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

### 2.3.3 `\lastkern` method

At the beginning of a new box, there is no `\kern`, the contents of the box is still empty and `\lastkern` returns 0 pt. This can be used to distinguish between direct and indirect boxes: We execute `\setbox` in a box with a preceding non-zero kern. After an indirect box, `\lastkern` sees this kern, otherwise it returns 0 pt.

```
\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  $\varepsilon$ -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  $\varepsilon$ -TeX the box must even be voided. However, the use case above requires that the box contents does not change at all. Therefore we must reserve a separate box register to avoid collisions with user box registers.

*Note:* Box register number 255 is special for the output routine, because TeX complains if this box is not voided by the output routine. However, this requirement does not apply to `\shipout` at all. In fact `\shipout` does not change any box register. This is usually done by a call of `\box`, but the output routine can do it later *after* invoking of `\shipout`.

## 2.6 Summary

### 2.6.1 With $\varepsilon$ -TeX

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

```
\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 $\varepsilon$ -TeX, traditional way

And for TeX without  $\varepsilon$ -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  $\text{\TeX}$  without  $\varepsilon\text{-TeX}$  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  $\varepsilon\text{-TeX}$ '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  $\text{\LaTeX}$ .

```

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 [2016/06/09 v1.18 At begin shipout hook (H0)]%

```

### 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{inwarerr}[2007/09/09]%
177 \TMP@RequirePackage{ltxcmds}[2010/03/01]%
178 \else
179 \RequirePackage{inwarerr}[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
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{Opt}

\AtBeginShipoutBoxDepth
273 \def\AtBeginShipoutBoxDepth{Opt}

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 }

```

BeginShipoutAddToBoxForeground

```

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{%
491     \ifx\pdfhorigin\undefined\pdfvariable horigin\else\pdfhorigin\fi}%
492   \def\AtBegShi@vorigin{%
493     \ifx\pdfvorigin\undefined\pdfvariable vorigin\else\pdfvorigin\fi}%
494 \else
495   \def\AtBegShi@horigin{72.27pt}%
496   \def\AtBegShi@vorigin{72.27pt}%
497 \fi

498 \begingroup
499 \ifcase
500   \expandafter\ifx\csname picture\endcsname\relax
501     1%
502   \else
503     \expandafter\ifx\csname endpicture\endcsname\relax
504       1%
505     \else
506       0%
507     \fi
508   \fi
509 \endgroup
510 \def\AtBegShi@BeginPicture{%
511   \begingroup
512   \picture(0,0)\relax
513   \begingroup\expandafter\expandafter\expandafter\endgroup
514   \expandafter\ifx\csname unitlength\endcsname\relax
515     \else
516       \unitlength=1pt\relax
517     \fi
518   \ignorespaces
519 }%
520 \def\AtBegShi@EndPicture{%

```

```

521 \endpicture
522 \endgroup
523 }%
524 \else
525 \endgroup
526 \def\AtBegShi@BeginPicture{%
527 \setbox\ltx@zero=\hbox\bgroup
528 \begingroup
529 \ignorespaces
530 }%
531 \def\AtBegShi@endPicture{%
532 \endgroup
533 \egroup
534 \ht\ltx@zero=0pt\relax
535 \dp\ltx@zero=0pt\relax
536 \copy\ltx@zero
537 }%
538 \fi

```

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

```

539 \def\AtBeginShipoutUpperLeft#1{%
540 \AtBeginShipoutAddToBox{%
541 \kern-\AtBegShi@horigin\relax
542 \vbox to 0pt{%
543 \kern-\AtBegShi@vorigin\relax
544 \AtBegShi@BeginPicture
545 #1%
546 \AtBegShi@endPicture
547 \vss
548 }%
549 }%
550 }

```

`\AtBeginShipoutUpperLeftForeground`

```

551 \def\AtBeginShipoutUpperLeftForeground#1{%
552 \AtBeginShipoutAddToBoxForeground{%
553 \kern-\AtBegShi@horigin\relax
554 \vbox to 0pt{%
555 \kern-\AtBegShi@vorigin\relax
556 \AtBegShi@BeginPicture
557 #1%
558 \AtBegShi@endPicture
559 \vss
560 }%
561 }%
562 }

```

### 3.6 Patches

Patches for  $\text{\LaTeX}$  packages that redefine `\shipout`.  $\text{\LaTeX}$  is now supposed to use  $\varepsilon\text{-TeX}$ . Thus we do not patch, without  $\text{\LaTeX}$  and  $\varepsilon\text{-TeX}$ .

```

563 \def\AtBegShi@AbortIfUndefined#1{%
564 \begingroup\expandafter\expandafter\expandafter\endgroup
565 \expandafter\ifx\csname#1\endcsname\relax
566 \expandafter\AtBegShi@AtEnd
567 \fi
568 }

```

```

569 \AtBegShi@AbortIfUndefined{currentgrouplevel}%
570 \AtBegShi@AbortIfUndefined{AtBeginDocument}%
571 \AtBegShi@AbortIfUndefined{@ifpackageloaded}%
572 \AtBegShi@AbortIfUndefined{@ifclassloaded}%

```

### 3.6.1 Package crop

Fix of method and box.

```

573 \def\AtBegShi@PatchCrop{%
574   \begingroup
575     \def\AtBegShi@Crop@shipout{%
576       \afterassignment\CROP@ship
577       \setbox\@cclv=%
578     }%
579     \def\AtBegShi@Crop@ship{%
580       \ifvoid\@cclv
581         \expandafter\aftergroup
582         \fi
583       \CROP@ship
584     }%
585     \def\AtBegShi@Crop@shiplist{%
586       \lineskip\z@
587       \lineskiplimit\z@
588       \baselineskip\z@
589       \CROP@kernel
590       \box\@cclv
591     }%
592     \def\AtBegShi@Crop@@ship{%
593       \CROP@shipout\vbox{%
594         \CROP@shiplist
595       }%
596     }%
597     \ifx\AtBegShi@Crop@ship\CROP@ship
598       \ifx\AtBegShi@Crop@shiplist\CROP@shiplist
599         \ifx\AtBegShi@Crop@@ship\CROP@@ship
600           \let\AtBegShi@found\relax
601           \ifx\shipout\AtBegShi@Crop@shipout
602             \def\AtBegShi@found{\shipout}%
603           \else\ifx\AtBeginShipoutOriginalShipout\AtBegShi@Crop@shipout
604             \def\AtBegShi@found{\AtBeginShipoutOriginalShipout}%
605           \else\ifx\@EveryShipout@Org@Shipout\AtBegShi@Crop@shipout
606             \def\AtBegShi@found{\@EveryShipout@Org@Shipout}%
607           \else\ifx\GPTorg@shipout\AtBegShi@Crop@shipout
608             \def\AtBegShi@found{\GPTorg@shipout}%
609           \else\ifx\THBorg@shipout\AtBegShi@Crop@shipout
610             \def\AtBegShi@found{\THBorg@shipout}%
611           \else\ifx\mem@oldshipout\AtBegShi@Crop@shipout
612             \def\AtBegShi@found{\mem@oldshipout}%
613           \fi\fi\fi\fi\fi
614           \ifx\AtBegShi@found\relax
615             \else
616               \expandafter\endgroup
617               \expandafter\def\AtBegShi@found{%
618                 \edef\AtBegShi@GroupLevel{\number\currentgrouplevel}%
619                 \afterassignment\CROP@ship
620                 \setbox\AtBeginShipoutBox=%
621               }%
622             \def\CROP@ship{%

```

```

623         \ifnum\AtBegShi@GroupLevel=\currentgrouplevel
624         \else
625             \expandafter\aftergroup
626         \fi
627         \CROP@@ship
628     }%
629     \def\CROP@shiplist{%
630         \lineskip Opt\relax
631         \lineskiplimit Opt\relax
632         \baselineskip Opt\relax
633         \CROP@kernel
634         \box\AtBeginShipoutBox
635     }%
636     \def\CROP@@ship{%
637         \ifvoid\AtBeginShipoutBox
638         \else
639             \setbox\AtBeginShipoutBox=\vbox{%
640                 \CROP@shiplist
641             }%
642             \AtBegShi@GetBoxSize\AtBeginShipoutBox
643             \expandafter\CROP@shipout
644             \expandafter\box
645             \expandafter\AtBeginShipoutBox
646         \fi
647     }%
648     \@PackageInfoNoLine{atbegshi}{Package `crop' patched}%
649     \begingroup
650     \fi
651     \fi
652     \fi
653     \fi
654     \endgroup
655     \let\AtBegShi@PatchCrop\relax
656 }
657 \@ifpackageloaded{crop}{%
658     \AtBegShi@PatchCrop
659 }{%
660     \AtBeginDocument{\AtBegShi@PatchCrop}%
661 }

```

### 3.6.2 Package everyshi

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

```

662 \def\AtBegShi@PatchEveryshi{%
663     \begingroup
664     \long\def\AtBegShi@Everyshi@shipout{%
665         \afterassignment\@EveryShipout@Test
666         \global\setbox\@cc1v= %
667     }%
668     \long\def\AtBegShi@Everyshi@Test{%
669         \ifvoid\@cc1v\relax
670             \aftergroup\@EveryShipout@Output
671         \else
672             \@EveryShipout@Output
673         \fi
674     }%
675     \ifx\AtBegShi@Everyshi@Test\@EveryShipout@Test
676         \let\AtBegShi@found\relax

```

```

677 \ifx\shipout\AtBegShi@Everyshi@shipout
678 \def\AtBegShi@found{\shipout}%
679 \else\ifx\AtBeginShipoutOriginalShipout\AtBegShi@Everyshi@shipout
680 \def\AtBegShi@found{\AtBeginShipoutOriginalShipout}%
681 \else\ifx\CROP@shipout\AtBegShi@Everyshi@shipout
682 \def\AtBegShi@found{\CROP@shipout}%
683 \else\ifx\GPTorg@shipout\AtBegShi@Everyshi@shipout
684 \def\AtBegShi@found{\GPTorg@shipout}%
685 \else\ifx\THBorg@shipout\AtBegShi@Everyshi@shipout
686 \def\AtBegShi@found{\THBorg@shipout}%
687 \else\ifx\mem@oldshipout\AtBegShi@Everyshi@shipout
688 \def\AtBegShi@found{\mem@oldshipout}%
689 \else
690 \expandafter\ifx\csname @EveryShipout@Org@Shipout\endcsname
691 \relax
692 \ifx\@EveryShipout@Shipout\AtBegShi@Everyshi@shipout
693 \def\AtBegShi@found{\@EveryShipout@Shipout}%
694 \fi
695 \fi
696 \fi\fi\fi\fi\fi\fi
697 \ifx\AtBegShi@found\relax
698 \else
699 \expandafter\endgroup
700 \expandafter\def\AtBegShi@found{%
701 \edef\AtBegShi@GroupLevel{\number\currentgrouplevel}%
702 \afterassignment\@EveryShipout@Test
703 \setbox\AtBeginShipoutBox=%
704 }%
705 \def\@EveryShipout@Test{%
706 \ifnum\AtBegShi@GroupLevel=\currentgrouplevel
707 \else
708 \expandafter\aftergroup
709 \fi
710 \AtBegShi@Everyshi@Output
711 }%
712 \def\AtBegShi@Everyshi@Output{%
713 \ifvoid\AtBeginShipoutBox
714 \else
715 \global\setbox\ltx@cclv\box\AtBeginShipoutBox
716 \expandafter\@EveryShipout@Output
717 \fi
718 }%
719 \@PackageInfoNoLine{atbegshi}{Package `everyshi' patched}%
720 \begingroup
721 \fi
722 \fi
723 \endgroup
724 \let\AtBegShi@PatchEveryshi\relax
725 }
726 \@ifpackageloaded{everyshi}{%
727 \AtBegShi@PatchEveryshi
728 }{%
729 \AtBeginDocument{\AtBegShi@PatchEveryshi}%
730 }

```

### 3.6.3 Class memoir

Fix of method and box.



```

731 \def\AtBegShi@PatchMemoir{%
732   \begingroup
733   \def\AtBegShi@Memoir@shipout{%
734     \afterassignment\mem@shipi
735     \setbox\@cclv=%
736   }%
737   \def\AtBegShi@Memoir@shipi{%
738     \ifvoid\@cclv
739       \expandafter\aftergroup
740       \fi
741     \mem@shipii
742   }%
743   \def\AtBegShi@Memoir@shipiiA{%
744     \mem@oldshipout\ vbox{%
745       \trimmarks
746       \unvbox\@cclv
747     }%
748   }%
749   \def\AtBegShi@Memoir@shipiiB{%
750     \ifvoid\@cclv
751       \mem@oldshipout\ box\@cclv
752     \else
753       \mem@oldshipout\ vbox{%
754         \trimmarks
755         \unvbox\@cclv
756       }%
757     \fi
758   }%
759   \def\AtBegShi@Memoir@PatchAB{%
760     \ifvoid\AtBeginShipoutBox
761     \else
762       \setbox\AtBeginShipoutBox=\ vbox{%
763         \trimmarks
764         \ifvbox\AtBeginShipoutBox
765           \unvbox\AtBeginShipoutBox
766         \else
767           \box\AtBeginShipoutBox
768         \fi
769       }%
770     \AtBegShi@GetBoxSize\AtBeginShipoutBox
771     \expandafter\mem@oldshipout
772     \expandafter\ box
773     \expandafter\AtBeginShipoutBox
774   \fi
775   }%
776   \def\AtBegShi@Memoir@shipiiC{% 2008/08/07 v1.6180339a
777     \ifvoid\@cclv
778       \mem@oldshipout\ box\@cclv
779     \else
780       \ifshowtrims
781         \mem@oldshipout\ vbox{\trimmarks\unvbox\@cclv}%
782       \else
783         \mem@oldshipout\ box\@cclv
784       \fi
785     \fi
786   }%
787   \def\AtBegShi@Memoir@shipiiD{% 2011/03/06 v3.6j
788     \ifvoid\@cclv

```

```

789     \mem@oldshipout\box\@cclv
790 \else
791     \ifshowtrims
792     \mem@oldshipout\ vbox{%
793     \trimmarks
794     \nointerlineskip
795     \box\@cclv
796     }%
797 \else
798     \mem@oldshipout\box\@cclv
799 \fi
800 \fi
801 }%
802 \def\AtBegShi@Memoir@PatchCD{%
803 \ifvoid\AtBeginShipoutBox
804 \else
805     \ifshowtrims
806     \setbox\AtBeginShipoutBox=\ vbox{%
807     \trimmarks
808     \nointerlineskip
809     \box\AtBeginShipoutBox
810     }%
811 \fi
812 \AtBegShi@GetBoxSize\AtBeginShipoutBox
813 \expandafter\mem@oldshipout
814 \expandafter\box
815 \expandafter\AtBeginShipoutBox
816 \fi
817 }%
818 \ifx\AtBegShi@Memoir@shipi\mem@shipi
819     \let\AtBegShi@found\ltx@one
820 \ifx\AtBegShi@Memoir@shipiiA\mem@shipii
821     \let\AtBegShi@found\ltx@zero
822     \global\let\AtBegShi@Memoir@PatchX\AtBegShi@Memoir@PatchAB
823 \else\ifx\AtBegShi@Memoir@shipiiB\mem@shipii
824     \let\AtBegShi@found\ltx@zero
825     \global\let\AtBegShi@Memoir@PatchX\AtBegShi@Memoir@PatchAB
826 \else\ifx\AtBegShi@Memoir@shipiiC\mem@shipii
827     \let\AtBegShi@found\ltx@zero
828     \global\let\AtBegShi@Memoir@PatchX\AtBegShi@Memoir@PatchCD
829 \else\ifx\AtBegShi@Memoir@shipiiD\mem@shipii
830     \let\AtBegShi@found\ltx@zero
831     \global\let\AtBegShi@Memoir@PatchX\AtBegShi@Memoir@PatchCD
832 \fi\fi\fi\fi
833 \ifcase\AtBegShi@found
834     \let\AtBegShi@found\relax
835 \ifx\shipout\AtBegShi@Memoir@shipout
836     \def\AtBegShi@found{\shipout}%
837 \else\ifx\AtBeginShipoutOriginalShipout\AtBegShi@Memoir@shipout
838     \def\AtBegShi@found{\AtBeginShipoutOriginalShipout}%
839 \else\ifx\CROP@shipout\AtBegShi@Memoir@shipout
840     \def\AtBegShi@found{\CROP@shipout}%
841 \else\ifx\GPTorg@shipout\AtBegShi@Memoir@shipout
842     \def\AtBegShi@found{\GPTorg@shipout}%
843 \else\ifx\THBorg@shipout\AtBegShi@Memoir@shipout
844     \def\AtBegShi@found{\THBorg@shipout}%
845 \else\ifx\@EveryShipout@Org@Shipout\AtBegShi@Memoir@shipout
846     \def\AtBegShi@found{\@EveryShipout@Org@Shipout}%

```

```

847 \fi\fi\fi\fi\fi\fi
848 \ifx\AtBegShi@found\relax
849 \else
850 \expandafter\endgroup
851 \expandafter\def\AtBegShi@found{%
852 \edef\AtBegShi@GroupLevel{\number\currentgrouplevel}%
853 \afterassignment\mem@shipi
854 \setbox\AtBeginShipoutBox=%
855 }%
856 \def\mem@shipi{%
857 \ifnum\AtBegShi@GroupLevel=\currentgrouplevel
858 \else
859 \expandafter\aftergroup
860 \fi
861 \mem@shipii
862 }%
863 \let\mem@shipii\AtBegShi@M memoir@PatchX
864 \@PackageInfoNoLine{atbegshi}{Class `memoir' patched}%
865 \begingroup
866 \fi
867 \fi
868 \fi
869 \endgroup
870 \let\AtBegShi@PatchMemoir\relax
871 }
872 \@ifclassloaded{memoir}{%
873 \AtBegShi@PatchMemoir
874 }{%
875 \AtBeginDocument{\AtBegShi@PatchMemoir}%
876 }
877 \AtBegShi@AtEnd%
878 \</package>

```

## 4 Test

### 4.1 Catcode checks for loading

```

879 \<test1>
880 \catcode`\{=1 %
881 \catcode`\}=2 %
882 \catcode`\#=6 %
883 \catcode`\@=11 %
884 \expandafter\ifx\csname count@\endcsname\relax
885 \countdef\count@=255 %
886 \fi
887 \expandafter\ifx\csname @gobble\endcsname\relax
888 \long\def\@gobble#1{}%
889 \fi
890 \expandafter\ifx\csname @firstofone\endcsname\relax
891 \long\def\@firstofone#1{#1}%
892 \fi
893 \expandafter\ifx\csname loop\endcsname\relax
894 \expandafter\@firstofone
895 \else
896 \expandafter\@gobble
897 \fi
898 {%

```

```

899 \def\loop#1\repeat{%
900   \def\body{#1}%
901   \iterate
902 }%
903 \def\iterate{%
904   \body
905   \let\next\iterate
906   \else
907   \let\next\relax
908   \fi
909   \next
910 }%
911 \let\repeat=\fi
912}%
913\def\RestoreCatcodes{}
914\count@=0 %
915\loop
916 \edef\RestoreCatcodes{%
917   \RestoreCatcodes
918   \catcode\the\count@=\the\catcode\count@\relax
919 }%
920\ifnum\count@<255 %
921 \advance\count@ 1 %
922\repeat
923
924\def\RangeCatcodeInvalid#1#2{%
925 \count@=#1\relax
926 \loop
927 \catcode\count@=15 %
928 \ifnum\count@<#2\relax
929 \advance\count@ 1 %
930 \repeat
931 }
932\def\RangeCatcodeCheck#1#2#3{%
933 \count@=#1\relax
934 \loop
935 \ifnum#3=\catcode\count@
936 \else
937 \errmessage{%
938   Character \the\count@\space
939   with wrong catcode \the\catcode\count@\space
940   instead of \number#3%
941 }%
942 \fi
943 \ifnum\count@<#2\relax
944 \advance\count@ 1 %
945 \repeat
946 }
947\def\space{ }
948\expandafter\ifx\csname LoadCommand\endcsname\relax
949 \def\LoadCommand{\input atbegshi.sty\relax}%
950\fi
951\def\Test{%
952 \RangeCatcodeInvalid{0}{47}%
953 \RangeCatcodeInvalid{58}{64}%
954 \RangeCatcodeInvalid{91}{96}%
955 \RangeCatcodeInvalid{123}{255}%
956 \catcode`\@=12 %

```

```

957 \catcode`\=0 %
958 \catcode`\%=14 %
959 \LoadCommand
960 \RangeCatcodeCheck{0}{36}{15}%
961 \RangeCatcodeCheck{37}{37}{14}%
962 \RangeCatcodeCheck{38}{47}{15}%
963 \RangeCatcodeCheck{48}{57}{12}%
964 \RangeCatcodeCheck{58}{63}{15}%
965 \RangeCatcodeCheck{64}{64}{12}%
966 \RangeCatcodeCheck{65}{90}{11}%
967 \RangeCatcodeCheck{91}{91}{15}%
968 \RangeCatcodeCheck{92}{92}{0}%
969 \RangeCatcodeCheck{93}{96}{15}%
970 \RangeCatcodeCheck{97}{122}{11}%
971 \RangeCatcodeCheck{123}{255}{15}%
972 \RestoreCatcodes
973 }
974 \Test
975 \csname @@end\endcsname
976 \end
977 /<test1>
978 (*test2)
979 \input atbegshi.sty\relax
980 \def\msg#{\immediate\write16}
981 \msg{File: atbegshi-test2.tex 2016/06/09 v1.18 Test file for plain-Tex}
982 \def\testmsg#1#2{%
983   \msg{%
984     \msg{*** Test with box (#1), expected page output [#2]}% hash-ok
985   }
986
987 \newbox\voidbox
988 \def\void{\box\voidbox}
989 \begingroup
990   \setbox\voidbox=\void
991 \endgroup
992
993 \count0=0\relax
994 \AtBeginShipout{%
995   \global\advance\count0 by 1\relax
996   \msg{* Inside \string\AtBeginShipout: [\the\count0]}%
997 }
998
999 \AtBeginShipoutFirst{%
1000   \msg{* Inside \string\AtBeginShipoutFirst}%
1001   Hello World%
1002 }
1003
1004 \testmsg{\string\null}{1}
1005 \shipout\null
1006
1007 \AtBeginShipoutFirst{%
1008   This is too late%
1009 }
1010
1011 \testmsg{void}{}
1012 \shipout\void
1013
1014 \testmsg{\string\copy255 (not void)}{2}

```

```

1015 \setbox255\hbox{\vrule height 10bp width 10bp}
1016 \shipout\copy255 %
1017
1018 \testmsg{\string\copy255 (again)}{3}
1019 \shipout\copy255 %
1020
1021 \testmsg{\string\box255}{4}
1022 \shipout\box255 %
1023
1024 \testmsg{\string\box255 (again)}{5}
1025 \shipout\box255 %
1026
1027 \testmsg{\string\hbox}{5}
1028 \shipout\hbox{\vrule height 5bp width 20bp}
1029
1030 \testmsg{\string\vbox}{6}
1031 \shipout\vbox{\hrule height 20bp width 5bp}
1032
1033 \testmsg{\string\null, voided by hook}{7}
1034 \def\VoidBox{%
1035   \begingroup
1036     \setbox\AtBeginShipoutBox=\box\AtBeginShipoutBox
1037   \endgroup
1038 }
1039 \AtBeginShipout{\VoidBox}
1040 \shipout\null
1041 \def\VoidBox{}
1042
1043 \msg{*** \string\begingroup}
1044 \begingroup
1045   \testmsg{void}{8}%
1046   \shipout(void
1047 \msg{*** \string\endgroup}
1048 \endgroup
1049
1050 \msg{*** \string\begingroup}
1051 \begingroup
1052   \testmsg{void}{9}%
1053   \shipout(void
1054   \testmsg{\string\null}{10}%
1055   \shipout\null
1056 \msg{*** \string\endgroup}
1057 \endgroup
1058
1059 \testmsg{output routine}{11}
1060 Hello World
1061 \vfill
1062 \eject
1063
1064 \testmsg{\string\null\space(discarded)}{12}
1065 \AtBeginShipout{%
1066   \msg{* Inside \string\AtBeginShipout: DISCARD}%
1067   \AtBeginShipoutDiscard
1068 }
1069 \shipout\null
1070
1071 \end
1072 \test2

```

```

1073 (*test3)
1074 \NeedsTeXFormat{LaTeX2e}
1075 \ProvidesFile{atbegshi-test3.tex}[2016/06/09 v1.18 Test file for LaTeX]
1076 \RequirePackage{color}
1077 \pagecolor{yellow}
1078 \documentclass[a5paper,showtrims]{memoir}
1079 \usepackage{atbegshi}
1080 \AtBeginShipout{%
1081   \setbox\AtBeginShipoutBox=\vbox{%
1082     \vbox to 0pt{%
1083       \kern-1.5in %
1084       \hbox to 0pt{%
1085         \kern-1.5in %
1086         \color{blue}%
1087         \rule{1in}{1in}%
1088         \hss
1089       }%
1090       \vss
1091     }%
1092     \hrule
1093     \hbox{\vrule\box\AtBeginShipoutBox\vrule}%
1094     \hrule
1095   }%
1096 }
1097 \usepackage{eso-pic}
1098 \makeatletter
1099 \@ifundefined{@EveryShipout@Init}{%
1100   \typeout{Test skipped}%
1101   \@@end
1102 }{}
1103 \@EveryShipout@Init
1104 \let\@EveryShipout@Init\relax
1105 \makeatother
1106 \AddToShipoutPicture{%
1107   \hspace{.52\paperwidth}%
1108   \colorbox{cyan}{%
1109     \rule{0mm}{\paperheight}%
1110     \hspace{.48\paperwidth}%
1111   }%
1112 }

```

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

```

1113 \expandafter\let\csname ver@crop.sty\endcsname\relax
1114 \usepackage[color=red,cross,a4,center]{crop}
1115 \begin{document}
1116 \shipout\null
1117 \shipout\box\csname voidb@x\endcsname
1118 \section{Hello World}
1119 \end{document}
1120 \end{test3}

```

## 5 Installation

### 5.1 Download

**Package.** This package is available on CTAN<sup>1</sup>:

---

<sup>1</sup>CTAN:pkg/atbegshi

[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 T<sub>E</sub>X Files” ([CTAN:pkg/tds](#)). 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
```

## 5.3 Package installation

**Unpacking.** The `.dtx` file is a self-extracting docstrip archive. The files are extracted by running the `.dtx` through plain T<sub>E</sub>X:

```
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):

<code>atbegshi.sty</code>	→ <code>tex/generic/oberdiek/atbegshi.sty</code>
<code>atbegshi.pdf</code>	→ <code>doc/latex/oberdiek/atbegshi.pdf</code>
<code>atbegshi-example1.tex</code>	→ <code>doc/latex/oberdiek/atbegshi-example1.tex</code>
<code>atbegshi-example2.tex</code>	→ <code>doc/latex/oberdiek/atbegshi-example2.tex</code>
<code>test/atbegshi-test1.tex</code>	→ <code>doc/latex/oberdiek/test/atbegshi-test1.tex</code>
<code>test/atbegshi-test2.tex</code>	→ <code>doc/latex/oberdiek/test/atbegshi-test2.tex</code>
<code>test/atbegshi-test3.tex</code>	→ <code>doc/latex/oberdiek/test/atbegshi-test3.tex</code>
<code>atbegshi.dtx</code>	→ <code>source/latex/oberdiek/atbegshi.dtx</code>

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 T<sub>E</sub>X distribution (T<sub>E</sub>X Live, mikT<sub>E</sub>X, ...) relies on file name databases, you must refresh these. For example, T<sub>E</sub>X Live users run `texhash` or `mktextlsr`.

## 5.5 Some details for the interested

**Unpacking with L<sup>A</sup>T<sub>E</sub>X.** The `.dtx` chooses its action depending on the format:

**plain T<sub>E</sub>X:** Run docstrip and extract the files.

**L<sup>A</sup>T<sub>E</sub>X:** Generate the documentation.

If you insist on using L<sup>A</sup>T<sub>E</sub>X for docstrip (really, docstrip does not need L<sup>A</sup>T<sub>E</sub>X), 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<sup>A</sup>T<sub>E</sub>X:

```
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  $\epsilon$ -T<sub>E</sub>X is missing.
- `\AtBeginShipoutDiscard` also resets `\deadcycles`.

[2007/04/19 v1.2]

- `\AtBeginShipoutEarly` removed for simplification reasons.
- Forgotten definition of `\AtBegShi@Info` added.
- Patches for packages `crop` and `everyshi` and class `memoir` added.

[2007/04/26 v1.3]

- Use of package `infwarerr`.
- Catcode section after generic header.

[2007/04/27 v1.4]

- Small optimizations.

[2007/06/06 v1.5]

- `\AtBeginShipoutUpperLeft` added.
- Example added.
- Fix in second test file for newer version of `memoir`.

[2007/09/09 v1.6]

- Catcode section rewritten.

[2008/07/18 v1.7]

- Documentation of `\AtBeginShipoutUpperLeft` fixed and extended.

[2008/07/19 v1.8]

- `\AtBeginShipoutUpperLeftForeground` added.

[2008/07/31 v1.9]

- Second example (TrimBox for dvipdfmx) added.
- No changes in package code.

[2009/12/02 v1.10]

- `\AtBeginShipoutOriginalShipout` added.
- Test file fixed.

[2010/03/01 v1.11]

- Compatibility with ini- $\TeX$  except for `\newbox`.

[2010/03/25 v1.12]

- `\AtBeginShipoutNext` can now be used inside `\AtBeginShipoutNext`.

[2010/08/18 v1.13]

- Fixes for `\AtBegShi@CheckDefinable`.

[2010/12/02 v1.14]

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

[2011/01/30 v1.15]

- Already loaded package files are not input in plain  $\TeX$ .

[2011/10/05 v1.16]

- `\AtBeginShipoutAddToBox`, `\AtBeginShipoutAddToBoxForeground` added.
- `\AtBeginShipoutBoxWidth`, `\AtBeginShipoutBoxHeight`, `\AtBeginShipoutBoxDepth` added.
- Updates for patches of class memoir.

[2016/05/16 v1.17]

- Documentation updates.

[2016/06/09 v1.18]

- Update for `\pdfhorign` in new Lua $\TeX$ .

## 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; plain numbers refer to the code lines where the entry is used.

Symbols	
\#	882
\%	958
\@	883, 956
\@end	1101
\@EveryShipout@Init	1103, 1104
\@EveryShipout@Org@Shipout	605, 606, 845, 846
\@EveryShipout@Output	670, 672, 716
\@EveryShipout@Shipout	692, 693
\@EveryShipout@Test	665, 675, 702, 705
\@PackageError	194, 338
\@PackageInfoNoLine	248, 648, 719, 864
\@PackageWarning	238, 300, 326
\@cclv	260, 577, 580, 590, 666, 669, 735, 738, 746, 750, 751, 755, 777, 778, 781, 783, 788, 789, 795, 798
\@ehc	341
\@ehd	196
\@firstofone	891, 894
\@gobble	888, 896
\@ifclassloaded	872
\@ifdefinable	201
\@ifpackageloaded	657, 726
\@ifundefined	1099
\@undefined	113, 188, 491, 493
\\	957
\{	880
\}	881
A	
\AddToShipoutPicture	1106
\advance	392, 453, 921, 929, 944, 995
\afterassignment	223, 576, 619, 665, 702, 734, 853
\aftergroup	84, 230, 581, 625, 670, 708, 739, 859
\AtBeginDocument	354, 660, 729, 875
\AtBeginShipout	2, 6, 44, 287, 994, 996, 1039, 1065, 1066, 1080
\AtBeginShipoutAddToBox	3, 279, 356, 540
\AtBeginShipoutAddToBoxForeground	415, 552
\AtBeginShipoutBox	45, 47, 220, 225, 237, 251, 257, 262, 325, 343, 357, 364, 374, 375, 379, 386, 387, 391, 392, 393, 407, 416, 423, 424, 425, 435, 436, 440, 447, 448, 452, 453, 454, 467, 620, 634, 637, 639, 642, 645, 703, 713, 715, 760, 762, 764, 765, 767, 770, 773, 803, 806, 809, 812, 815, 854, 1036, 1081, 1093
\AtBeginShipoutBoxDepth	269, 273
\AtBeginShipoutBoxHeight	268, 272
\AtBeginShipoutBoxWidth	4, 267, 271
\AtBeginShipoutDiscard	3, 29, 205, 330, 1067
\AtBeginShipoutFirst	3, 282, 295, 302, 999, 1000, 1007
\AtBeginShipoutInit	3, 335, 352, 354
\AtBeginShipoutNext	3, 14, 28, 291
\AtBeginShipoutOriginalShipout	4, 262, 344, 345, 603, 604, 679, 680, 837, 838
\AtBeginShipoutUpperLeft	4, 7, 15, 539
\AtBeginShipoutUpperLeftForeground	4, 551
\AtBegShi@AbortIfUndefined	563, 569, 570, 571, 572
\AtBegShi@AddHook	289, 293, 316
\AtBegShi@AddTo	297, 305, 317
\AtBegShi@AtEnd	150, 151, 167, 566, 877
\AtBegShi@BeginPicture	510, 526, 544, 556
\AtBegShi@CheckDefinable	182, 205, 287, 291, 295, 335, 344
\AtBegShi@Crop@ship	592, 599
\AtBegShi@Crop@ship	579, 597
\AtBegShi@Crop@shiplist	585, 598
\AtBegShi@Crop@shipout	575, 601, 603, 605, 607, 609, 611
\AtBegShi@Discardedfalse	242, 249
\AtBegShi@Discardedtrue	208
\AtBegShi@EndPicture	520, 531, 546, 558
\AtBegShi@Everyshi@Output	710, 712
\AtBegShi@Everyshi@shipout	664, 677, 679, 681, 683, 685, 687, 692
\AtBegShi@Everyshi@Test	668, 675
\AtBegShi@First	255, 276
\AtBegShi@FirstDisabled	282, 299
\AtBegShi@found	600, 602, 604, 606, 608, 610, 612, 614, 617, 676, 678, 680, 682, 684, 686, 688, 693, 697, 700, 819, 821, 824, 827, 830, 833, 834, 836, 838, 840, 842, 844, 846, 848, 851
\AtBegShi@GetBoxSize	257, 260, 266, 642, 770, 812
\AtBegShi@GroupLevel	222, 229, 618, 623, 701, 706, 852, 857
\AtBegShi@Hook	243, 284, 289

<code>\AtBegShi@HookFirst</code>	277, 279, 286, 297	<code>\count@</code>	885, 914, 918, 920, 921, 925, 927, 928, 929, 933, 935, 938, 939, 943, 944
<code>\AtBegShi@HookNext</code>	244, 246, 285, 293	<code>\countdef</code>	885
<code>\AtBegShi@horigin</code>	490, 495, 541, 553	<code>\CROP@ship</code>	583, 599, 627, 636
<code>\AtBegShi@Item</code>	317, 319	<code>\CROP@kernel</code>	589, 633
<code>\AtBegShi@Memoir@PatchAB</code>	759, 822, 825	<code>\CROP@ship</code>	576, 597, 619, 622
<code>\AtBegShi@Memoir@PatchCD</code>	802, 828, 831	<code>\CROP@shiplist</code>	594, 598, 629, 640
<code>\AtBegShi@Memoir@PatchX</code>	822, 825, 828, 831, 863	<code>\CROP@shipout</code>	593, 643, 681, 682, 839, 840
<code>\AtBegShi@Memoir@shipi</code>	737, 818	<code>\csname</code>	69, 76, 105, 121, 131, 169, 172, 183, 211, 241, 306, 340, 343, 351, 478, 481, 500, 503, 514, 565, 690, 884, 887, 890, 893, 948, 975, 1113, 1117
<code>\AtBegShi@Memoir@shipiiA</code>	743, 820	<code>\currentgrouplevel</code>	222, 229, 618, 623, 701, 706, 852, 857
<code>\AtBegShi@Memoir@shipiiB</code>	749, 823	<b>D</b>	
<code>\AtBegShi@Memoir@shipiiC</code>	776, 826	<code>\deadcycles</code>	207
<code>\AtBegShi@Memoir@shipiiD</code>	787, 829	<code>\dimen</code>	383, 391, 392, 393, 444, 452, 453, 454, 468, 470
<code>\AtBegShi@Memoir@shipout</code>	733, 835, 837, 839, 841, 843, 845	<code>\documentclass</code>	2, 37, 1078
<code>\AtBegShi@OrgProtect</code>	240, 253, 256	<code>\dp</code>	269, 373, 387, 392, 402, 434, 448, 453, 463, 535
<code>\AtBegShi@Output</code>	232, 234	<b>E</b>	
<code>\AtBegShi@PatchCrop</code>	573, 655, 658, 660	<code>\E</code>	275
<code>\AtBegShi@PatchEveryshi</code>	662, 724, 727, 729	<code>\eject</code>	1062
<code>\AtBegShi@PatchMemoir</code>	731, 870, 873, 875	<code>\empty</code>	72, 73
<code>\AtBegShi@restore</code>	358, 367, 377, 380, 396, 410, 417, 428, 438, 441, 457, 472	<code>\end</code>	34, 54, 976, 1071, 1119
<code>\AtBegShi@restorebox</code>	385, 411, 446, 473	<code>\endcsname</code>	69, 76, 105, 121, 131, 169, 172, 183, 211, 241, 306, 340, 343, 351, 478, 481, 500, 503, 514, 565, 690, 884, 887, 890, 893, 948, 975, 1113, 1117
<code>\AtBegShi@Shipout</code>	218, 346	<code>\endinput</code>	84, 167
<code>\AtBegShi@Test</code>	223, 227	<code>\endlinechar</code>	59, 90, 126, 132, 144
<code>\AtBegShi@vorigin</code>	492, 496, 543, 555	<code>\endpicture</code>	521
<code>\AtNextShipout</code>	260	<code>\errmessage</code>	937
<b>B</b>		<b>F</b>	
<code>\baselineskip</code>	403, 464, 588, 632	<code>\fill</code>	25
<code>\begin</code>	11, 50, 1115	<b>G</b>	
<code>\body</code>	900, 904	<code>\g@addto@macro</code>	314
<code>\box</code>	47, 251, 262, 590, 634, 644, 715, 751, 767, 772, 778, 783, 789, 795, 798, 809, 814, 988, 1021, 1022, 1024, 1025, 1036, 1093, 1117	<code>\gdef</code>	244, 284, 285, 286, 348
<b>C</b>		<code>\GPTorg@shipout</code>	607, 608, 683, 684, 841, 842
<code>\catcode</code>	57, 58, 60, 61, 62, 63, 64, 65, 66, 67, 68, 88, 89, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 124, 125, 127, 128, 129, 133, 134, 135, 136, 137, 138, 139, 142, 143, 145, 146, 147, 148, 152, 154, 212, 213, 215, 216, 274, 275, 880, 881, 882, 883, 918, 927, 935, 939, 956, 957, 958	<b>H</b>	
<code>\circle</code>	8	<code>\hbadness</code>	360, 363, 419, 422
<code>\color</code>	16, 1086	<code>\hbox</code>	45, 220, 364, 365, 394, 423, 426, 455, 527, 1015, 1027, 1028, 1084, 1093
<code>\colorbox</code>	1108	<code>\hfuzz</code>	359, 362, 418, 421
<code>\copy</code>	374, 406, 435, 469, 536, 1014, 1016, 1018, 1019	<code>\hrule</code>	1031, 1092, 1094
<code>\count</code>	993, 995, 996	<code>\hspace</code>	1107, 1110
		<code>\hss</code>	1088

<code>\ht</code> .....	268, 372, 374, 386, 391, 401, 433, 435, 447, 452, 462, 534	<code>\mem@oldshipout</code> .....	611, 612, 687, 688, 744, 751, 753, 771, 778, 781, 783, 789, 792, 798, 813
<b>I</b>		<code>\mem@shipi</code> .....	734, 818, 853, 856
<code>\ifAtBegShi@Discarded</code> .....	204, 247, 320, 323	<code>\mem@shipii</code> .....	741, 820, 823, 826, 829, 861, 863
<code>\ifcase</code> .....	185, 499, 833	<code>\MessageBreak</code> .....	301, 327, 339
<code>\ifdim</code> .....	228	<code>\msg</code> .....	980, 981, 983, 984, 996, 1000, 1043, 1047, 1050, 1056, 1066
<code>\ifhbox</code> .....	357, 416	<b>N</b>	
<code>\ifnum</code> .....	229, 623, 706, 857, 920, 928, 935, 943	<code>\NeedsTeXFormat</code> .....	1074
<code>\ifpdf</code> .....	489	<code>\newbox</code> .....	987
<code>\ifshowtrims</code> .....	780, 791, 805	<code>\newpage</code> .....	13, 22, 27, 32, 52
<code>\ifvbox</code> .....	379, 440, 764	<code>\next</code> .....	905, 907, 909
<code>\ifvoid</code> ....	237, 325, 580, 637, 669, 713, 738, 750, 760, 777, 788, 803	<code>\nointerlineskip</code> .....	794, 808
<code>\ifx</code> .....	70, 73, 76, 105, 113, 116, 169, 172, 183, 185, 188, 211, 277, 306, 351, 478, 481, 491, 493, 500, 503, 514, 565, 597, 598, 599, 601, 603, 605, 607, 609, 611, 614, 675, 677, 679, 681, 683, 685, 687, 690, 692, 697, 818, 820, 823, 826, 829, 835, 837, 839, 841, 843, 845, 848, 884, 887, 890, 893, 948	<code>\null</code> .....	1004, 1005, 1033, 1040, 1054, 1055, 1064, 1069, 1116
<code>\ignorespaces</code> .....	518, 529	<code>\number</code> .....	222, 618, 701, 852, 940
<code>\immediate</code> .....	78, 107, 980	<b>P</b>	
<code>\input</code> .....	173, 482, 949, 979	<code>\p@</code> .....	221
<code>\iterate</code> .....	901, 903, 905	<code>\PackageInfo</code> .....	81
<b>K</b>		<code>\pagecolor</code> .....	1077
<code>\kern</code> .....	221, 408, 425, 436, 468, 470, 541, 543, 553, 555, 1083, 1085	<code>\paperheight</code> .....	8, 17, 18, 1109
<b>L</b>		<code>\paperwidth</code> .....	8, 17, 18, 1107, 1110
<code>\lastkern</code> .....	228	<code>\par</code> .....	24
<code>\line</code> .....	17, 18	<code>\pdfhorigin</code> .....	491
<code>\lineskip</code> .....	404, 465, 586, 630	<code>\pdfvariable</code> .....	491, 493
<code>\lineskiplimit</code> ....	405, 466, 587, 631	<code>\pdfvorigin</code> .....	493
<code>\LoadCommand</code> .....	949, 959	<code>\picture</code> .....	512
<code>\loop</code> .....	899, 915, 926, 934	<code>\protect</code> .....	240, 253, 256
<code>\ltx@cclv</code> .....	715	<code>\ProvidesFile</code> .....	1075
<code>\ltx@empty</code> .....	277, 281	<code>\ProvidesPackage</code> .....	74, 122
<code>\ltx@ifUndefined</code> .....	337	<code>\put</code> .....	8, 17, 18
<code>\ltx@ifundefined</code> .....	258	<b>R</b>	
<code>\ltx@newif</code> .....	204	<code>\raise</code> .....	374, 435
<code>\ltx@one</code> .....	186, 189, 819	<code>\RangeCatcodeCheck</code> .....	932, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971
<code>\ltx@zero</code> ..	191, 207, 309, 310, 365, 371, 372, 373, 374, 383, 391, 392, 393, 394, 400, 401, 402, 406, 426, 432, 433, 434, 435, 444, 452, 453, 454, 455, 461, 462, 463, 468, 469, 470, 527, 534, 535, 536, 821, 824, 827, 830	<code>\RangeCatcodeInvalid</code> .....	924, 952, 953, 954, 955
<b>M</b>		<code>\repeat</code> .....	899, 911, 922, 930, 945
<code>\makeatletter</code> .....	1098	<code>\RequirePackage</code> ...	179, 180, 487, 1076
<code>\makeatother</code> .....	1105	<code>\RestoreCatcodes</code> ..	913, 916, 917, 972
		<code>\rule</code> .....	1087, 1109
		<b>S</b>	
		<code>\section</code> .....	12, 1118
		<code>\setbox</code> .....	45, 220, 225, 251, 364, 365, 393, 394, 423, 426, 454, 455, 527, 577, 620, 639, 666, 703, 715, 735, 762, 806, 854, 990, 1015, 1036, 1081
		<code>\shipout</code> .....	345, 346, 601, 602, 677, 678, 835, 836, 1005, 1012, 1016, 1019,

1022, 1025, 1028, 1031, 1040, 1046, 1053, 1055, 1069, 1116, 1117	\usepackage . . . 3, 4, 5, 38, 39, 1079, 1097, 1114
\space . . . 195, 339, 938, 939, 947, 1064	
\special . . . . . 46	
<b>T</b>	<b>V</b>
\Test . . . . . 951, 974	\vbadness . . . . . 382, 390, 443, 451
\testmsg . . . . 982, 1004, 1011, 1014, 1018, 1021, 1024, 1027, 1030, 1033, 1045, 1052, 1054, 1059, 1064	\vbox . . . . . 393, 454, 542, 554, 593, 639, 744, 753, 762, 781, 792, 806, 1030, 1031, 1081, 1082
\THBorg@shipout . . . . .	\vfill . . . . . 1061
. . . . 609, 610, 685, 686, 843, 844	\vfuzz . . . . . 381, 389, 442, 450
\the . . 132, 133, 134, 135, 136, 137, 138, 139, 152, 267, 268, 269, 310, 359, 360, 381, 382, 383, 386, 387, 418, 419, 442, 443, 444, 447, 448, 918, 938, 939, 996	\void . . . . . 988, 990, 1012, 1046, 1053
\TMP@EnsureCode . . . . .	\VoidBox . . . . . 1034, 1039, 1041
. . . . 149, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166	\voidbox . . . . . 987, 988, 990
\TMP@RequirePackage . . . . .	\vrule . . . . . 1015, 1028, 1093
. . . . . 170, 176, 177, 479, 485	\vspace . . . . . 25
\toks . . . . . 309, 310	\vss . . . . . 547, 559, 1090
\trimmarks 745, 754, 763, 781, 793, 807	
\typeout . . . . . 1100	<b>W</b>
<b>U</b>	\wd . . . . . 267, 364, 371, 400, 423, 425, 432, 436, 461
\unhcopy . . . . . 375, 424	\write . . . . . 78, 107, 980
\unitlength . . . . . 516	
\unvbox . . . 407, 467, 746, 755, 765, 781	<b>X</b>
	\X . . . . . 274
	\x . . . . . 69, 70, 73, 77, 81, 83, 106, 111, 121, 130, 142
	<b>Z</b>
	\z@ . . . . . 586, 587, 588