

The atbegshi package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2008/07/31 v1.9

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-TeX are supported.

Contents

1	Documentation	2
1.1	Examples	3
1.1.1	Example: circle in background	3
1.1.2	Example: adding TrimBox for dvipdfmx	4
2	Method of <code>\shipout</code> overloading	5
2.1	<code>\shipout</code>	5
2.2	<code>\afterassignment</code>	5
2.3	Test for direct or indirect boxes	6
2.3.1	With ε -TeX	6
2.3.2	Without ε -TeX	6
2.3.3	<code>\lastkern</code> method	7
2.4	Output	8
2.5	Separate box register	8
2.6	Summary	8
2.6.1	With ε -TeX	8
2.6.2	Without ε -TeX, traditional way	9
2.6.3	<code>\lastkern</code> method	9
3	Implementation	10
3.1	Reload check and package identification	10
3.2	Catcodes	11
3.3	Preparations	12
3.4	Positioning	15
3.5	Patches	17
3.5.1	Package <code>crop</code>	17
3.5.2	Package <code>everyshi</code>	18
3.5.3	Class <code>memoir</code>	19
4	Test	21
4.1	Catcode checks for loading	21
5	Installation	25
5.1	Download	25
5.2	Bundle installation	25
5.3	Package installation	25
5.4	Refresh file name databases	26
5.5	Some details for the interested	26

6 History	26
[2007/04/17 v1.0]	26
[2007/04/18 v1.1]	27
[2007/04/19 v1.2]	27
[2007/04/26 v1.3]	27
[2007/04/27 v1.4]	27
[2007/06/06 v1.5]	27
[2007/09/09 v1.6]	27
[2008/07/18 v1.7]	27
[2008/07/19 v1.8]	27
[2008/07/31 v1.9]	27
7 Index	27

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.

The macro `\AtBeginShipoutUpperLeft` is intended to be used in one of the hook setting macros, such as `\AtBeginShipout`, `\AtBeginShipoutFirst`, or `\AtBeginShipoutNext`.

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}
```

`\AtBeginShipoutUpperLeftForeground {<foreground material>}`

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

1.1 Examples

1.1.1 Example: circle in background

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

```
1 (*example1)
2 \documentclass[a4paper]{article}
3 \usepackage{color}
4 \usepackage{atbegshi}
```

Package `picture` makes life a little easier, because we can now also use length specifications in `picture`'s commands.

```
5 \usepackage{picture}
```

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

```

6 \AtBeginShipout{%
7   \AtBeginShipoutUpperLeft{%
8     \put(0.5\paperwidth,-0.5\paperheight){\circle{10}}%
9   }%
10 }
11 \begin{document}
12 \section{Hello World}
13 \newpage
14 \AtBeginShipoutNext{%
15   \AtBeginShipoutUpperLeft{%
16     \color{red}%
17     \put(0,-0.5\paperheight){\line(1,0){\paperwidth}}%
18     \put(0.5\paperwidth, 0){\line(0,-1){\paperheight}}%
19   }%
20 }
21 Only on this page we add a red cross.
22 \newpage
23 This page has the circle only.
24 \par
25 \vspace{\fill}
26 The next page will be discarded.
27 \newpage
28 \AtBeginShipoutNext{%
29   \AtBeginShipoutDiscard
30 }
31 This page is discarded.
32 \newpage
33 The last page.
34 \end{document}
35 \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 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 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  $\Rightarrow$  \setbox\mybox=\box0 \@test
```

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

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

2.3 Test for direct or indirect boxes

Now we want to execute `\@test`, but where are we? We can be after the completed box assignment, if `\shipout` was called with an indirect box. Or we are right at the beginning of a direct box.

2.3.1 With 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 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 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 -TeX, traditional way

And for TeX without ε -TeX:

```

\newbox\mybox
\begingroup
    \setbox\mybox=\box\mybox % ensure \mybox is void
\endgroup
\let\original@shipout\shipout

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

\@test :=
    \ifvoid\mybox
        \expandafter\aftergroup
    \fi
    \@output

\@output :=
    \ifvoid\mybox
        % cancel output of void box
    \else
        \@hook
        \ifvoid\mybox
            % user code in \@hook could have voided the box
        \else
            \original@shipout\box\mybox
        \fi
    \fi

```

2.6.3 \lastkern method

And for TeX without ε -TeX using the \lastkern method:

```

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

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

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

\@output :=

```

```

\egroup
\endgroup
\ifvoid\mybox
% cancel output of void box
\else
\@hook
\ifvoid\mybox
% user code in \@hook could have voided the box
\else
\original@shipout\box\mybox
\fi
\fi

```

3 Implementation

Package atbegshi uses ε -TeX's `\currentgrouplevel`, if it is available. Otherwise the `\lastkern` method is used.

```
56 <*package>
```

3.1 Reload check and package identification

Reload check, especially if the package is not used with L^AT_EX.

```

57 \begingroup
58 \catcode44 12 % ,
59 \catcode45 12 % -
60 \catcode46 12 % .
61 \catcode58 12 % :
62 \catcode64 11 % @
63 \expandafter\let\expandafter\x\csname ver@atbegshi.sty\endcsname
64 \ifcase 0%
65 \ifx\x\relax % plain
66 \else
67 \ifx\x\empty % LaTeX
68 \else
69 1%
70 \fi
71 \fi
72 \else
73 \catcode35 6 % #
74 \catcode123 1 % {
75 \catcode125 2 % }
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 \endgroup
85 \expandafter\endinput
86 \fi
87 \endgroup

```

Package identification:

```

88 \begingroup
89 \catcode35 6 % #
90 \catcode40 12 % (
91 \catcode41 12 % )
92 \catcode44 12 % ,
93 \catcode45 12 % -

```

```

94 \catcode46 12 % .
95 \catcode47 12 % /
96 \catcode58 12 % :
97 \catcode64 11 % @
98 \catcode123 1 % {
99 \catcode125 2 % }
100 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
101 \def\x#1#2#3[#4]{\endgroup
102 \immediate\write-1{Package: #3 #4}%
103 \xdef#1{#4}%
104 }%
105 \else
106 \def\x#1#2[#3]{\endgroup
107 #2[#3]}%
108 \ifx#1\undefined
109 \xdef#1{#3}%
110 \fi
111 \ifx#1\relax
112 \xdef#1{#3}%
113 \fi
114 }%
115 \fi
116 \expandafter\x\csname ver@atbegshi.sty\endcsname
117 \ProvidesPackage{atbegshi}%
118 [2008/07/31 v1.9 At begin shipout hook (H0)]

```

3.2 Catcodes

```

119 \begingroup
120 \catcode123 1 % {
121 \catcode125 2 % }
122 \def\x{\endgroup
123 \expandafter\edef\csname AtBegShi@AtEnd\endcsname{%
124 \catcode35 \the\catcode35\relax
125 \catcode64 \the\catcode64\relax
126 \catcode123 \the\catcode123\relax
127 \catcode125 \the\catcode125\relax
128 }%
129 }%
130 \x
131 \catcode35 6 % #
132 \catcode64 11 % @
133 \catcode123 1 % {
134 \catcode125 2 % }
135 \def\TMP@EnsureCode#1#2{%
136 \edef\AtBegShi@AtEnd{%
137 \AtBegShi@AtEnd
138 \catcode#1 \the\catcode#1\relax
139 }%
140 \catcode#1 #2\relax
141 }
142 \TMP@EnsureCode{40}{12}% (
143 \TMP@EnsureCode{41}{12}% )
144 \TMP@EnsureCode{44}{12}% ,
145 \TMP@EnsureCode{45}{12}% -
146 \TMP@EnsureCode{47}{12}% /
147 \TMP@EnsureCode{46}{12}% .
148 \TMP@EnsureCode{58}{12}% :
149 \TMP@EnsureCode{61}{12}% =
150 \TMP@EnsureCode{94}{7}% ^ (superscript)
151 \TMP@EnsureCode{96}{12}% '

```

3.3 Preparations

```

152 \begingroup\expandafter\expandafter\expandafter\endgroup
153 \expandafter\ifx\csname RequirePackage\endcsname\relax
154   \input infwarerr.sty\relax
155 \else
156   \RequirePackage{infwarerr}[2007/09/09]%
157 \fi

\AtBegShi@CheckDefinable

158 \begingroup\expandafter\expandafter\expandafter\endgroup
159 \expandafter\ifx\csname @ifdefinable\endcsname\relax
160   \def\AtBegShi@CheckDefinable#1{%
161     \ifcase\ifx#1\relax
162       \@ne
163     \else
164       \ifx#1\@undefined
165         \@ne
166       \else
167         \z@
168       \fi
169     \fi
170     \errmessage{%
171       Package atbegshi: \string#1\space
172       is already defined%
173     }%
174   \endgroup
175 \fi
176 }%
177 \else
178   \def\AtBegShi@CheckDefinable#1{%
179     \@ifdefinable{#1}{}%
180   }%
181 \fi

182 \newif\ifAtBegShi@Discarded

\AtBeginShipoutDiscard

183 \AtBegShi@CheckDefinable\AtBeginShipoutDiscard
184 \def\AtBeginShipoutDiscard{%
185   \deadcycles=\z@
186   \global\AtBegShi@Discardedtrue
187 }

188 \begingroup\expandafter\expandafter\expandafter\endgroup
189 \expandafter\ifx\csname currentgrouplevel\endcsname\relax
190   \catcode'X=9 % ignore
191   \catcode'E=14 % comment
192 \else
193   \catcode'X=14 % comment
194   \catcode'E=9 % ignore
195 \fi

\AtBegShi@Shipout

196 \def\AtBegShi@Shipout{%
197 X \begingroup
198 X \setbox\AtBeginShipoutBox=\hbox\bgroup
199 X \kern\p@
200 E \edef\AtBegShi@GroupLevel{\number\currentgrouplevel}%
201 \afterassignment\AtBegShi@Test
202 X \global
203 \setbox\AtBeginShipoutBox=%
204 }
```

\AtBegShi@Test

```
205 \def\AtBegShi@Test{%
206 X \ifdim\lastkern=\z@
207 E \ifnum\AtBegShi@GroupLevel<\currentgrouplevel
208 \expandafter\aftergroup
209 \fi
210 \AtBegShi@Output
211 }
```

\AtBegShi@Output

```
212 \def\AtBegShi@Output{%
213 X \egroup
214 X \endgroup
215 \ifvoid\AtBeginShipoutBox
216 \@PackageWarning{atbegshi}{Ignoring void shipout box}%
217 \else
218 \let\AtBegShi@OrgProtect\protect
219 \csname set@typeset@protect\endcsname
220 \global\AtBegShi@Discardedfalse
221 \AtBegShi@Hook
222 \AtBegShi@HookNext
223 \gdef\AtBegShi@HookNext{}%
224 \ifAtBegShi@Discarded
225 \@PackageInfoNoLine{atbegshi}{Shipout page discarded}%
226 \global\AtBegShi@Discardedfalse
227 \begingroup
228 \setbox\AtBeginShipoutBox\box\AtBeginShipoutBox
229 \endgroup
230 \let\protect\AtBegShi@OrgProtect
231 \else
232 \AtBegShi@First
233 \let\protect\AtBegShi@OrgProtect
234 \AtBegShi@OrgShipout\box\AtBeginShipoutBox
235 \fi
236 \fi
237 }

238 \catcode'\X=11 %
239 \catcode'\E=11 %
```

\AtBegShi@First

```
240 \def\AtBegShi@First{%
241 \begingroup
242 \def\@empty{}%
243 \ifx\AtBegShi@HookFirst\@empty
244 \else
245 \setbox\z@=\vbox{%
246 \begingroup
247 \AtBegShi@HookFirst
248 \endgroup
249 }%
250 \wd\z@=\z@
251 \ht\z@=\z@
252 \dp\z@=\z@
253 \global\setbox\AtBeginShipoutBox=\vbox{%
254 \baselineskip\z@skip
255 \lineskip\z@skip
256 \lineskiplimit\z@
257 \copy\z@
258 \copy\AtBeginShipoutBox
259 }%
260 \fi
```

```

261     \global\let\AtBegShi@First\@empty
262     \global\let\AtBeginShipoutFirst\AtBegShi@FirstDisabled
263 \endgroup
264 }

\AtBegShi@Hook
265 \gdef\AtBegShi@Hook{}

\AtBegShi@HookNext
266 \gdef\AtBegShi@HookNext{}

\AtBegShi@HookFirst
267 \gdef\AtBegShi@HookFirst{}

\AtBeginShipout
268 \AtBegShi@CheckDefinable\AtBeginShipout
269 \def\AtBeginShipout{%
270   \AtBegShi@AddHook\AtBegShi@Hook
271 }

\AtBeginShipoutNext
272 \AtBegShi@CheckDefinable\AtBeginShipoutNext
273 \def\AtBeginShipoutNext{%
274   \AtBegShi@AddHook\AtBegShi@HookNext
275 }

\AtBeginShipoutFirst
276 \AtBegShi@CheckDefinable\AtBeginShipoutFirst
277 \def\AtBeginShipoutFirst{%
278   \AtBegShi@AddTo\AtBegShi@HookFirst
279 }

\AtBegShi@FirstDisabled
280 \long\def\AtBegShi@FirstDisabled#1{%
281   \@PackageWarning{atbegshi}{%
282     First page is already shipped out, ignoring\MessageBreak
283     \string\AtBeginShipoutFirst
284   }%
285 }

\AtBegShi@AddTo
286 \begingroup\expandafter\expandafter\expandafter\endgroup
287 \expandafter\ifx\csname g@addto@macro\endcsname\relax
288   \long\def\AtBegShi@AddTo#1#2{%
289     \begingroup
290       \toks\z@\expandafter{#1#2}%
291       \xdef#1{\the\toks\z0}%
292     \endgroup
293   }%
294 \else
295   \let\AtBegShi@AddTo\g@addto@macro
296 \fi

\AtBegShi@AddHook
297 \long\def\AtBegShi@AddHook#1#2{%
298   \AtBegShi@AddTo#1{\AtBegShi@Item{#2}}%
299 }

```

\AtBegShi@Item

```

300 \long\def\AtBegShi@Item#1{%
301   \ifAtBegShi@Discarded
302   \else
303     #1%
304     \ifvoid\AtBeginShipoutBox
305       \@PackageWarning{atbegshi}{%
306         Shipout box was voided by hook,\MessageBreak
307         ignoring shipout box%
308       }%
309       \AtBeginShipoutDiscard
310     \fi
311   \fi
312 }
```

\AtBeginShipoutInit

```

313 \AtBegShi@CheckDefinable\AtBeginShipoutInit
314 \def\AtBeginShipoutInit{%
315   \csname newbox\endcsname\AtBeginShipoutBox
316   \AtBegShi@CheckDefinable\AtBegShi@OrgShipout
317   \global\let\AtBegShi@OrgShipout\shipout
318   \global\let\shipout\AtBegShi@Shipout
319   \gdef\AtBeginShipoutInit{}%
320 }

321 \begingroup\expandafter\expandafter\expandafter\endgroup
322 \expandafter\ifx\csname AtBeginDocument\endcsname\relax
323   \AtBeginShipoutInit
324 \else
325   \AtBeginDocument{\AtBeginShipoutInit}%
326 \fi
```

3.4 Positioning

```

327 \begingroup\expandafter\expandafter\expandafter\endgroup
328 \expandafter\ifx\csname RequirePackage\endcsname\relax
329   \input ifpdf.sty\relax
330 \else
331   \RequirePackage{ifpdf}\relax
332 \fi

333 \ifpdf
334   \def\AtBegShi@horigin{\pdfhorigin}%
335   \def\AtBegShi@vorigin{\pdfvorigin}%
336 \else
337   \def\AtBegShi@horigin{72.27pt}%
338   \def\AtBegShi@vorigin{72.27pt}%
339 \fi

340 \begingroup
341 \ifcase
342   \expandafter\ifx\csname picture\endcsname\relax
343     1%
344   \else
345     \expandafter\ifx\csname endpicture\endcsname\relax
346       1%
347     \else
348       0%
349     \fi
350   \fi
351 \endgroup
352 \def\AtBegShi@BeginPicture{%
353   \begingroup
```

```

354     \picture(0,0)\relax
355     \begingroup\expandafter\expandafter\expandafter\endgroup
356     \expandafter\ifx\csname unitlength\endcsname\relax
357     \else
358         \unitlength=1pt\relax
359     \fi
360     \ignorespaces
361 }%
362 \def\AtBegShi@EndPicture{%
363     \endpicture
364     \endgroup
365 }%
366 \else
367     \endgroup
368     \def\AtBegShi@BeginPicture{%
369         \setbox0=\hbox\bgroup
370         \begingroup
371         \ignorespaces
372     }%
373     \def\AtBegShi@EndPicture{%
374         \endgroup
375         \egroup
376         \ht0=0pt\relax
377         \dp0=0pt\relax
378         \copy0 %
379     }%
380 \fi

381 \def\AtBeginShipoutUpperLeft#1{%
382     \global\setbox\AtBeginShipoutBox=\hbox{%
383         \rlap{%
384             \kern-\AtBegShi@horigin\relax
385             \vbox to Opt{%
386                 \kern-\AtBegShi@vorigin\relax
387                 \kern-\ht\AtBeginShipoutBox
388                 \AtBegShi@BeginPicture
389                 #1%
390                 \AtBegShi@EndPicture
391                 \vss
392             }%
393         }%
394         \box\AtBeginShipoutBox
395     }%
396 }

397 \def\AtBeginShipoutUpperLeftForeground#1{%
398     \global\setbox\AtBeginShipoutBox=\hbox to \wd\AtBeginShipoutBox{%
399         \rlap{%
400             \copy\AtBeginShipoutBox
401         }%
402         \rlap{%
403             \kern-\AtBegShi@horigin\relax
404             \vbox to Opt{%
405                 \kern-\AtBegShi@vorigin\relax
406                 \kern-\ht\AtBeginShipoutBox
407                 \AtBegShi@BeginPicture
408                 #1%
409                 \AtBegShi@EndPicture
410                 \vss
411             }%
412         }%
413         \hss
414     }%
415 }

```


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.

```

416 \def\AtBegShi@AbortIfUndefined#1{%
417   \begingroup\expandafter\expandafter\expandafter\endgroup
418   \expandafter\ifx\csname#1\endcsname\relax
419     \AtBegShi@AtEnd
420   \expandafter\endinput
421   \fi
422 }
423 \AtBegShi@AbortIfUndefined{currentgrouplevel}
424 \AtBegShi@AbortIfUndefined{AtBeginDocument}
425 \AtBegShi@AbortIfUndefined{@ifpackageloaded}
426 \AtBegShi@AbortIfUndefined{@ifclassloaded}

```

3.5.1 Package crop

Fix of method and box.

```

427 \def\AtBegShi@PatchCrop{%
428   \begingroup
429   \def\AtBegShi@Crop@shipout{%
430     \afterassignment\CROP@ship
431     \setbox\@cclv=%
432   }%
433   \def\AtBegShi@Crop@ship{%
434     \ifvoid\@cclv
435       \expandafter\aftergroup
436     \fi
437     \CROP@@ship
438   }%
439   \def\AtBegShi@Crop@shiplist{%
440     \lineskip\z@
441     \lineskiplimit\z@
442     \baselineskip\z@
443     \CROP@kernel
444     \box\@cclv
445   }%
446   \def\AtBegShi@Crop@@ship{%
447     \CROP@shipout\vbox{%
448       \CROP@shiplist
449     }%
450   }%
451   \ifx\AtBegShi@Crop@ship\CROP@ship
452     \ifx\AtBegShi@Crop@shiplist\CROP@shiplist
453       \ifx\AtBegShi@Crop@@ship\CROP@@ship
454         \let\AtBegShi@found\relax
455         \ifx\shipout\AtBegShi@Crop@shipout
456           \def\AtBegShi@found{\shipout}%
457         \else\ifx\AtBegShi@OrgShipout\AtBegShi@Crop@shipout
458           \def\AtBegShi@found{\AtBegShi@OrgShipout}%
459         \else\ifx\@EveryShipout@Org@Shipout\AtBegShi@Crop@shipout
460           \def\AtBegShi@found{\@EveryShipout@Org@Shipout}%
461         \else\ifx\GPTorg@shipout\AtBegShi@Crop@shipout
462           \def\AtBegShi@found{\GPTorg@shipout}%
463         \else\ifx\THBorg@shipout\AtBegShi@Crop@shipout
464           \def\AtBegShi@found{\THBorg@shipout}%
465         \else\ifx\mem@oldshipout\AtBegShi@Crop@shipout
466           \def\AtBegShi@found{\mem@oldshipout}%
467         \fi\fi\fi\fi\fi\fi
468         \ifx\AtBegShi@found\relax
469           \else

```

```

470         \expandafter\endgroup
471         \expandafter\def\AtBegShi@found{%
472             \edef\AtBegShi@GroupLevel{\number\currentgrouplevel}%
473             \afterassignment\CROP@ship
474             \setbox\AtBeginShipoutBox=
475         }%
476         \def\CROP@ship{%
477             \ifnum\AtBegShi@GroupLevel=\currentgrouplevel
478             \else
479                 \expandafter\aftergroup
480             \fi
481             \CROP@@ship
482         }%
483         \def\CROP@shiplist{%
484             \lineskip\z@
485             \lineskiplimit\z@
486             \baselineskip\z@
487             \CROP@kernel
488             \box\AtBeginShipoutBox
489         }%
490         \def\CROP@@ship{%
491             \ifvoid\AtBeginShipoutBox
492             \else
493                 \setbox\AtBeginShipoutBox=\vbox{%
494                     \CROP@shiplist
495                 }%
496                 \expandafter\CROP@shipout
497                 \expandafter\box
498                 \expandafter\AtBeginShipoutBox
499             \fi
500         }%
501         \@PackageInfoNoLine{atbegshi}{Package ‘crop’ patched}%
502         \begingroup
503     \fi
504 \fi
505 \fi
506 \fi
507 \endgroup
508 \let\AtBegShi@PatchCrop\relax
509 }
510 \ifpackageloaded{crop}{%
511     \AtBegShi@PatchCrop
512 }{%
513     \AtBeginDocument{\AtBegShi@PatchCrop}%
514 }

```

3.5.2 Package everyshi

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

```

515 \def\AtBegShi@PatchEveryshi{%
516     \begingroup
517     \long\def\AtBegShi@Everyshi@shipout{%
518         \afterassignment\@EveryShipout@Test
519         \global\setbox\@cclv=
520     }%
521     \long\def\AtBegShi@Everyshi@Test{%
522         \ifvoid\@cclv\relax
523             \aftergroup\@EveryShipout@Output
524         \else
525             \@EveryShipout@Output
526         \fi
527     }%

```

```

528 \ifx\AtBegShi@Everyshi@Test\@EveryShipout@Test
529 \let\AtBegShi@found\relax
530 \ifx\shipout\AtBegShi@Everyshi@shipout
531 \def\AtBegShi@found{\shipout}%
532 \else\ifx\AtBegShi@OrgShipout\AtBegShi@Everyshi@shipout
533 \def\AtBegShi@found{\AtBegShi@OrgShipout}%
534 \else\ifx\CROP@shipout\AtBegShi@Everyshi@shipout
535 \def\AtBegShi@found{\CROP@shipout}%
536 \else\ifx\GPTorg@shipout\AtBegShi@Everyshi@shipout
537 \def\AtBegShi@found{\GPTorg@shipout}%
538 \else\ifx\THBorg@shipout\AtBegShi@Everyshi@shipout
539 \def\AtBegShi@found{\THBorg@shipout}%
540 \else\ifx\mem@oldshipout\AtBegShi@Everyshi@shipout
541 \def\AtBegShi@found{\mem@oldshipout}%
542 \else
543 \expandafter\ifx\csname @EveryShipout@Org@Shipout\endcsname
544 \relax
545 \ifx\@EveryShipout@Shipout\AtBegShi@Everyshi@shipout
546 \def\AtBegShi@found{\@EveryShipout@Shipout}%
547 \fi
548 \fi
549 \fi\fi\fi\fi\fi
550 \ifx\AtBegShi@found\relax
551 \else
552 \expandafter\endgroup
553 \expandafter\def\AtBegShi@found{%
554 \edef\AtBegShi@GroupLevel{\number\currentgrouplevel}%
555 \afterassignment\@EveryShipout@Test
556 \setbox\AtBeginShipoutBox=%
557 }%
558 \def\@EveryShipout@Test{%
559 \ifnum\AtBegShi@GroupLevel=\currentgrouplevel
560 \else
561 \expandafter\aftergroup
562 \fi
563 \AtBegShi@Everyshi@Output
564 }%
565 \def\AtBegShi@Everyshi@Output{%
566 \ifvoid\AtBeginShipoutBox
567 \else
568 \global\setbox\@cclv\box\AtBeginShipoutBox
569 \expandafter\@EveryShipout@Output
570 \fi
571 }%
572 \@PackageInfoNoLine{atbegshi}{Package 'everyshi' patched}%
573 \begingroup
574 \fi
575 \fi
576 \endgroup
577 \let\AtBegShi@PatchEveryshi\relax
578 }
579 \ifpackageloaded{everyshi}{%
580 \AtBegShi@PatchEveryshi
581 }{%
582 \AtBeginDocument{\AtBegShi@PatchEveryshi}%
583 }

```

3.5.3 Class memoir

Fix of method and box.

```

584 \def\AtBegShi@PatchMemoir{%
585 \begingroup

```

```

586 \def\AtBegShi@Memoir@shipout{%
587 \afterassignment\mem@shipi
588 \setbox\@cclv=%
589 }%
590 \def\AtBegShi@Memoir@shipi{%
591 \ifvoid\@cclv
592 \expandafter\aftergroup
593 \fi
594 \mem@shipii
595 }%
596 \def\AtBegShi@Memoir@shipiiA{%
597 \mem@oldshipout\vbox{%
598 \trimmarks
599 \unvbox\@cclv
600 }%
601 }%
602 \def\AtBegShi@Memoir@shipiiB{%
603 \ifvoid\@cclv
604 \mem@oldshipout\box\@cclv
605 \else
606 \mem@oldshipout\vbox{%
607 \trimmarks
608 \unvbox\@cclv
609 }%
610 \fi
611 }%
612 \ifx\AtBegShi@Memoir@shipi\mem@shipi
613 \ifcase\ifx\AtBegShi@Memoir@shipiiA\mem@shipii
614 \z@
615 \else
616 \ifx\AtBegShi@Memoir@shipiiB\mem@shipii
617 \z@
618 \else
619 \@ne
620 \fi
621 \fi
622 \let\AtBegShi@found\relax
623 \ifx\shipout\AtBegShi@Memoir@shipout
624 \def\AtBegShi@found{\shipout}%
625 \else\ifx\AtBegShi@OrgShipout\AtBegShi@Memoir@shipout
626 \def\AtBegShi@found{\AtBegShi@OrgShipout}%
627 \else\ifx\CROP@shipout\AtBegShi@Memoir@shipout
628 \def\AtBegShi@found{\CROP@shipout}%
629 \else\ifx\GPTorg@shipout\AtBegShi@Memoir@shipout
630 \def\AtBegShi@found{\GPTorg@shipout}%
631 \else\ifx\THBorg@shipout\AtBegShi@Memoir@shipout
632 \def\AtBegShi@found{\THBorg@shipout}%
633 \else\ifx\@EveryShipout@Org@Shipout\AtBegShi@Memoir@shipout
634 \def\AtBegShi@found{\@EveryShipout@Org@Shipout}%
635 \fi\fi\fi\fi\fi\fi
636 \ifx\AtBegShi@found\relax
637 \else
638 \expandafter\endgroup
639 \expandafter\def\AtBegShi@found{%
640 \edef\AtBegShi@GroupLevel{\number\currentgrouplevel}%
641 \afterassignment\mem@shipi
642 \setbox\AtBeginShipoutBox=%
643 }%
644 \def\mem@shipi{%
645 \ifnum\AtBegShi@GroupLevel=\currentgrouplevel
646 \else
647 \expandafter\aftergroup

```

```

648         \fi
649         \mem@shipii
650     }%
651     \def\mem@shipii{%
652         \ifvoid\AtBeginShipoutBox
653         \else
654             \setbox\AtBeginShipoutBox=\vbox{%
655                 \trimmarks
656                 \ifvbox\AtBeginShipoutBox
657                     \unvbox\AtBeginShipoutBox
658                 \else
659                     \box\AtBeginShipoutBox
660                 \fi
661             }%
662             \expandafter\mem@oldshipout
663             \expandafter\box
664             \expandafter\AtBeginShipoutBox
665         \fi
666     }%
667     \@PackageInfoNoLine{atbegshi}{Class 'memoir' patched}%
668     \begingroup
669     \fi
670     \fi
671     \fi
672 \endgroup
673 \let\AtBegShi@PatchMemoir\relax
674 }
675 \@ifclassloaded{memoir}{%
676     \AtBegShi@PatchMemoir
677 }{%
678     \AtBeginDocument{\AtBegShi@PatchMemoir}%
679 }
680 \AtBegShi@AtEnd
681 \end{package}

```

4 Test

4.1 Catcode checks for loading

```

682 \test1
683 \catcode'\{=1 %
684 \catcode'\}=2 %
685 \catcode'\#=6 %
686 \catcode'\@=11 %
687 \expandafter\ifx\csname count@\endcsname\relax
688     \countdef\count@=255 %
689 \fi
690 \expandafter\ifx\csname @gobble\endcsname\relax
691     \long\def\@gobble#1{}%
692 \fi
693 \expandafter\ifx\csname @firstofone\endcsname\relax
694     \long\def\@firstofone#1{#1}%
695 \fi
696 \expandafter\ifx\csname loop\endcsname\relax
697     \expandafter\@firstofone
698 \else
699     \expandafter\@gobble
700 \fi
701 {%
702     \def\loop#1\repeat{%
703         \def\body{#1}%

```

```

704     \iterate
705 }%
706 \def\iterate{%
707     \body
708     \let\next\iterate
709     \else
710     \let\next\relax
711     \fi
712     \next
713 }%
714 \let\repeat=\fi
715 }%
716 \def\RestoreCatcodes{}
717 \count@=0 %
718 \loop
719     \edef\RestoreCatcodes{%
720         \RestoreCatcodes
721         \catcode\the\count@=\the\catcode\count@\relax
722     }%
723 \ifnum\count@<255 %
724     \advance\count@ 1 %
725 \repeat
726
727 \def\RangeCatcodeInvalid#1#2{%
728     \count@=#1\relax
729     \loop
730         \catcode\count@=15 %
731         \ifnum\count@<#2\relax
732             \advance\count@ 1 %
733         \repeat
734 }
735 \expandafter\ifx\csname LoadCommand\endcsname\relax
736 \def\LoadCommand{\input atbegshi.sty\relax}%
737 \fi
738 \def\Test{%
739     \RangeCatcodeInvalid{0}{47}%
740     \RangeCatcodeInvalid{58}{64}%
741     \RangeCatcodeInvalid{91}{96}%
742     \RangeCatcodeInvalid{123}{255}%
743     \catcode'\@=12 %
744     \catcode'\=0 %
745     \catcode'\{=1 %
746     \catcode'\}=2 %
747     \catcode'\#=6 %
748     \catcode'\[=12 %
749     \catcode'\]=12 %
750     \catcode'\%=14 %
751     \catcode'\ =10 %
752     \catcode13=5 %
753     \LoadCommand
754     \RestoreCatcodes
755 }
756 \Test
757 \csname @@end\endcsname
758 \end
759 </test1>
760 <*test2>
761 \input atbegshi.sty\relax
762 \def\msg#{\immediate\write16}
763 \msg{File: atbegshi-test2.tex 2008/07/31 v1.9 Test file for plain-TeX}
764 \def\testmsg#1#2{%
765     \msg{}%

```

```

766 \msg{*** Test with box (#1), expected page output [#2]]% hash-ok
767 }
768
769 \newbox\voidbox
770 \def\void{\box\voidbox}
771 \begingroup
772 \setbox\voidbox=\void
773 \endgroup
774
775 \count0=0\relax
776 \AtBeginShipout{%
777 \global\advance\count0 by 1\relax
778 \msg{* Inside \string\AtBeginShipout: [\the\count0]]%
779 }
780
781 \AtBeginShipoutFirst{%
782 \msg{* Inside \string\AtBeginShipoutFirst}%
783 Hello World%
784 }
785
786 \testmsg{\string\null}{1}
787 \shipout\null
788
789 \AtBeginShipoutFirst{%
790 This is too late%
791 }
792
793 \testmsg{void}{}
794 \shipout\void
795
796 \testmsg{\string\copy255 (not void)}{2}
797 \setbox255\hbox{\vrule height 10bp width 10bp}
798 \shipout\copy255 %
799
800 \testmsg{\string\copy255 (again)}{3}
801 \shipout\copy255 %
802
803 \testmsg{\string\box255}{4}
804 \shipout\box255 %
805
806 \testmsg{\string\box255 (again)}{}
807 \shipout\box255 %
808
809 \testmsg{\string\hbox}{5}
810 \shipout\hbox{\vrule height 5bp width 20bp}
811
812 \testmsg{\string\vbox}{6}
813 \shipout\vbox{\hrule height 20bp width 5bp}
814
815 \testmsg{\string\null, voided by hook}{}
816 \def\VoidBox{%
817 \begingroup
818 \setbox\AtBeginShipoutBox=\box\AtBeginShipoutBox
819 \endgroup
820 }
821 \AtBeginShipout{\VoidBox}
822 \shipout\null
823 \def\VoidBox{}
824
825 \msg{*** \string\begingroup}
826 \begingroup
827 \testmsg{void}{}%

```

```

828 \shipout\void
829 \msg{*** \string\endgroup}
830 \endgroup
831
832 \msg{*** \string\begingroup}
833 \begingroup
834 \testmsg{void}{}%
835 \shipout\void
836 \testmsg{\string\null}{8}%
837 \shipout\null
838 \msg{*** \string\endgroup}
839 \endgroup
840
841 \testmsg{output routine}{9}
842 Hello World
843 \vfill
844 \eject
845
846 \testmsg{\string\null\space(discarded)}{}
847 \AtBeginShipout{%
848 \msg{* Inside \string\AtBeginShipout: DISCARD}%
849 \AtBeginShipoutDiscard
850 }
851 \shipout\null
852
853 \end
854 </test2>

855 <*test3>
856 \NeedsTeXFormat{LaTeX2e}
857 \ProvidesFile{atbegshi-test3.tex}[2008/07/31 v1.9 Test file for LaTeX]
858 \RequirePackage{color}
859 \pagecolor{yellow}
860 \documentclass[a5paper,showtrims]{memoir}
861 \usepackage{atbegshi}
862 \AtBeginShipout{%
863 \setbox\AtBeginShipoutBox=\vbox{%
864 \vbox to Opt{%
865 \kern-1.5in %
866 \hbox to Opt{%
867 \kern-1.5in %
868 \color{blue}%
869 \rule{1in}{1in}%
870 \hss
871 }%
872 \vss
873 }%
874 \hrule
875 \hbox{\vrule\box\AtBeginShipoutBox\vrule}%
876 \hrule
877 }%
878 }
879 \usepackage{eso-pic}
880 \makeatletter
881 \@EveryShipout@Init
882 \let\@EveryShipout@Init\relax
883 \makeatother
884 \AddToShipoutPicture{%
885 \hspace{.52\paperwidth}%
886 \colorbox{cyan}{%
887 \rule{0mm}{\paperheight}%
888 \hspace{.48\paperwidth}%
889 }%

```


890 }

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

```
891 \expandafter\let\csname ver@crop.sty\endcsname\relax
892 \usepackage[color=red,cross,a4,center]{crop}
893 \begin{document}
894 \shipout\null
895 \shipout\box\csname voidb@x\endcsname
896 \section{Hello World}
897 \end{document}
898 \test3
```

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:install/macros/latex/contrib/oberdiek.tds.zip](#)

TDS refers to the standard “A Directory Structure for \TeX Files” ([CTAN:tds/tds.pdf](#)). Directories with `texmf` in their name are usually organized this way.

5.2 Bundle installation

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

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

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

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

5.3 Package installation

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

```
tex atbegshi.dtx
```

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

¹<http://ftp.ctan.org/tex-archive/>

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

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

5.4 Refresh file name databases

If your \TeX distribution (`te \TeX` , `mik \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 `pdf \LaTeX` :

```
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 -TeX is missing.
- `\AtBeginShipoutDiscard` also resets `\deadcycles`.

[2007/04/19 v1.2]

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

[2007/04/26 v1.3]

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

[2007/04/27 v1.4]

- Small optimizations.

[2007/06/06 v1.5]

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

[2007/09/09 v1.6]

- Catcode section rewritten.

[2008/07/18 v1.7]

- Documentation of `\AtBeginShipoutUpperLeft` fixed and extended.

[2008/07/19 v1.8]

- `\AtBeginShipoutUpperLeftForeground` added.

[2008/07/31 v1.9]

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

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@Org@Shipout</code>	
<code>\#</code>	685, 747		459, 460, 633, 634
<code>\%</code>	750	<code>\@EveryShipout@Output</code> .	523, 525, 569
<code>\@</code>	686, 743	<code>\@EveryShipout@Shipout</code>	545, 546
<code>\@EveryShipout@Init</code>	881, 882	<code>\@EveryShipout@Test</code> .	518, 528, 555, 558

\@PackageInfoNoLine	225, 501, 572, 667	\AtBegShi@Everyshi@Output	563, 565
\@PackageWarning	216, 281, 305	\AtBegShi@Everyshi@shipout	517, 530, 532, 534, 536, 538, 540, 545
\@cclv	431, 434, 444, 519, 522, 568, 588, 591, 599, 603, 604, 608	\AtBegShi@Everyshi@Test	521, 528
\@empty	242, 243, 261	\AtBegShi@First	232, 240
\@firstofone	694, 697	\AtBegShi@FirstDisabled	262, 280
\@gobble	691, 699	\AtBegShi@found	454, 456, 458, 460, 462, 464, 466, 468, 471, 529, 531, 533, 535, 537, 539, 541, 546, 550, 553, 622, 624, 626, 628, 630, 632, 634, 636, 639
\@ifclassloaded	675	\AtBegShi@GroupLevel	200, 207, 472, 477, 554, 559, 640, 645
\@ifdefinable	179	\AtBegShi@Hook	221, 265, 270
\@ifpackageloaded	510, 579	\AtBegShi@HookFirst	243, 247, 267, 278
\@one	162, 165, 619	\AtBegShi@HookNext	222, 223, 266, 274
\@undefined	108, 164	\AtBegShi@horigin	334, 337, 384, 403
\[748	\AtBegShi@Item	298, 300
\\	744	\AtBegShi@Memoir@shipi	590, 612
\{	683, 745	\AtBegShi@Memoir@shipiiA	596, 613
\}	684, 746	\AtBegShi@Memoir@shipiiB	602, 616
\]	749	\AtBegShi@Memoir@shipout	586, 623, 625, 627, 629, 631, 633
_	751	\AtBegShi@OrgProtect	218, 230, 233
A			
\AddToShipoutPicture	884	\AtBegShi@OrgShipout	234, 316, 317, 457, 458, 532, 533, 625, 626
\advance	724, 732, 777	\AtBegShi@Output	210, 212
\afterassignment	201, 430, 473, 518, 555, 587, 641	\AtBegShi@PatchCrop	427, 508, 511, 513
\aftergroup	208, 435, 479, 523, 561, 592, 647	\AtBegShi@PatchEveryshi	515, 577, 580, 582
\AtBeginDocument	325, 513, 582, 678	\AtBegShi@PatchMemoir	584, 673, 676, 678
\AtBeginShipout	2, 6, 44, 268, 776, 778, 821, 847, 848, 862	\AtBegShi@Shipout	196, 318
\AtBeginShipoutBox	45, 47, 198, 203, 215, 228, 234, 253, 258, 304, 315, 382, 387, 394, 398, 400, 406, 474, 488, 491, 493, 498, 556, 566, 568, 642, 652, 654, 656, 657, 659, 664, 818, 863, 875	\AtBegShi@Test	201, 205
\AtBeginShipoutDiscard	3, 29, 183, 309, 849	\AtBegShi@vorigin	335, 338, 386, 405
\AtBeginShipoutFirst	3, 262, 276, 283, 781, 782, 789	B	
\AtBeginShipoutInit	3, 313, 323, 325	\baselineskip	254, 442, 486
\AtBeginShipoutNext	2, 14, 28, 272	\begin	11, 50, 893
\AtBeginShipoutUpperLeft	3, 7, 15, 381	\body	703, 707
\AtBeginShipoutUpperLeftForeground	3, 397	\box	47, 228, 234, 394, 444, 488, 497, 568, 604, 659, 663, 770, 803, 804, 806, 807, 818, 875, 895
\AtBegShi@AbortIfUndefined	416, 423, 424, 425, 426	C	
\AtBegShi@AddHook	270, 274, 297	\catcode	58, 59, 60, 61, 62, 73, 74, 75, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 120, 121, 124, 125, 126, 127, 131, 132, 133, 134, 138, 140, 190, 191, 193, 194, 238, 239, 683, 684, 685, 686, 721, 730, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752
\AtBegShi@AddTo	278, 286, 298	\circle	8
\AtBegShi@AtEnd	136, 137, 419, 680	\color	16, 868
\AtBegShi@BeginPicture	352, 368, 388, 407	\colorbox	886
\AtBegShi@CheckDefinable	158, 183, 268, 272, 276, 313, 316	\copy	257, 258, 378, 400, 796, 798, 800, 801
\AtBegShi@Crop@ship	446, 453	\count	775, 777, 778
\AtBegShi@Crop@ship	433, 451	\count@	688, 717, 721, 723, 724, 728, 730, 731, 732
\AtBegShi@Crop@shiplist	439, 452	\countdef	688
\AtBegShi@Crop@shipout	429, 455, 457, 459, 461, 463, 465	\CROP@@ship	437, 453, 481, 490
\AtBegShi@Discardedfalse	220, 226		
\AtBegShi@Discardedtrue	186		
\AtBegShi@EndPicture	362, 373, 390, 409		

<code>\rule</code>	869, 887	<code>\unvbox</code>	599, 608, 657
		<code>\usepackage</code>	3, 4, 5, 38, 39, 861, 879, 892
S			
<code>\section</code>	12, 896	V	
<code>\setbox</code>	45, 198, 203,	<code>\vbox</code> ..	245, 253, 385, 404, 447, 493,
	228, 245, 253, 369, 382, 398,		597, 606, 654, 812, 813, 863, 864
	431, 474, 493, 519, 556, 568,	<code>\vfill</code>	843
	588, 642, 654, 772, 797, 818, 863	<code>\void</code>	770, 772, 794, 828, 835
<code>\shipout</code>	317, 318, 455, 456,	<code>\VoidBox</code>	816, 821, 823
	530, 531, 623, 624, 787, 794,	<code>\voidbox</code>	769, 770, 772
	798, 801, 804, 807, 810, 813,	<code>\vrule</code>	797, 810, 875
	822, 828, 835, 837, 851, 894, 895	<code>\vspace</code>	25
<code>\space</code>	171, 846	<code>\vss</code>	391, 410, 872
<code>\special</code>	46		
T			
<code>\Test</code>	738, 756	W	
<code>\testmsg</code>	764, 786,	<code>\wd</code>	250, 398
	793, 796, 800, 803, 806, 809,	<code>\write</code>	78, 102, 762
	812, 815, 827, 834, 836, 841, 846		
<code>\THBorg@shipout</code>		X	
	463, 464, 538, 539, 631, 632	<code>\X</code>	238
<code>\the</code>	124, 125, 126, 127, 138, 291, 721, 778	<code>\x</code>	63, 65, 67,
<code>\TMP@EnsureCode</code>	135, 142, 143, 144,		77, 81, 83, 101, 106, 116, 122, 130
	145, 146, 147, 148, 149, 150, 151		
<code>\toks</code>	290, 291	Z	
<code>\trimmarks</code>	598, 607, 655	<code>\z@</code> ...	167, 185, 206, 245, 250, 251,
			252, 256, 257, 290, 291, 440,
			441, 442, 484, 485, 486, 614, 617
U			
<code>\unitlength</code>	358	<code>\z@skip</code>	254, 255