

The atbegshi package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2008/07/19 v1.8

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	Example	3
2	Method of <code>\shipout</code> overloading	4
2.1	<code>\shipout</code>	4
2.2	<code>\afterassignment</code>	4
2.3	Test for direct or indirect boxes	5
2.3.1	With ε -TeX	5
2.3.2	Without ε -TeX	6
2.3.3	<code>\lastkern</code> method	7
2.4	Output	7
2.5	Separate box register	7
2.6	Summary	8
2.6.1	With ε -TeX	8
2.6.2	Without ε -TeX, traditional way	8
2.6.3	<code>\lastkern</code> method	9
3	Implementation	9
3.1	Reload check and package identification	9
3.2	Catcodes	10
3.3	Preparations	11
3.4	Positioning	14
3.5	Patches	16
3.5.1	Package <code>crop</code>	16
3.5.2	Package <code>everyshi</code>	18
3.5.3	Class <code>memoir</code>	19
4	Test	20
4.1	Catcode checks for loading	20
5	Installation	24
5.1	Download	24
5.2	Bundle installation	24
5.3	Package installation	25
5.4	Refresh file name databases	25
5.5	Some details for the interested	25

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

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

```
1 <example>
2 \documentclass[a4paper]{article}
3 \usepackage{color}
4 \usepackage{atbegshi}
```

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

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

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

```

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

```

2 Method of `\shipout` overloading

2.1 `\shipout`

The \TeX primitive command `\shipout` takes a box specification and puts the box as a new page in the output file. There are two kinds of box specifications:

Direct boxes: They are given by `\hbox`, `\vbox`, or `\vtop`,
e.g. `\shipout\hbox{Hello World}`.

Indirect boxes: `\box` or `\copy` references a box register by number. The box register contains the contents of the box.

Note: `\box` also clears the box register globally.

Then we have to differentiate between void and empty boxes:

Void: Initially or after `\box` there is no box in the box register. In this cases the box register is not empty, but *void*.

Empty: A box with empty contents, such as `\hbox{}` ($= \text{\null}$) 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 ⇒ \setbox\mybox=\box0 \@test

```

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

```

\shipout\hbox{Hello World}
⇒ \setbox\mybox=\hbox{\@test Hello World}

```

2.3 Test for direct or indirect boxes

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

2.3.1 With ε -TeX

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

```

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

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

```

2.3.2 Without ε -TeX

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

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

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

Unhappily all these qualities even combined are not sufficient for constructing an initial box value, because `\shipout` can be called with a box that is accidentally just the same as the 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
\begin{group}
  \setbox\mybox=\box\mybox % ensure \mybox is void
\end{group}
\let\original@shipout\shipout

\shipout :=
  % trick to get a void box \mybox
  \begin{group}
    \setbox\mybox=\box\mybox
  \end{group}
  \afterassignment\@test
  \setbox\mybox=

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

\@output :=
```



```

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

```

2.6.3 \lastkern method

And for \TeX without $\varepsilon\text{-}\text{\TeX}$ 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 $\varepsilon\text{-}\text{\TeX}$'s `\currentgrouplevel`, if it is available. Otherwise the `\lastkern` method is used.

```

36 \package

```

3.1 Reload check and package identification

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

```

37 \begingroup
38 \catcode44 12 % ,
39 \catcode45 12 % -
40 \catcode46 12 % .
41 \catcode58 12 % :
42 \catcode64 11 % @
43 \expandafter\let\expandafter\x\csname ver@atbegshi.sty\endcsname

```

```

44 \ifcase 0%
45   \ifx\x\relax % plain
46   \else
47     \ifx\x\empty % LaTeX
48     \else
49       1%
50     \fi
51   \fi
52 \else
53   \catcode35 6 % #
54   \catcode123 1 % {
55   \catcode125 2 % }
56   \expandafter\ifx\csname PackageInfo\endcsname\relax
57     \def\x#1#2{%
58       \immediate\write-1{Package #1 Info: #2.}%
59     }%
60   \else
61     \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
62   \fi
63   \x{atbegshi}{The package is already loaded}%
64   \endgroup
65   \expandafter\endinput
66 \fi
67 \endgroup

```

Package identification:

```

68 \begingroup
69   \catcode35 6 % #
70   \catcode40 12 % (
71   \catcode41 12 % )
72   \catcode44 12 % ,
73   \catcode45 12 % -
74   \catcode46 12 % .
75   \catcode47 12 % /
76   \catcode58 12 % :
77   \catcode64 11 % @
78   \catcode123 1 % {
79   \catcode125 2 % }
80   \expandafter\ifx\csname ProvidesPackage\endcsname\relax
81     \def\x#1#2#3[#4]{\endgroup
82       \immediate\write-1{Package: #3 #4}%
83       \xdef#1{#4}%
84     }%
85   \else
86     \def\x#1#2[#3]{\endgroup
87       #2[#{#3}]%
88       \ifx#1\relax
89         \xdef#1{#3}%
90       \fi
91     }%
92   \fi
93 \expandafter\x\csname ver@atbegshi.sty\endcsname
94 \ProvidesPackage{atbegshi}%
95 [2008/07/19 v1.8 At begin shipout hook (H0)]

```

3.2 Catcodes

```

96 \begingroup
97   \catcode123 1 % {
98   \catcode125 2 % }
99   \def\x{\endgroup
100     \expandafter\edef\csname AtBegShi@AtEnd\endcsname{%
101       \catcode35 \the\catcode35\relax

```

```

102      \catcode64 \the\catcode64\relax
103      \catcode123 \the\catcode123\relax
104      \catcode125 \the\catcode125\relax
105    }%
106  }%
107  \x
108  \catcode35 6 % #
109  \catcode64 11 % @
110  \catcode123 1 % {
111  \catcode125 2 % }
112  \def\TMP@EnsureCode#1#2{%
113    \edef\AtBegShi@AtEnd{%
114      \AtBegShi@AtEnd
115      \catcode#1 \the\catcode#1\relax
116    }%
117    \catcode#1 #2\relax
118  }
119  \TMP@EnsureCode{40}{12}% (
120  \TMP@EnsureCode{41}{12}% )
121  \TMP@EnsureCode{44}{12}% ,
122  \TMP@EnsureCode{45}{12}% -
123  \TMP@EnsureCode{47}{12}% /
124  \TMP@EnsureCode{46}{12}% .
125  \TMP@EnsureCode{58}{12}% :
126  \TMP@EnsureCode{61}{12}% =
127  \TMP@EnsureCode{94}{7}% ^ (superscript)
128  \TMP@EnsureCode{96}{12}% ‘

```

3.3 Preparations

```

129 \begingroup\expandafter\expandafter\expandafter\endgroup
130 \expandafter\ifx\csname RequirePackage\endcsname\relax
131   \input infwarerr.sty\relax
132 \else
133   \RequirePackage{infwarerr}[2007/09/09]%
134 \fi

```

\AtBegShi@CheckDefinable

```

135 \begingroup\expandafter\expandafter\expandafter\endgroup
136 \expandafter\ifx\csname @ifdefinable\endcsname\relax
137   \def\AtBegShi@CheckDefinable#1{%
138     \ifcase\ifx#1\relax
139       \@ne
140     \else
141       \ifx#1\@undefined
142         \@ne
143       \else
144         \z@
145       \fi
146     \fi
147     \errmessage{%
148       Package atbegshi: \string#1\space
149       is already defined%
150     }%
151   \endgroup
152 \fi
153 }%
154 \else
155   \def\AtBegShi@CheckDefinable#1{%
156     \@ifdefinable{#1}{}%
157   }%
158 \fi

159 \newif\ifAtBegShi@Discarded

```

```

\AtBeginShipoutDiscard
160 \AtBegShi@CheckDefinable\AtBeginShipoutDiscard
161 \def\AtBeginShipoutDiscard{%
162   \deadcycles=\z@
163   \global\AtBegShi@Discardedtrue
164 }

165 \begingroup\expandafter\expandafter\expandafter\endgroup
166 \expandafter\ifx\csname currentgrouplevel\endcsname\relax
167   \catcode'X=9 % ignore
168   \catcode'E=14 % comment
169 \else
170   \catcode'X=14 % comment
171   \catcode'E=9 % ignore
172 \fi

\AtBegShi@Shipout
173 \def\AtBegShi@Shipout{%
174 X \begingroup
175 X \setbox\AtBeginShipoutBox=\hbox\bgroup
176 X \kern\p@
177 E \edef\AtBegShi@GroupLevel{\number\currentgrouplevel}%
178 \afterassignment\AtBegShi@Test
179 X \global
180 \setbox\AtBeginShipoutBox=%
181 }

\AtBegShi@Test
182 \def\AtBegShi@Test{%
183 X \ifdim\lastkern=\z@
184 E \ifnum\AtBegShi@GroupLevel<\currentgrouplevel
185   \expandafter\aftergroup
186   \fi
187   \AtBegShi@Output
188 }

\AtBegShi@Output
189 \def\AtBegShi@Output{%
190 X \egroup
191 X \endgroup
192 \ifvoid\AtBeginShipoutBox
193   \@PackageWarning{atbegshi}{Ignoring void shipout box}%
194 \else
195   \let\AtBegShi@OrgProtect\protect
196   \csname set@typeset@protect\endcsname
197   \global\AtBegShi@Discardedfalse
198   \AtBegShi@Hook
199   \AtBegShi@HookNext
200   \gdef\AtBegShi@HookNext{}%
201   \ifAtBegShi@Discarded
202     \@PackageInfoNoLine{atbegshi}{Shipout page discarded}%
203     \global\AtBegShi@Discardedfalse
204     \begingroup
205     \setbox\AtBeginShipoutBox\box\AtBeginShipoutBox
206     \endgroup
207     \let\protect\AtBegShi@OrgProtect
208   \else
209     \AtBegShi@First
210     \let\protect\AtBegShi@OrgProtect
211     \AtBegShi@OrgShipout\box\AtBeginShipoutBox
212   \fi
213 \fi
214 }

```

```

215 \catcode'\X=11 %
216 \catcode'\E=11 %

\AtBegShi@First
217 \def\AtBegShi@First{%
218   \begingroup
219   \def\@empty{%
220     \ifx\AtBegShi@HookFirst\@empty
221     \else
222       \setbox\z@=\vbox{%
223         \begingroup
224           \AtBegShi@HookFirst
225         \endgroup
226       }%
227       \wd\z@=\z@
228       \ht\z@=\z@
229       \dp\z@=\z@
230       \global\setbox\AtBeginShipoutBox=\vbox{%
231         \baselineskip\z@skip
232         \lineskip\z@skip
233         \lineskiplimit\z@
234         \copy\z@
235         \copy\AtBeginShipoutBox
236       }%
237     \fi
238     \global\let\AtBegShi@First\@empty
239     \global\let\AtBeginShipoutFirst\AtBegShi@FirstDisabled
240   \endgroup
241 }

\AtBegShi@Hook
242 \gdef\AtBegShi@Hook{}

\AtBegShi@HookNext
243 \gdef\AtBegShi@HookNext{}

\AtBegShi@HookFirst
244 \gdef\AtBegShi@HookFirst{}

\AtBeginShipout
245 \AtBegShi@CheckDefinable\AtBeginShipout
246 \def\AtBeginShipout{%
247   \AtBegShi@AddHook\AtBegShi@Hook
248 }

\AtBeginShipoutNext
249 \AtBegShi@CheckDefinable\AtBeginShipoutNext
250 \def\AtBeginShipoutNext{%
251   \AtBegShi@AddHook\AtBegShi@HookNext
252 }

\AtBeginShipoutFirst
253 \AtBegShi@CheckDefinable\AtBeginShipoutFirst
254 \def\AtBeginShipoutFirst{%
255   \AtBegShi@AddTo\AtBegShi@HookFirst
256 }

\AtBegShi@FirstDisabled
257 \long\def\AtBegShi@FirstDisabled#1{%
258   \@PackageWarning{atbegshi}{%
259     First page is already shipped out, ignoring\MessageBreak

```

```

260     \string\AtBeginShipoutFirst
261 }%
262 }

\AtBegShi@AddTo

263 \begingroup\expandafter\expandafter\expandafter\endgroup
264 \expandafter\ifx\csname g@addto@macro\endcsname\relax
265   \long\def\AtBegShi@AddTo#1#2{%
266     \begingroup
267       \toks\z@\expandafter{#1#2}%
268       \xdef#1{\the\toks\z@}%
269     \endgroup
270   }%
271 \else
272   \let\AtBegShi@AddTo\g@addto@macro
273 \fi

\AtBegShi@AddHook

274 \long\def\AtBegShi@AddHook#1#2{%
275   \AtBegShi@AddTo#1{\AtBegShi@Item{#2}}%
276 }

\AtBegShi@Item

277 \long\def\AtBegShi@Item#1{%
278   \ifAtBegShi@Discarded
279   \else
280     #1%
281     \ifvoid\AtBeginShipoutBox
282       \@PackageWarning{atbegshi}{%
283         Shipout box was voided by hook,\MessageBreak
284         ignoring shipout box%
285       }%
286       \AtBeginShipoutDiscard
287     \fi
288   \fi
289 }

\AtBeginShipoutInit

290 \AtBegShi@CheckDefinable\AtBeginShipoutInit
291 \def\AtBeginShipoutInit{%
292   \csname newbox\endcsname\AtBeginShipoutBox
293   \AtBegShi@CheckDefinable\AtBegShi@OrgShipout
294   \global\let\AtBegShi@OrgShipout\shipout
295   \global\let\shipout\AtBegShi@Shipout
296   \gdef\AtBeginShipoutInit{}%
297 }

298 \begingroup\expandafter\expandafter\expandafter\endgroup
299 \expandafter\ifx\csname AtBeginDocument\endcsname\relax
300   \AtBeginShipoutInit
301 \else
302   \AtBeginDocument{\AtBeginShipoutInit}%
303 \fi

```

3.4 Positioning

```

304 \begingroup\expandafter\expandafter\expandafter\endgroup
305 \expandafter\ifx\csname RequirePackage\endcsname\relax
306   \input ifpdf.sty\relax
307 \else
308   \RequirePackage{ifpdf}\relax
309 \fi

```

```

310 \ifpdf
311   \def\AtBegShi@horigin{\pdfhorigin}%
312   \def\AtBegShi@vorigin{\pdfvorigin}%
313 \else
314   \def\AtBegShi@horigin{72.27pt}%
315   \def\AtBegShi@vorigin{72.27pt}%
316 \fi

317 \begingroup
318 \ifcase
319   \expandafter\ifx\csname picture\endcsname\relax
320     1%
321   \else
322     \expandafter\ifx\csname endpicture\endcsname\relax
323       1%
324     \else
325       0%
326     \fi
327   \fi
328 \endgroup
329 \def\AtBegShi@BeginPicture{%
330   \begingroup
331   \picture(0,0)\relax
332   \begingroup\expandafter\expandafter\expandafter\endgroup
333   \expandafter\ifx\csname unitlength\endcsname\relax
334     \else
335       \unitlength=1pt\relax
336     \fi
337   \ignorespaces
338 }%
339 \def\AtBegShi@EndPicture{%
340   \endpicture
341   \endgroup
342 }%
343 \else
344   \endgroup
345   \def\AtBegShi@BeginPicture{%
346     \setbox0=\hbox\bgroup
347     \begingroup
348     \ignorespaces
349   }%
350   \def\AtBegShi@EndPicture{%
351     \endgroup
352     \egroup
353     \ht0=0pt\relax
354     \dp0=0pt\relax
355     \copy0 %
356   }%
357 \fi

358 \def\AtBeginShipoutUpperLeft#1{%
359   \global\setbox\AtBeginShipoutBox=\hbox{%
360     \rlap{%
361       \kern-\AtBegShi@horigin\relax
362       \vbox to 0pt{%
363         \kern-\AtBegShi@vorigin\relax
364         \kern-\ht\AtBeginShipoutBox
365         \AtBegShi@BeginPicture
366         #1%
367         \AtBegShi@EndPicture
368         \vss
369       }%
370     }%
371     \box\AtBeginShipoutBox

```

```

372 }%
373 }
374 \def\AtBeginShipoutUpperLeftForeground#1{%
375   \global\setbox\AtBeginShipoutBox=\hbox to \wd\AtBeginShipoutBox{%
376     \rlap{%
377       \copy\AtBeginShipoutBox
378     }%
379     \rlap{%
380       \kern-\AtBegShi@horigin\relax
381       \vbox to \Opt{%
382         \kern-\AtBegShi@vorigin\relax
383         \kern-\ht\AtBeginShipoutBox
384         \AtBegShi@BeginPicture
385         #1%
386         \AtBegShi@endPicture
387         \vss
388       }%
389     }%
390     \hss
391   }%
392 }

```

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.

```

393 \def\AtBegShi@AbortIfUndefined#1{%
394   \begingroup\expandafter\expandafter\expandafter\endgroup
395   \expandafter\ifx\csname#1\endcsname\relax
396     \AtBegShi@AtEnd
397   \expandafter\endinput
398 \fi
399 }
400 \AtBegShi@AbortIfUndefined{currentgrouplevel}
401 \AtBegShi@AbortIfUndefined{AtBeginDocument}
402 \AtBegShi@AbortIfUndefined{@ifpackageloaded}
403 \AtBegShi@AbortIfUndefined{@ifclassloaded}

```

3.5.1 Package crop

Fix of method and box.

```

404 \def\AtBegShi@PatchCrop{%
405   \begingroup
406     \def\AtBegShi@Crop@shipout{%
407       \afterassignment\CROP@ship
408       \setbox\@cclv=%
409     }%
410     \def\AtBegShi@Crop@ship{%
411       \ifvoid\@cclv
412         \expandafter\aftergroup
413       \fi
414       \CROP@@ship
415     }%
416     \def\AtBegShi@Crop@shiplist{%
417       \lineskip\z@
418       \lineskiplimit\z@
419       \baselineskip\z@
420       \CROP@kernel
421       \box\@cclv
422     }%
423     \def\AtBegShi@Crop@@ship{%
424       \CROP@shipout\vbox{%

```



```

425         \CROP@shiplist
426     }%
427 }%
428 \ifx\AtBegShi@Crop@ship\CROP@ship
429     \ifx\AtBegShi@Crop@shiplist\CROP@shiplist
430         \ifx\AtBegShi@Crop@@@ship\CROP@@@ship
431             \let\AtBegShi@found\relax
432             \ifx\shipout\AtBegShi@Crop@shipout
433                 \def\AtBegShi@found{\shipout}%
434             \else\ifx\AtBegShi@OrgShipout\AtBegShi@Crop@shipout
435                 \def\AtBegShi@found{\AtBegShi@OrgShipout}%
436             \else\ifx\@EveryShipout@Org@Shipout\AtBegShi@Crop@shipout
437                 \def\AtBegShi@found{\@EveryShipout@Org@Shipout}%
438             \else\ifx\GPTorg@shipout\AtBegShi@Crop@shipout
439                 \def\AtBegShi@found{\GPTorg@shipout}%
440             \else\ifx\THBorg@shipout\AtBegShi@Crop@shipout
441                 \def\AtBegShi@found{\THBorg@shipout}%
442             \else\ifx\mem@oldshipout\AtBegShi@Crop@shipout
443                 \def\AtBegShi@found{\mem@oldshipout}%
444             \fi\fi\fi\fi\fi\fi
445             \ifx\AtBegShi@found\relax
446             \else
447                 \expandafter\endgroup
448                 \expandafter\def\AtBegShi@found{%
449                     \edef\AtBegShi@GroupLevel{\number\currentgrouplevel}%
450                     \afterassignment\CROP@ship
451                     \setbox\AtBeginShipoutBox=%
452                 }%
453                 \def\CROP@ship{%
454                     \ifnum\AtBegShi@GroupLevel=\currentgrouplevel
455                     \else
456                         \expandafter\aftergroup
457                         \fi
458                     \CROP@@@ship
459                 }%
460                 \def\CROP@shiplist{%
461                     \lineskip\z@
462                     \lineskiplimit\z@
463                     \baselineskip\z@
464                     \CROP@kernel
465                     \box\AtBeginShipoutBox
466                 }%
467                 \def\CROP@@@ship{%
468                     \ifvoid\AtBeginShipoutBox
469                     \else
470                         \setbox\AtBeginShipoutBox=\vbox{%
471                             \CROP@shiplist
472                         }%
473                         \expandafter\CROP@shipout
474                         \expandafter\box
475                         \expandafter\AtBeginShipoutBox
476                     \fi
477                 }%
478                 \@PackageInfoNoLine{atbegshi}{Package 'crop' patched}%
479                 \begingroup
480             \fi
481         \fi
482     \fi
483 \fi
484 \endgroup
485 \let\AtBegShi@PatchCrop\relax
486 }

```

```

487 \@ifpackageloaded{crop}{%
488   \AtBegShi@PatchCrop
489 }{%
490   \AtBeginDocument{\AtBegShi@PatchCrop}%
491 }

```

3.5.2 Package everyshi

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

```

492 \def\AtBegShi@PatchEveryshi{%
493   \begingroup
494     \long\def\AtBegShi@Everyshi@shipout{%
495       \afterassignment\@EveryShipout@Test
496       \global\setbox\@cclv= %
497     }%
498     \long\def\AtBegShi@Everyshi@Test{%
499       \ifvoid\@cclv\relax
500         \aftergroup\@EveryShipout@Output
501       \else
502         \@EveryShipout@Output
503       \fi
504     }%
505     \ifx\AtBegShi@Everyshi@Test\@EveryShipout@Test
506       \let\AtBegShi@found\relax
507       \ifx\shipout\AtBegShi@Everyshi@shipout
508         \def\AtBegShi@found{\shipout}%
509       \else\ifx\AtBegShi@OrgShipout\AtBegShi@Everyshi@shipout
510         \def\AtBegShi@found{\AtBegShi@OrgShipout}%
511       \else\ifx\CROP@shipout\AtBegShi@Everyshi@shipout
512         \def\AtBegShi@found{\CROP@shipout}%
513       \else\ifx\GPTorg@shipout\AtBegShi@Everyshi@shipout
514         \def\AtBegShi@found{\GPTorg@shipout}%
515       \else\ifx\THBorg@shipout\AtBegShi@Everyshi@shipout
516         \def\AtBegShi@found{\THBorg@shipout}%
517       \else\ifx\mem@oldshipout\AtBegShi@Everyshi@shipout
518         \def\AtBegShi@found{\mem@oldshipout}%
519       \else
520         \expandafter\ifx\csname @EveryShipout@Org@Shipout\endcsname
521           \relax
522           \ifx\@EveryShipout@Shipout\AtBegShi@Everyshi@shipout
523             \def\AtBegShi@found{\@EveryShipout@Shipout}%
524           \fi
525         \fi
526       \fi\fi\fi\fi\fi\fi
527       \ifx\AtBegShi@found\relax
528         \else
529           \expandafter\endgroup
530           \expandafter\def\AtBegShi@found{%
531             \edef\AtBegShi@GroupLevel{\number\currentgrouplevel}%
532             \afterassignment\@EveryShipout@Test
533             \setbox\AtBeginShipoutBox=%
534           }%
535           \def\@EveryShipout@Test{%
536             \ifnum\AtBegShi@GroupLevel=\currentgrouplevel
537             \else
538               \expandafter\aftergroup
539             \fi
540             \AtBegShi@Everyshi@Output
541           }%
542           \def\AtBegShi@Everyshi@Output{%
543             \ifvoid\AtBeginShipoutBox
544             \else

```

```

545         \global\setbox\@cclv\box\AtBeginShipoutBox
546         \expandafter\@EveryShipout@Output
547     \fi
548 }%
549 \@PackageInfoNoLine{atbegshi}{Package 'everyshi' patched}%
550 \begingroup
551 \fi
552 \fi
553 \endgroup
554 \let\AtBegShi@PatchEveryshi\relax
555 }
556 \ifpackageloaded{everyshi}{%
557   \AtBegShi@PatchEveryshi
558 }{%
559   \AtBeginDocument{\AtBegShi@PatchEveryshi}%
560 }

```

3.5.3 Class memoir

Fix of method and box.

```

561 \def\AtBegShi@PatchMemoir{%
562   \begingroup
563     \def\AtBegShi@Memoir@shipout{%
564       \afterassignment\mem@shipi
565       \setbox\@cclv=%
566     }%
567     \def\AtBegShi@Memoir@shipi{%
568       \ifvoid\@cclv
569         \expandafter\aftergroup
570       \fi
571       \mem@shipii
572     }%
573     \def\AtBegShi@Memoir@shipiiA{%
574       \mem@oldshipout\vbox{%
575         \trimmarks
576         \unvbox\@cclv
577       }%
578     }%
579     \def\AtBegShi@Memoir@shipiiB{%
580       \ifvoid\@cclv
581         \mem@oldshipout\box\@cclv
582       \else
583         \mem@oldshipout\vbox{%
584           \trimmarks
585           \unvbox\@cclv
586         }%
587       \fi
588     }%
589     \ifx\AtBegShi@Memoir@shipi\mem@shipi
590       \ifcase\ifx\AtBegShi@Memoir@shipiiA\mem@shipii
591         \z@
592       \else
593         \ifx\AtBegShi@Memoir@shipiiB\mem@shipii
594           \z@
595         \else
596           \@ne
597         \fi
598       \fi
599       \let\AtBegShi@found\relax
600       \ifx\shipout\AtBegShi@Memoir@shipout
601         \def\AtBegShi@found{\shipout}%
602       \else\ifx\AtBegShi@OrgShipout\AtBegShi@Memoir@shipout

```

```

603     \def\AtBegShi@found{\AtBegShi@OrgShipout}%
604 \else\ifx\CROP@shipout\AtBegShi@Memoir@shipout
605     \def\AtBegShi@found{\CROP@shipout}%
606 \else\ifx\GPTorg@shipout\AtBegShi@Memoir@shipout
607     \def\AtBegShi@found{\GPTorg@shipout}%
608 \else\ifx\THBorg@shipout\AtBegShi@Memoir@shipout
609     \def\AtBegShi@found{\THBorg@shipout}%
610 \else\ifx@EveryShipout@Org@Shipout\AtBegShi@Memoir@shipout
611     \def\AtBegShi@found{@EveryShipout@Org@Shipout}%
612 \fi\fi\fi\fi\fi
613 \ifx\AtBegShi@found\relax
614 \else
615     \expandafter\endgroup
616     \expandafter\def\AtBegShi@found{%
617         \edef\AtBegShi@GroupLevel{\number\currentgrouplevel}%
618         \afterassignment\mem@shipi
619         \setbox\AtBeginShipoutBox=%
620     }%
621     \def\mem@shipi{%
622         \ifnum\AtBegShi@GroupLevel=\currentgrouplevel
623         \else
624             \expandafter\aftergroup
625             \fi
626         \mem@shipii
627     }%
628     \def\mem@shipii{%
629         \ifvoid\AtBeginShipoutBox
630         \else
631             \setbox\AtBeginShipoutBox=\vbox{%
632                 \trimmarks
633                 \ifvbox\AtBeginShipoutBox
634                     \unvbox\AtBeginShipoutBox
635                 \else
636                     \box\AtBeginShipoutBox
637                 \fi
638             }%
639             \expandafter\mem@oldshipout
640             \expandafter\box
641             \expandafter\AtBeginShipoutBox
642         \fi
643     }%
644     \@PackageInfoNoLine{atbegshi}{Class 'memoir' patched}%
645     \begingroup
646 \fi
647 \fi
648 \fi
649 \endgroup
650 \let\AtBegShi@PatchMemoir\relax
651 }
652 \@ifclassloaded{memoir}{%
653     \AtBegShi@PatchMemoir
654 }{%
655     \AtBeginDocument{\AtBegShi@PatchMemoir}%
656 }
657 \AtBegShi@AtEnd
658 \end{package}

```

4 Test

4.1 Catcode checks for loading

```

659 (*test1)

660 \catcode'\{=1 %
661 \catcode'\}=2 %
662 \catcode'\#=6 %
663 \catcode'\@=11 %
664 \expandafter\ifx\csname count@\endcsname\relax
665   \countdef\count@=255 %
666 \fi
667 \expandafter\ifx\csname @gobble\endcsname\relax
668   \long\def\@gobble#1{}%
669 \fi
670 \expandafter\ifx\csname @firstofone\endcsname\relax
671   \long\def\@firstofone#1{#1}%
672 \fi
673 \expandafter\ifx\csname loop\endcsname\relax
674   \expandafter\@firstofone
675 \else
676   \expandafter\@gobble
677 \fi
678 {%
679   \def\loop#1\repeat{%
680     \def\body{#1}%
681     \iterate
682   }%
683   \def\iterate{%
684     \body
685     \let\next\iterate
686   \else
687     \let\next\relax
688   \fi
689   \next
690 }%
691 \let\repeat=\fi
692 }%
693 \def\RestoreCatcodes{}
694 \count@=0 %
695 \loop
696   \edef\RestoreCatcodes{%
697     \RestoreCatcodes
698     \catcode\the\count@=\the\catcode\count@\relax
699   }%
700 \ifnum\count@<255 %
701   \advance\count@ 1 %
702 \repeat
703
704 \def\RangeCatcodeInvalid#1#2{%
705   \count@=#1\relax
706   \loop
707     \catcode\count@=15 %
708   \ifnum\count@<#2\relax
709     \advance\count@ 1 %
710   \repeat
711 }
712 \expandafter\ifx\csname LoadCommand\endcsname\relax
713   \def\LoadCommand{\input atbegshi.sty\relax}%
714 \fi
715 \def\Test{%
716   \RangeCatcodeInvalid{0}{47}%
717   \RangeCatcodeInvalid{58}{64}%
718   \RangeCatcodeInvalid{91}{96}%
719   \RangeCatcodeInvalid{123}{255}%
720   \catcode'\@=12 %

```

```

721 \catcode'\=0 %
722 \catcode'\{=1 %
723 \catcode'\}=2 %
724 \catcode'\#=6 %
725 \catcode'\[=12 %
726 \catcode'\]=12 %
727 \catcode'\%=14 %
728 \catcode'\ =10 %
729 \catcode13=5 %
730 \LoadCommand
731 \RestoreCatcodes
732 }
733 \Test
734 \csname @@end\endcsname
735 \end
736 </test1>

737 <*test2>
738 \input atbegshi.sty\relax
739 \def\msg#\{\immediate\write16}
740 \msg{File: atbegshi-test2.tex 2008/07/19 v1.8 Test file for plain-TeX}
741 \def\testmsg#1#2{%
742   \msg{}%
743   \msg{*** Test with box (#1), expected page output [#2]}% hash-ok
744 }
745
746 \newbox\voidbox
747 \def\void{\box\voidbox}
748 \begingroup
749   \setbox\voidbox=\void
750 \endgroup
751
752 \count0=0\relax
753 \AtBeginShipout{%
754   \global\advance\count0 by 1\relax
755   \msg{* Inside \string\AtBeginShipout: [\the\count0]}%
756 }
757
758 \AtBeginShipoutFirst{%
759   \msg{* Inside \string\AtBeginShipoutFirst}%
760   Hello World%
761 }
762
763 \testmsg{\string\null}{1}
764 \shipout\null
765
766 \AtBeginShipoutFirst{%
767   This is too late%
768 }
769
770 \testmsg{void}{1}
771 \shipout\void
772
773 \testmsg{\string\copy255 (not void)}{2}
774 \setbox255\hbox{\vrule height 10bp width 10bp}
775 \shipout\copy255 %
776
777 \testmsg{\string\copy255 (again)}{3}
778 \shipout\copy255 %
779
780 \testmsg{\string\box255}{4}
781 \shipout\box255 %
782

```

```

783 \testmsg{\string\box255 (again)}{}
784 \shipout\box255 %
785
786 \testmsg{\string\hbox}{5}
787 \shipout\hbox{\vrule height 5bp width 20bp}
788
789 \testmsg{\string\vbox}{6}
790 \shipout\vbox{\hrule height 20bp width 5bp}
791
792 \testmsg{\string\null, voided by hook}{}
793 \def\VoidBox{%
794   \begingroup
795     \setbox\AtBeginShipoutBox=\box\AtBeginShipoutBox
796   \endgroup
797 }
798 \AtBeginShipout{\VoidBox}
799 \shipout\null
800 \def\VoidBox{}
801
802 \msg{*** \string\begingroup}
803 \begingroup
804   \testmsg{void}{}%
805   \shipout(void
806 \msg{*** \string\endgroup}
807 \endgroup
808
809 \msg{*** \string\begingroup}
810 \begingroup
811   \testmsg{void}{}%
812   \shipout(void
813   \testmsg{\string\null}{8}%
814   \shipout\null
815 \msg{*** \string\endgroup}
816 \endgroup
817
818 \testmsg{output routine}{9}
819 Hello World
820 \vfill
821 \eject
822
823 \testmsg{\string\null\space(discarded)}{}
824 \AtBeginShipout{%
825   \msg{* Inside \string\AtBeginShipout: DISCARD}%
826   \AtBeginShipoutDiscard
827 }
828 \shipout\null
829
830 \end
831 /test2)
832 <*test3>
833 \NeedsTeXFormat{LaTeX2e}
834 \ProvidesFile{atbegshi-test3.tex}[2008/07/19 v1.8 Test file for LaTeX]
835 \RequirePackage{color}
836 \pagecolor{yellow}
837 \documentclass[a5paper,showtrims]{memoir}
838 \usepackage{atbegshi}
839 \AtBeginShipout{%
840   \setbox\AtBeginShipoutBox=\vbox{%
841     \vbox to Opt{%
842       \kern-1.5in %
843       \hbox to Opt{%
844         \kern-1.5in %

```

```

845         \color{blue}%
846         \rule{1in}{1in}%
847         \hss
848     }%
849     \vss
850 }%
851 \hrule
852 \hbox{\vrule\box\AtBeginShipoutBox\vrule}%
853 \hrule
854 }%
855 }
856 \usepackage{eso-pic}
857 \makeatletter
858 \@EveryShipout@Init
859 \let\@EveryShipout@Init\relax
860 \makeatother
861 \AddToShipoutPicture{%
862     \hspace{.52\paperwidth}%
863     \colorbox{cyan}{%
864         \rule{0mm}{\paperheight}%
865         \hspace{.48\paperwidth}%
866     }%
867 }

```

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

```

868 \expandafter\let\csname ver@crop.sty\endcsname\relax
869 \usepackage[color=red,cross,a4,center]{crop}
870 \begin{document}
871 \shipout\null
872 \shipout\box\csname voidb@x\endcsname
873 \section{Hello World}
874 \end{document}
875 </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
```

¹<ftp://ftp.ctan.org/tex-archive/>

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

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

5.3 Package installation

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

```
tex atbegshi.dtx
```

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

```
atbegshi.sty      → tex/generic/oberdiek/atbegshi.sty
atbegshi.pdf      → doc/latex/oberdiek/atbegshi.pdf
atbegshi-example.tex → doc/latex/oberdiek/atbegshi-example.tex
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 pdfL^AT_EX:

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

6 History

[2007/04/17 v1.0]

- First version.

[2007/04/18 v1.1]

- New method based on `\lastkern` is used if ϵ -T_EX is missing.
- `\AtBeginShipoutDiscard` also resets `\deadcycles`.

[2007/04/19 v1.2]

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

[2007/04/26 v1.3]

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

[2007/04/27 v1.4]

- Small optimizations.

[2007/06/06 v1.5]

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

[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.

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	
\#	662, 724
\%	727
\@	663, 720
\@EveryShipout@Init	858, 859
\@EveryShipout@Org@Shipout	436, 437, 610, 611
\@EveryShipout@Output	500, 502, 546
\@EveryShipout@Shipout	522, 523
\@EveryShipout@Test	495, 505, 532, 535
\@PackageInfoNoLine	202, 478, 549, 644
\@PackageWarning	193, 258, 282
\@ccclv	408, 411, 421, 496, 499, 545, 565, 568, 576, 580, 581, 585
\@empty	219, 220, 238
\@firstofone	671, 674
\@gobble	668, 676
\@ifclassloaded	652
\@ifdefinable	156
\@ifpackageloaded	487, 556
\@one	139, 142, 596
\@undefined	141
\[725
\]	721
\{	660, 722
\}	661, 723
\]	726
_	728
A	
\AddToShipoutPicture	861
\advance	701, 709, 754
\afterassignment	178, 407, 450, 495, 532, 564, 618
\aftergroup	185, 412, 456, 500, 538, 569, 624
\AtBeginDocument	302, 490, 559, 655
\AtBeginShipout	2, 6, 245, 753, 755, 798, 824, 825, 839
\AtBeginShipoutBox	175, 180, 192, 205, 211, 230, 235, 281, 292, 359, 364, 371, 375, 377, 383, 451, 465, 468, 470, 475, 533, 543, 545, 619, 629, 631, 633, 634, 636, 641, 795, 840, 852
\AtBeginShipoutDiscard	3, 29, 160, 286, 826
\AtBeginShipoutFirst	3, 239, 253, 260, 758, 759, 766
\AtBeginShipoutInit	3, 290, 300, 302
\AtBeginShipoutNext	2, 14, 28, 249
\AtBeginShipoutUpperLeft	3, 7, 15, 358
\AtBeginShipoutUpperLeftForeground	3, 374
\AtBegShi@AbortIfUndefined	393, 400, 401, 402, 403
\AtBegShi@AddHook	247, 251, 274
\AtBegShi@AddTo	255, 263, 275
\AtBegShi@AtEnd	113, 114, 396, 657
\AtBegShi@BeginPicture	329, 345, 365, 384
\AtBegShi@CheckDefinable	135, 160, 245, 249, 253, 290, 293
\AtBegShi@Crop@ship	423, 430
\AtBegShi@Crop@ship	410, 428
\AtBegShi@Crop@shiplist	416, 429
\AtBegShi@Crop@shipout	406, 432, 434, 436, 438, 440, 442
\AtBegShi@Discardedfalse	197, 203
\AtBegShi@Discardedtrue	163
\AtBegShi@EndPicture	339, 350, 367, 386
\AtBegShi@Everyshi@Output	540, 542
\AtBegShi@Everyshi@shipout	494, 507, 509, 511, 513, 515, 517, 522
\AtBegShi@Everyshi@Test	498, 505
\AtBegShi@First	209, 217
\AtBegShi@FirstDisabled	239, 257
\AtBegShi@found	431, 433, 435, 437, 439, 441, 443, 445, 448, 506, 508, 510, 512, 514, 516, 518, 523, 527, 530, 599, 601, 603, 605, 607, 609, 611, 613, 616
\AtBegShi@GroupLevel	177, 184, 449, 454, 531, 536, 617, 622
\AtBegShi@Hook	198, 242, 247
\AtBegShi@HookFirst	220, 224, 244, 255
\AtBegShi@HookNext	199, 200, 243, 251
\AtBegShi@horigin	311, 314, 361, 380
\AtBegShi@Item	275, 277
\AtBegShi@Memoir@shipi	567, 589
\AtBegShi@Memoir@shipiiA	573, 590
\AtBegShi@Memoir@shipiiB	579, 593
\AtBegShi@Memoir@shipout	563, 600, 602, 604, 606, 608, 610
\AtBegShi@OrgProtect	195, 207, 210
\AtBegShi@OrgShipout	211, 293, 294, 434, 435, 509, 510, 602, 603
\AtBegShi@Output	187, 189
\AtBegShi@PatchCrop	404, 485, 488, 490
\AtBegShi@PatchEveryshi	492, 554, 557, 559
\AtBegShi@PatchMemoir	561, 650, 653, 655
\AtBegShi@Shipout	173, 295
\AtBegShi@Test	178, 182
\AtBegShi@vorigin	312, 315, 363, 382
B	
\baselineskip	231, 419, 463
\begin	11, 870

\body	680, 684	H	
\box	205, 211, 371, 421, 465, 474, 545, 581, 636, 640, 747, 780, 781, 783, 784, 795, 852, 872	\hbox	175, 346, 359, 375, 774, 786, 787, 843, 852
C		\hrule	790, 851, 853
\catcode	38, 39, 40, 41, 42, 53, 54, 55, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 97, 98, 101, 102, 103, 104, 108, 109, 110, 111, 115, 117, 167, 168, 170, 171, 215, 216, 660, 661, 662, 663, 698, 707, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729	\hspace	862, 865
\circle	8	\hss	390, 847
\color	16, 845	\ht	228, 353, 364, 383
\colorbox	863	I	
\copy	234, 235, 355, 377, 773, 775, 777, 778	\ifAtBegShi@Discarded	159, 201, 278
\count	752, 754, 755	\ifcase	44, 138, 318, 590
\count@	665, 694, 698, 700, 701, 705, 707, 708, 709	\ifdim	183
\countdef	665	\ifnum	184, 454, 536, 622, 700, 708
\CROP@ship	414, 430, 458, 467	\ifpdf	310
\CROP@kernel	420, 464	\ifvbox	633
\CROP@ship	407, 428, 450, 453	\ifvoid	192, 281, 411, 468, 499, 543, 568, 580, 629
\CROP@shiplist	425, 429, 460, 471	\ifx	45, 47, 56, 80, 88, 130, 136, 138, 141, 166, 220, 264, 299, 305, 319, 322, 333, 395, 428, 429, 430, 432, 434, 436, 438, 440, 442, 445, 505, 507, 509, 511, 513, 515, 517, 520, 522, 527, 589, 590, 593, 600, 602, 604, 606, 608, 610, 613, 664, 667, 670, 673, 712
\CROP@shipout	424, 473, 511, 512, 604, 605	\ignorespaces	337, 348
\csname	43, 56, 80, 93, 100, 130, 136, 166, 196, 264, 292, 299, 305, 319, 322, 333, 395, 520, 664, 667, 670, 673, 712, 734, 868, 872	\immediate	58, 82, 739
\currentgrouplevel	177, 184, 449, 454, 531, 536, 617, 622	\input	131, 306, 713, 738
D		\iterate	681, 683, 685
\deadcycles	162	K	
\documentclass	2, 837	\kern	176, 361, 363, 364, 380, 382, 383, 842, 844
\dp	229, 354	L	
E		\lastkern	183
\E	216	\line	17, 18
\eject	821	\lineskip	232, 417, 461
\empty	47	\lineskiplimit	233, 418, 462
\end	34, 735, 830, 874	\LoadCommand	713, 730
\endcsname	43, 56, 80, 93, 100, 130, 136, 166, 196, 264, 292, 299, 305, 319, 322, 333, 395, 520, 664, 667, 670, 673, 712, 734, 868, 872	\loop	679, 695, 706
\endinput	65, 397	M	
\endpicture	340	\makeatletter	857
\errmessage	147	\makeatother	860
F		\mem@oldshipout	442, 443, 517, 518, 574, 581, 583, 639
\fill	25	\mem@shipi	564, 589, 618, 621
G		\mem@shipii	571, 590, 593, 626, 628
\g@addto@macro	272	\MessageBreak	259, 283
\gdef	200, 242, 243, 244, 296	\msg	739, 740, 742, 743, 755, 759, 802, 806, 809, 815, 825
\GPTorg@shipout	438, 439, 513, 514, 606, 607	N	
		\NeedsTeXFormat	833
		\newbox	746
		\newif	159
		\newpage	13, 22, 27, 32
		\next	685, 687, 689
		\null	763, 764, 792, 799, 813, 814, 823, 828, 871
		\number	177, 449, 531, 617

P		\testmsg 741, 763, 770, 773, 777, 780, 783, 786, 789, 792, 804, 811, 813, 818, 823	
\p@	176	\THBorg@shipout	440, 441, 515, 516, 608, 609
\PackageInfo	61	\the 101, 102, 103, 104, 115, 268, 698, 755	
\pagecolor	836	\TMP@EnsureCode 112, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128	
\paperheight	8, 17, 18, 864	\toks	267, 268
\paperwidth	8, 17, 18, 862, 865	\trimmarks	575, 584, 632
\par	24	U	
\pdfhorigin	311	\unitlength	335
\pdfvorigin	312	\unvbox	576, 585, 634
\picture	331	\usepackage	3, 4, 5, 838, 856, 869
\protect	195, 207, 210	V	
\ProvidesFile	834	\vbox	222, 230, 362, 381, 424, 470, 574, 583, 631, 789, 790, 840, 841
\ProvidesPackage	94	\vfill	820
\put	8, 17, 18	\void	747, 749, 771, 805, 812
R		\VoidBox	793, 798, 800
\RangeCatcodeInvalid	704, 716, 717, 718, 719	\voidbox	746, 747, 749
\repeat	679, 691, 702, 710	\vrule	774, 787, 852
\RequirePackage	133, 308, 835	\vspace	25
\RestoreCatcodes	693, 696, 697, 731	\vss	368, 387, 849
\rlap	360, 376, 379	W	
\rule	846, 864	\wd	227, 375
S		\write	58, 82, 739
\section	12, 873	X	
\setbox	175, 180, 205, 222, 230, 346, 359, 375, 408, 451, 470, 496, 533, 545, 565, 619, 631, 749, 774, 795, 840	\X	215
\shipout	294, 295, 432, 433, 507, 508, 600, 601, 764, 771, 775, 778, 781, 784, 787, 790, 799, 805, 812, 814, 828, 871, 872	\x 43, 45, 47, 57, 61, 63, 81, 86, 93, 99, 107	
\space	148, 823	Z	
T		\z@	144, 162, 183, 222, 227, 228, 229, 233, 234, 267, 268, 417, 418, 419, 461, 462, 463, 591, 594
\Test	715, 733	\z@skip	231, 232