The atbegshi package

Heiko Oberdiek*
<heiko.oberdiek at googlemail.com>

2016/06/09 v1.18

Abstract

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

Contents

1 Documentation **2**
1.1 Examples . . . 4
1.1.1 Example: circle in background . . . 4
1.1.2 Example: adding TrimBox for dvipdfmx . . . 5

2 Method of \shipout overloading **6**
2.1 \shipout . . . 6
2.2 \afterassignment . . . 6
2.3 Test for direct or indirect boxes . . . 7
2.3.1 With ε-TEX . . . 7
2.3.2 Without ε-TEX . . . 7
2.3.3 \lastkern method . . . 8
2.4 Output . . . 9
2.5 Separate box register . . . 9
2.6 Summary . . . 9
2.6.1 With ε-TEX . . . 9
2.6.2 Without ε-TEX, traditional way . . . 10
2.6.3 \lastkern method . . . 10

3 Implementation **11**
3.1 Reload check and package identification . . . 11
3.2 Catcodes . . . 12
3.3 Preparations . . . 13
3.4 Additions to the shipout box . . . 17
3.5 Positioning . . . 19
3.6 Patches . . . 20
3.6.1 Package crop . . . 20
3.6.2 Package everyshi . . . 22
3.6.3 Class memoir . . . 23

4 Test **26**
4.1 Catcode checks for loading . . . 26

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

5 Installation **30**
5.1 Download . . . 30
5.2 Bundle installation . . . 30
5.3 Package installation . . . 30
5.4 Refresh file name databases . . . 30
5.5 Some details for the interested . . . 31

6 Catalogue **31**

7 History **32**
[2007/04/17 v1.0] . . . 32
[2007/04/18 v1.1] . . . 32
[2007/04/19 v1.2] . . . 32
[2007/04/26 v1.3] . . . 32
[2007/04/27 v1.4] . . . 32
[2007/06/06 v1.5] . . . 32
[2007/09/09 v1.6] . . . 32
[2008/07/18 v1.7] . . . 32
[2008/07/19 v1.8] . . . 32
[2008/07/31 v1.9] . . . 32
[2009/12/02 v1.10] . . . 32
[2010/03/01 v1.11] . . . 33
[2010/03/25 v1.12] . . . 33
[2010/08/18 v1.13] . . . 33
[2010/12/02 v1.14] . . . 33
[2011/01/30 v1.15] . . . 33
[2011/10/05 v1.16] . . . 33
[2016/05/16 v1.17] . . . 33
[2016/06/09 v1.18] . . . 33

1 Documentation

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

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

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

\AtBeginShipout {⟨*code*⟩}
\AtBeginShipoutBox

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

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

If LaTeX calls \shipout in \@outputpage (part of its output routine), the meaning of \protect is \noexpand. LaTeX sets \protect to the appropriate \@typeset@protect in the box that is shipped out. This is too late for the hooks, they are called earlier in the redefined \shipout. Therefore package atbegshi sets \protect to \@typeset@protect before it calls the hooks. (In \EveryShipout of package everyshi the user is responsible for the correct setting of \protect.)

\AtBeginShipoutNext {⟨*code*⟩}

This reimplements package everyshi's \AtNextShipout. The ⟨*code*⟩ is executed at shipout time of the next page only. It is just a convenience macro, it can be easily replaced by something like:

```
\newcommand{\MyShipoutHook}{}%
\AtBeginShipout{\MyShipoutHook}
\gdef\MyShipoutHook{%
  ... do something with next page ...
  \gdef\MyShipoutHook{}%
}
```

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

\AtBeginShipoutFirst {⟨*code*⟩}

This reimplements LaTeX's \AtBeginDvi. This hook is usually used for \special commands that include PostScript header files. The \code is directly executed in a \vbox that is put at the beginning of the output page. Dealing with the output box \AtBeginShipoutBox is not necessary and not permitted here.

\AtBeginShipoutDiscard

This macro notifies package atbegshi that the output page is discarded. The remaining hook code and the remaining hooks are not executed and the page is thrown away. Also \deadcycles is cleared to zero like an ordinary \shipout would do.

\AtBeginShipoutInit

Usually the redefinition of \shipout is delayed by \AtBeginDocument (if this macro exists). This can be too late, if other packages also redefines \shipout and the order does matter. \AtBeginShipoutInit forces the immediate redefinition of \shipout.

\AtBeginShipoutAddToBox {⟨*stuff*⟩}
\AtBeginShipoutAddToBoxForeground {⟨*stuff*⟩}

A quite common use case is the addition of \special or other whatsits to the page output box. Macro \AtBeginShipoutAddToBox puts ⟨*stuff*⟩ in a box with zeroed dimensions. The box with the ⟨*stuff*⟩ is put in the upper left corner of the shipout box \AtBeginShipoutBox. Macro \AtBeginShipoutAddToBox puts the ⟨*stuff*⟩ in the background, the other macro \AtBeginShipoutAddToBoxForeground in the foreground after the original shipout box contents is set.

A void shipout box (that means a discarded page) remains void that means ⟨*stuff*⟩ is ignored in this case. The box type of \AtBeginShipoutBox is preserved. Also the box nesting level for the original contents of \AtBeginShipoutBox remains, for example, to avoid trouble with links across pages in case of pdfTeX.

`\AtBeginShipoutUpperLeft {`⟨*background material*⟩`}`

This is a macro that puts material in the background of box `\AtBeginShipoutBox`. The ⟨*background material*⟩ is set in an `\hbox`, the reference point is the upper left corner of the output page. In case of pdfTEX in PDF mode, the settings of `\pdfhorigin` and `\pdfvorigin` are respected.

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

For LATEX users the ⟨*background material*⟩ is set inside a `picture` environment:

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

`\AtBeginShipoutUpperLeftForeground {`⟨*foreground material*⟩`}`

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

`\AtBeginShipoutOriginalShipout` ⟨*box*⟩

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

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

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

1.1 Examples

1.1.1 Example: circle in background

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

```
⟨*example1⟩
\documentclass[a4paper]{article}
\usepackage{color}
\usepackage{atbegshi}
```

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

```
\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.

```
\AtBeginShipout{%
  \AtBeginShipoutUpperLeft{%
    \put(0.5\paperwidth,-0.5\paperheight){\circle{10}}%
```

```
  }%
}
\begin{document}
\section{Hello World}
\newpage
\AtBeginShipoutNext{%
  \AtBeginShipoutUpperLeft{%
    \color{red}%
    \put(0,-0.5\paperheight){\line(1,0){\paperwidth}}%
    \put(0.5\paperwidth, 0){\line(0,-1){\paperheight}}%
  }%
}
Only on this page we add a red cross.
\newpage
This page has the circle only.
\par
\vspace{\fill}
The next page will be discarded.
\newpage
\AtBeginShipoutNext{%
  \AtBeginShipoutDiscard
}
This page is discarded.
\newpage
The last page.
\end{document}
```

⟨/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.

```
⟨*example2⟩
\documentclass{minimal}
\usepackage{atbegshi}
\usepackage[
  dvipdfm,
  paperwidth=630bp,
  paperheight=810bp
]{geometry}
\AtBeginShipout{%
  \setbox\AtBeginShipoutBox=\hbox{%
    \special{pdf: put @thispage <</TrimBox[9 9 621 801]>>}%
    \box\AtBeginShipoutBox
  }%
}
\begin{document}
  First page
  \newpage
  Second page
\end{document}
⟨/example2⟩
```

Remember, in `\AtBeginShipoutBoxFirst` the `\setbox` wrapper code is implicitly given and the `\special` is used directly.

2 Method of \shipout overloading

2.1 \shipout

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

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

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

Note: \box also clears the box register globally.

Then we have to differentiate between void and empty boxes:

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

Empty: A box with empty contents, such as \hbox{} (= \null) or \vbox{} is an *empty* \hbox or *empty* \vbox. If a box register holds such a box, the box still exists, therefore the box register is *not void*.

2.2 \afterassignment

We want to overload \shipout to do something with the box. It is quite impossible to do this reliable by catching the box using macro arguments. The variety of box specifications is too large, Examples:

```
\shipout\null
\shipout\vbox{...}
\shipout\vtop\bgroup ...\egroup
\shipout\box255
```

Even worse, the braces don't need to be balanced:

```
\shipout\hbox\bgroup}
\shipout\vbox{\egroup
```

Happily TeX provides a reliable way via \afterassignment. It takes a macro name and executes it just after the assignment.

Now we can redefine \shipout. The box specification that follows \shipout is catched by \setbox. This is an assignment to a box register. \afterassignment notifies TEX, that we want to call \@test right after the assignment:

```
\shipout :=
 \afterassignment\@test
 \setbox\mybox=
```

We have seen different box specifications. Indirect boxes are easy to understand:

\shipout\box0 ⇒ \setbox\mybox=\box0 \@test

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

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

2.3 Test for direct or indirect boxes

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

2.3.1 With ε-TEX

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

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

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

2.3.2 Without ε-TEX

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

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

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

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

Nevertheless we have two alternatives for an initial value:

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

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

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

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

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

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

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

2.3.3 `\lastkern` method

At the beginning of a new box, there is no `\kern`, the contents of the box is still empty and `\lastkern` returns 0 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.

```
⟨*package⟩
```

3.1 Reload check and package identification

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

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \catcode35=6 % #
  \catcode39=12 % '
  \catcode44=12 % ,
  \catcode45=12 % -
  \catcode46=12 % .
  \catcode58=12 % :
  \catcode64=11 % @
  \catcode123=1 % {
  \catcode125=2 % }
  \expandafter\let\expandafter\x\csname ver@atbegshi.sty\endcsname
  \ifx\x\relax % plain-TeX, first loading
  \else
    \def\empty{}%
    \ifx\x\empty % LaTeX, first loading,
      % variable is initialized, but \ProvidesPackage not yet seen
    \else
      \expandafter\ifx\csname PackageInfo\endcsname\relax
        \def\x#1#2{%
          \immediate\write-1{Package #1 Info: #2.}%
        }%
      \else
        \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
      \fi
      \x{atbegshi}{The package is already loaded}%
      \aftergroup\endinput
    \fi
  \fi
\endgroup%
```

Package identification:

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \catcode35=6 % #
  \catcode39=12 % '
  \catcode40=12 % (
```

```
\catcode41=12 % )
\catcode44=12 % ,
\catcode45=12 % -
\catcode46=12 % .
\catcode47=12 % /
\catcode58=12 % :
\catcode64=11 % @
\catcode91=12 % [
\catcode93=12 % ]
\catcode123=1 % {
\catcode125=2 % }
\expandafter\ifx\csname ProvidesPackage\endcsname\relax
  \def\x#1#2#3[#4]{\endgroup
    \immediate\write-1{Package: #3 #4}%
    \xdef#1{#4}%
  }%
\else
  \def\x#1#2[#3]{\endgroup
    #2[{#3}]%
    \ifx#1\@undefined
      \xdef#1{#3}%
    \fi
    \ifx#1\relax
      \xdef#1{#3}%
    \fi
  }%
\fi
\expandafter\x\csname ver@atbegshi.sty\endcsname
\ProvidesPackage{atbegshi}%
 [2016/06/09 v1.18 At begin shipout hook (HO)]%
```

3.2 Catcodes

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \catcode123=1 % {
  \catcode125=2 % }
  \catcode64=11 % @
  \def\x{\endgroup
    \expandafter\edef\csname AtBegShi@AtEnd\endcsname{%
      \endlinechar=\the\endlinechar\relax
      \catcode13=\the\catcode13\relax
      \catcode32=\the\catcode32\relax
      \catcode35=\the\catcode35\relax
      \catcode61=\the\catcode61\relax
      \catcode64=\the\catcode64\relax
      \catcode123=\the\catcode123\relax
      \catcode125=\the\catcode125\relax
    }%
  }%
\x\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 % ^^M
\endlinechar=13 %
\catcode35=6 % #
\catcode64=11 % @
\catcode123=1 % {
\catcode125=2 % }
\def\TMP@EnsureCode#1#2{%
  \edef\AtBegShi@AtEnd{%
    \AtBegShi@AtEnd
    \catcode#1=\the\catcode#1\relax
```

```
  }%
  \catcode#1=#2\relax
}
\TMP@EnsureCode{40}{12}% (
\TMP@EnsureCode{41}{12}% )
\TMP@EnsureCode{44}{12}% ,
\TMP@EnsureCode{45}{12}% -
\TMP@EnsureCode{47}{12}% /
\TMP@EnsureCode{46}{12}% .
\TMP@EnsureCode{58}{12}% :
\TMP@EnsureCode{91}{12}% [
\TMP@EnsureCode{93}{12}% ]
\TMP@EnsureCode{94}{7}% ^(superscript)
\TMP@EnsureCode{96}{12}% `
\edef\AtBegShi@AtEnd{\AtBegShi@AtEnd\noexpand\endinput}
```

3.3 Preparations

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname RequirePackage\endcsname\relax
  \def\TMP@RequirePackage#1[#2]{%
    \begingroup\expandafter\expandafter\expandafter\endgroup
    \expandafter\ifx\csname ver@#1.sty\endcsname\relax
      \input #1.sty\relax
    \fi
  }%
  \TMP@RequirePackage{infwarerr}[2007/09/09]%
  \TMP@RequirePackage{ltxcmds}[2010/03/01]%
\else
  \RequirePackage{infwarerr}[2007/09/09]%
  \RequirePackage{ltxcmds}[2010/03/01]%
\fi
```

\AtBegShi@CheckDefinable

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname @ifdefinable\endcsname\relax
  \def\AtBegShi@CheckDefinable#1{%
    \ifcase\ifx#1\relax
            \ltx@one
          \else
            \ifx#1\@undefined
              \ltx@one
            \else
              \ltx@zero
            \fi
          \fi
      \@PackageError{atbegshi}{%
        \string#1\space is already defined%
      }\@ehd
    \fi
  }%
\else
  \def\AtBegShi@CheckDefinable#1{%
    \@ifdefinable{#1}{}%
  }%
\fi
```

\ifAtBegShi@Discarded

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

\AtBeginShipoutDiscard

```
\AtBegShi@CheckDefinable\AtBeginShipoutDiscard
\def\AtBeginShipoutDiscard{%
```

```
  \deadcycles=\ltx@zero
  \global\AtBegShi@Discardedtrue
}

\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname currentgrouplevel\endcsname\relax
  \catcode`X=9  % ignore
  \catcode`E=14 % comment
\else
  \catcode`X=14 % comment
  \catcode`E=9  % ignore
\fi
```

\AtBegShi@Shipout

```
\def\AtBegShi@Shipout{%
X \begingroup
X \setbox\AtBeginShipoutBox=\hbox\bgroup
X \kern\p@
E \edef\AtBegShi@GroupLevel{\number\currentgrouplevel}%
  \afterassignment\AtBegShi@Test
X \global
  \setbox\AtBeginShipoutBox=%
}
```

\AtBegShi@Test

```
\def\AtBegShi@Test{%
X \ifdim\lastkern=0pt %
E \ifnum\AtBegShi@GroupLevel<\currentgrouplevel
    \expandafter\aftergroup
  \fi
  \AtBegShi@Output
}
```

\AtBegShi@Output

```
\def\AtBegShi@Output{%
X \egroup
X \endgroup
  \ifvoid\AtBeginShipoutBox
    \@PackageWarning{atbegshi}{Ignoring void shipout box}%
  \else
    \let\AtBegShi@OrgProtect\protect
    \csname set@typeset@protect\endcsname
    \global\AtBegShi@Discardedfalse
    \AtBegShi@Hook
    \expandafter\gdef\expandafter\AtBegShi@HookNext
    \expandafter{\expandafter}%
    \AtBegShi@HookNext
    \ifAtBegShi@Discarded
      \@PackageInfoNoLine{atbegshi}{Shipout page discarded}%
      \global\AtBegShi@Discardedfalse
      \begingroup
        \setbox\AtBeginShipoutBox\box\AtBeginShipoutBox
      \endgroup
      \let\protect\AtBegShi@OrgProtect
    \else
      \AtBegShi@First
      \let\protect\AtBegShi@OrgProtect
      \AtBegShi@GetBoxSize\AtBeginShipoutBox
      \ltx@ifundefined{AtNextShipout}{%
      }{%
        \AtNextShipout{\AtBegShi@GetBoxSize\@cclv}%
      }%
      \AtBeginShipoutOriginalShipout\box\AtBeginShipoutBox
```

```
    \fi
  \fi
}
```

\AtBegShi@GetBoxSize

```
\def\AtBegShi@GetBoxSize#1{%
  \xdef\AtBeginShipoutBoxWidth{\the\wd#1}%
  \xdef\AtBeginShipoutBoxHeight{\the\ht#1}%
  \xdef\AtBeginShipoutBoxDepth{\the\dp#1}%
}
```

\AtBeginShipoutBoxWidth

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

\AtBeginShipoutBoxHeight

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

\AtBeginShipoutBoxDepth

```
\def\AtBeginShipoutBoxDepth{0pt}

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

\AtBegShi@First

```
\def\AtBegShi@First{%
  \ifx\AtBegShi@HookFirst\ltx@empty
  \else
    \AtBeginShipoutAddToBox{\AtBegShi@HookFirst}%
  \fi
  \global\let\AtBegShi@First\ltx@empty
  \global\let\AtBeginShipoutFirst\AtBegShi@FirstDisabled
}
```

\AtBegShi@Hook

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

\AtBegShi@HookNext

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

\AtBegShi@HookFirst

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

\AtBeginShipout

```
\AtBegShi@CheckDefinable\AtBeginShipout
\def\AtBeginShipout{%
  \AtBegShi@AddHook\AtBegShi@Hook
}
```

\AtBeginShipoutNext

```
\AtBegShi@CheckDefinable\AtBeginShipoutNext
\def\AtBeginShipoutNext{%
  \AtBegShi@AddHook\AtBegShi@HookNext
}
```

\AtBeginShipoutFirst

```
\AtBegShi@CheckDefinable\AtBeginShipoutFirst
\def\AtBeginShipoutFirst{%
  \AtBegShi@AddTo\AtBegShi@HookFirst
}
```

\AtBegShi@FirstDisabled

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

\AtBegShi@AddTo

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname g@addto@macro\endcsname\relax
  \long\def\AtBegShi@AddTo#1#2{%
    \begingroup
      \toks\ltx@zero\expandafter{#1#2}%
      \xdef#1{\the\toks\ltx@zero}%
    \endgroup
  }%
\else
  \let\AtBegShi@AddTo\g@addto@macro
\fi
```

\AtBegShi@AddHook

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

\AtBegShi@Item

```
\long\def\AtBegShi@Item#1{%
  \ifAtBegShi@Discarded
  \else
    #1%
    \ifAtBegShi@Discarded
    \else
      \ifvoid\AtBeginShipoutBox
        \@PackageWarning{atbegshi}{%
          Shipout box was voided by hook,\MessageBreak
          ignoring shipout box%
        }%
        \AtBeginShipoutDiscard
      \fi
    \fi
  \fi
}
```

\AtBeginShipoutInit

```
\AtBegShi@CheckDefinable\AtBeginShipoutInit
\def\AtBeginShipoutInit{%
  \ltx@IfUndefined{newbox}{%
    \@PackageError{atbegshi}{%
      \string\AtBeginShipoutInit\space failed\MessageBreak
       because of missing \expandafter\string\csname newbox\endcsname
    }\@ehc
  }{%
    \csname newbox\endcsname\AtBeginShipoutBox
    \AtBegShi@CheckDefinable\AtBeginShipoutOriginalShipout
    \global\let\AtBeginShipoutOriginalShipout\shipout
    \global\let\shipout\AtBegShi@Shipout
  }%
  \gdef\AtBeginShipoutInit{}%
}

\begingroup\expandafter\expandafter\expandafter\endgroup
```

```
\expandafter\ifx\csname AtBeginDocument\endcsname\relax
  \AtBeginShipoutInit
\else
  \AtBeginDocument{\AtBeginShipoutInit}%
\fi
```

3.4 Additions to the shipout box

\AtBeginShipoutAddToBox

```
\def\AtBeginShipoutAddToBox#1{%
  \ifhbox\AtBeginShipoutBox
    \edef\AtBegShi@restore{%
      \hfuzz=\the\hfuzz\relax
      \hbadness=\the\hbadness\relax
    }%
    \hfuzz=1073741823sp\relax
    \hbadness=2147483647\relax
    \setbox\AtBeginShipoutBox=\hbox to \wd\AtBeginShipoutBox{%
      \setbox\ltx@zero=\hbox{%
        \begingroup
          \AtBegShi@restore
          #1%
        \endgroup
      }%
      \wd\ltx@zero=0pt\relax
      \ht\ltx@zero=0pt\relax
      \dp\ltx@zero=0pt\relax
      \raise\ht\AtBeginShipoutBox\copy\ltx@zero
      \unhcopy\AtBeginShipoutBox
    }%
    \AtBegShi@restore
  \else
    \ifvbox\AtBeginShipoutBox
      \edef\AtBegShi@restore{%
        \vfuzz=\the\vfuzz\relax
        \vbadness=\the\vbadness\relax
        \dimen\ltx@zero=\the\dimen\ltx@zero\relax
      }%
      \edef\AtBegShi@restorebox{%
        \ht\AtBeginShipoutBox=\the\ht\AtBeginShipoutBox\relax
        \dp\AtBeginShipoutBox=\the\dp\AtBeginShipoutBox\relax
      }%
      \vfuzz=1073741823sp\relax
      \vbadness=2147483647\relax
      \dimen\ltx@zero=\ht\AtBeginShipoutBox
      \advance\dimen\ltx@zero by \dp\AtBeginShipoutBox
      \setbox\AtBeginShipoutBox=\vbox to \dimen\ltx@zero{%
        \setbox\ltx@zero=\hbox{%
          \begingroup
            \AtBegShi@restore
            #1%
          \endgroup
        }%
        \wd\ltx@zero=0pt\relax
        \ht\ltx@zero=0pt\relax
        \dp\ltx@zero=0pt\relax
        \baselineskip=0pt\relax
        \lineskip=0pt\relax
        \lineskiplimit=0pt\relax
        \copy\ltx@zero
        \unvbox\AtBeginShipoutBox
        \kern0pt%
```

```
    }%
    \AtBegShi@restore
    \AtBegShi@restorebox
  \fi
 \fi
}
```

eginShipoutAddToBoxForeground

```
\def\AtBeginShipoutAddToBoxForeground#1{%
 \ifhbox\AtBeginShipoutBox
  \edef\AtBegShi@restore{%
   \hfuzz=\the\hfuzz\relax
   \hbadness=\the\hbadness\relax
  }%
  \hfuzz=1073741823sp\relax
  \hbadness=2147483647\relax
  \setbox\AtBeginShipoutBox=\hbox to \wd\AtBeginShipoutBox{%
   \unhcopy\AtBeginShipoutBox
   \kern-\wd\AtBeginShipoutBox
   \setbox\ltx@zero=\hbox{%
    \begingroup
     \AtBegShi@restore
     #1%
    \endgroup
   }%
   \wd\ltx@zero=0pt\relax
   \ht\ltx@zero=0pt\relax
   \dp\ltx@zero=0pt\relax
   \raise\ht\AtBeginShipoutBox\copy\ltx@zero
   \kern\wd\AtBeginShipoutBox
  }%
  \AtBegShi@restore
 \else
  \ifvbox\AtBeginShipoutBox
   \edef\AtBegShi@restore{%
    \vfuzz=\the\vfuzz\relax
    \vbadness=\the\vbadness\relax
    \dimen\ltx@zero=\the\dimen\ltx@zero\relax
   }%
   \edef\AtBegShi@restorebox{%
    \ht\AtBeginShipoutBox=\the\ht\AtBeginShipoutBox\relax
    \dp\AtBeginShipoutBox=\the\dp\AtBeginShipoutBox\relax
   }%
   \vfuzz=1073741823sp\relax
   \vbadness=2147483647\relax
   \dimen\ltx@zero=\ht\AtBeginShipoutBox
   \advance\dimen\ltx@zero by \dp\AtBeginShipoutBox
   \setbox\AtBeginShipoutBox=\vbox to \dimen\ltx@zero{%
    \setbox\ltx@zero=\hbox{%
     \begingroup
      \AtBegShi@restore
      #1%
     \endgroup
    }%
    \wd\ltx@zero=0pt\relax
    \ht\ltx@zero=0pt\relax
    \dp\ltx@zero=0pt\relax
    \baselineskip=0pt\relax
    \lineskip=0pt\relax
    \lineskiplimit=0pt\relax
    \unvbox\AtBeginShipoutBox
    \kern-\dimen\ltx@zero
    \copy\ltx@zero
```

```
      \kern\dimen\ltx@zero
    }%
    \AtBegShi@restore
    \AtBegShi@restorebox
  \fi
 \fi
}
```

3.5 Positioning

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname RequirePackage\endcsname\relax
 \def\TMP@RequirePackage#1[#2]{%
  \begingroup\expandafter\expandafter\expandafter\endgroup
  \expandafter\ifx\csname ver@#1.sty\endcsname\relax
    \input #1.sty\relax
  \fi
 }%
 \TMP@RequirePackage{ifpdf}[2011/01/30]%
\else
 \RequirePackage{ifpdf}[2011/01/30]%
\fi
```

```
\ifpdf
 \def\AtBegShi@horigin{%
  \ifx\pdfhorigin\@undefined\pdfvariable horigin\else\pdfhorigin\fi}%
 \def\AtBegShi@vorigin{%
  \ifx\pdfvorigin\@undefined\pdfvariable vorigin\else\pdfvorigin\fi}%
\else
 \def\AtBegShi@horigin{72.27pt}%
 \def\AtBegShi@vorigin{72.27pt}%
\fi
```

```
\begingroup
\ifcase
  \expandafter\ifx\csname picture\endcsname\relax
    1%
  \else
    \expandafter\ifx\csname endpicture\endcsname\relax
      1%
    \else
      0%
    \fi
  \fi
 \endgroup
 \def\AtBegShi@BeginPicture{%
  \begingroup
  \picture(0,0)\relax
  \begingroup\expandafter\expandafter\expandafter\endgroup
  \expandafter\ifx\csname unitlength\endcsname\relax
  \else
    \unitlength=1pt\relax
  \fi
  \ignorespaces
 }%
 \def\AtBegShi@EndPicture{%
  \endpicture
  \endgroup
 }%
\else
 \endgroup
 \def\AtBegShi@BeginPicture{%
  \setbox\ltx@zero=\hbox\bgroup
  \begingroup
```

```
    \ignorespaces
  }%
  \def\AtBegShi@EndPicture{%
    \endgroup
    \egroup
    \ht\ltx@zero=0pt\relax
    \dp\ltx@zero=0pt\relax
    \copy\ltx@zero
  }%
\fi
```

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

```
\def\AtBeginShipoutUpperLeft#1{%
  \AtBeginShipoutAddToBox{%
    \kern-\AtBegShi@horigin\relax
    \vbox to 0pt{%
      \kern-\AtBegShi@vorigin\relax
      \AtBegShi@BeginPicture
      #1%
      \AtBegShi@EndPicture
      \vss
    }%
  }%
}
```

eginShipoutUpperLeftForeground

```
\def\AtBeginShipoutUpperLeftForeground#1{%
  \AtBeginShipoutAddToBoxForeground{%
    \kern-\AtBegShi@horigin\relax
    \vbox to 0pt{%
      \kern-\AtBegShi@vorigin\relax
      \AtBegShi@BeginPicture
      #1%
      \AtBegShi@EndPicture
      \vss
    }%
  }%
}
```

3.6 Patches

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

```
\def\AtBegShi@AbortIfUndefined#1{%
  \begingroup\expandafter\expandafter\expandafter\endgroup
  \expandafter\ifx\csname#1\endcsname\relax
    \expandafter\AtBegShi@AtEnd
  \fi
}
\AtBegShi@AbortIfUndefined{currentgrouplevel}%
\AtBegShi@AbortIfUndefined{AtBeginDocument}%
\AtBegShi@AbortIfUndefined{@ifpackageloaded}%
\AtBegShi@AbortIfUndefined{@ifclassloaded}%
```

3.6.1 Package crop

Fix of method and box.

```
\def\AtBegShi@PatchCrop{%
  \begingroup
    \def\AtBegShi@Crop@shipout{%
      \afterassignment\CROP@ship
```

```
  \setbox\@cclv=%
}%
\def\AtBegShi@Crop@ship{%
  \ifvoid\@cclv
    \expandafter\aftergroup
  \fi
  \CROP@@ship
}%
\def\AtBegShi@Crop@shiplist{%
  \lineskip\z@
  \lineskiplimit\z@
  \baselineskip\z@
  \CROP@kernel
  \box\@cclv
}%
\def\AtBegShi@Crop@@ship{%
  \CROP@shipout\vbox{%
    \CROP@shiplist
  }%
}%
\ifx\AtBegShi@Crop@ship\CROP@ship
  \ifx\AtBegShi@Crop@shiplist\CROP@shiplist
    \ifx\AtBegShi@Crop@@ship\CROP@@ship
      \let\AtBegShi@found\relax
      \ifx\shipout\AtBegShi@Crop@shipout
        \def\AtBegShi@found{\shipout}%
      \else\ifx\AtBeginShipoutOriginalShipout\AtBegShi@Crop@shipout
        \def\AtBegShi@found{\AtBeginShipoutOriginalShipout}%
      \else\ifx\@EveryShipout@Org@Shipout\AtBegShi@Crop@shipout
        \def\AtBegShi@found{\@EveryShipout@Org@Shipout}%
      \else\ifx\GPTorg@shipout\AtBegShi@Crop@shipout
        \def\AtBegShi@found{\GPTorg@shipout}%
      \else\ifx\THBorg@shipout\AtBegShi@Crop@shipout
        \def\AtBegShi@found{\THBorg@shipout}%
      \else\ifx\mem@oldshipout\AtBegShi@Crop@shipout
        \def\AtBegShi@found{\mem@oldshipout}%
      \fi\fi\fi\fi\fi\fi
      \ifx\AtBegShi@found\relax
      \else
        \expandafter\endgroup
        \expandafter\def\AtBegShi@found{%
          \edef\AtBegShi@GroupLevel{\number\currentgrouplevel}%
          \afterassignment\CROP@ship
          \setbox\AtBeginShipoutBox=%
        }%
        \def\CROP@ship{%
          \ifnum\AtBegShi@GroupLevel=\currentgrouplevel
          \else
            \expandafter\aftergroup
          \fi
          \CROP@@ship
        }%
        \def\CROP@shiplist{%
          \lineskip 0pt\relax
          \lineskiplimit 0pt\relax
          \baselineskip 0pt\relax
          \CROP@kernel
          \box\AtBeginShipoutBox
        }%
        \def\CROP@@ship{%
          \ifvoid\AtBeginShipoutBox
          \else
```

```
            \setbox\AtBeginShipoutBox=\vbox{%
              \CROP@shiplist
            }%
            \AtBegShi@GetBoxSize\AtBeginShipoutBox
            \expandafter\CROP@shipout
            \expandafter\box
            \expandafter\AtBeginShipoutBox
          \fi
        }%
        \@PackageInfoNoLine{atbegshi}{Package `crop' patched}%
        \begingroup
      \fi
    \fi
  \fi
 \endgroup
 \let\AtBegShi@PatchCrop\relax
}
\@ifpackageloaded{crop}{%
 \AtBegShi@PatchCrop
}{%
 \AtBeginDocument{\AtBegShi@PatchCrop}%
}
```

3.6.2 Package everyshi

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

```
\def\AtBegShi@PatchEveryshi{%
 \begingroup
  \long\def\AtBegShi@Everyshi@shipout{%
   \afterassignment\@EveryShipout@Test
   \global\setbox\@cclv= %
  }%
  \long\def\AtBegShi@Everyshi@Test{%
   \ifvoid\@cclv\relax
    \aftergroup\@EveryShipout@Output
   \else
    \@EveryShipout@Output
   \fi
  }%
  \ifx\AtBegShi@Everyshi@Test\@EveryShipout@Test
   \let\AtBegShi@found\relax
   \ifx\shipout\AtBegShi@Everyshi@shipout
    \def\AtBegShi@found{\shipout}%
   \else\ifx\AtBeginShipoutOriginalShipout\AtBegShi@Everyshi@shipout
    \def\AtBegShi@found{\AtBeginShipoutOriginalShipout}%
   \else\ifx\CROP@shipout\AtBegShi@Everyshi@shipout
    \def\AtBegShi@found{\CROP@shipout}%
   \else\ifx\GPTorg@shipout\AtBegShi@Everyshi@shipout
    \def\AtBegShi@found{\GPTorg@shipout}%
   \else\ifx\THBorg@shipout\AtBegShi@Everyshi@shipout
    \def\AtBegShi@found{\THBorg@shipout}%
   \else\ifx\mem@oldshipout\AtBegShi@Everyshi@shipout
    \def\AtBegShi@found{\mem@oldshipout}%
   \else
    \expandafter\ifx\csname @EveryShipout@Org@Shipout\endcsname
                \relax
      \ifx\@EveryShipout@Shipout\AtBegShi@Everyshi@shipout
       \def\AtBegShi@found{\@EveryShipout@Shipout}%
      \fi
    \fi
   \fi\fi\fi\fi\fi\fi
```

```
    \ifx\AtBegShi@found\relax
    \else
      \expandafter\endgroup
      \expandafter\def\AtBegShi@found{%
        \edef\AtBegShi@GroupLevel{\number\currentgrouplevel}%
        \afterassignment\@EveryShipout@Test
        \setbox\AtBeginShipoutBox=%
      }%
      \def\@EveryShipout@Test{%
        \ifnum\AtBegShi@GroupLevel=\currentgrouplevel
        \else
          \expandafter\aftergroup
        \fi
        \AtBegShi@Everyshi@Output
      }%
      \def\AtBegShi@Everyshi@Output{%
        \ifvoid\AtBeginShipoutBox
        \else
          \global\setbox\ltx@cclv\box\AtBeginShipoutBox
          \expandafter\@EveryShipout@Output
        \fi
      }%
      \@PackageInfoNoLine{atbegshi}{Package `everyshi' patched}%
      \begingroup
    \fi
  \fi
  \endgroup
  \let\AtBegShi@PatchEveryshi\relax
}
\@ifpackageloaded{everyshi}{%
  \AtBegShi@PatchEveryshi
}{%
  \AtBeginDocument{\AtBegShi@PatchEveryshi}%
}
```

3.6.3 Class memoir

Fix of method and box.

```
\def\AtBegShi@PatchMemoir{%
  \begingroup
    \def\AtBegShi@Memoir@shipout{%
      \afterassignment\mem@shipi
      \setbox\@cclv=%
    }%
    \def\AtBegShi@Memoir@shipi{%
      \ifvoid\@cclv
        \expandafter\aftergroup
      \fi
      \mem@shipii
    }%
    \def\AtBegShi@Memoir@shipiiA{%
      \mem@oldshipout\vbox{%
        \trimmarks
        \unvbox\@cclv
      }%
    }%
    \def\AtBegShi@Memoir@shipiiB{%
      \ifvoid\@cclv
        \mem@oldshipout\box\@cclv
      \else
        \mem@oldshipout\vbox{%
          \trimmarks
```

```
      \unvbox\@cclv
    }%
  \fi
}%
\def\AtBegShi@Memoir@PatchAB{%
  \ifvoid\AtBeginShipoutBox
  \else
    \setbox\AtBeginShipoutBox=\vbox{%
      \trimmarks
      \ifvbox\AtBeginShipoutBox
        \unvbox\AtBeginShipoutBox
      \else
        \box\AtBeginShipoutBox
      \fi
    }%
    \AtBegShi@GetBoxSize\AtBeginShipoutBox
    \expandafter\mem@oldshipout
    \expandafter\box
    \expandafter\AtBeginShipoutBox
  \fi
}%
\def\AtBegShi@Memoir@shipiiC{% 2008/08/07 v1.6180339a
  \ifvoid\@cclv
    \mem@oldshipout\box\@cclv
  \else
    \ifshowtrims
      \mem@oldshipout\vbox{\trimmarks\unvbox\@cclv}%
    \else
      \mem@oldshipout\box\@cclv
    \fi
  \fi
}%
\def\AtBegShi@Memoir@shipiiD{% 2011/03/06 v3.6j
  \ifvoid\@cclv
    \mem@oldshipout\box\@cclv
  \else
    \ifshowtrims
      \mem@oldshipout\vbox{%
        \trimmarks
        \nointerlineskip
        \box\@cclv
      }%
    \else
      \mem@oldshipout\box\@cclv
    \fi
  \fi
}%
\def\AtBegShi@Memoir@PatchCD{%
  \ifvoid\AtBeginShipoutBox
  \else
    \ifshowtrims
      \setbox\AtBeginShipoutBox=\vbox{%
        \trimmarks
        \nointerlineskip
        \box\AtBeginShipoutBox
      }%
    \fi
    \AtBegShi@GetBoxSize\AtBeginShipoutBox
    \expandafter\mem@oldshipout
    \expandafter\box
    \expandafter\AtBeginShipoutBox
  \fi
```

```
  }%
  \ifx\AtBegShi@Memoir@shipi\mem@shipi
    \let\AtBegShi@found\ltx@one
    \ifx\AtBegShi@Memoir@shipiiA\mem@shipii
      \let\AtBegShi@found\ltx@zero
      \global\let\AtBegShi@Memoir@PatchX\AtBegShi@Memoir@PatchAB
    \else\ifx\AtBegShi@Memoir@shipiiB\mem@shipii
      \let\AtBegShi@found\ltx@zero
      \global\let\AtBegShi@Memoir@PatchX\AtBegShi@Memoir@PatchAB
    \else\ifx\AtBegShi@Memoir@shipiiC\mem@shipii
      \let\AtBegShi@found\ltx@zero
      \global\let\AtBegShi@Memoir@PatchX\AtBegShi@Memoir@PatchCD
    \else\ifx\AtBegShi@Memoir@shipiiD\mem@shipii
      \let\AtBegShi@found\ltx@zero
      \global\let\AtBegShi@Memoir@PatchX\AtBegShi@Memoir@PatchCD
    \fi\fi\fi\fi
    \ifcase\AtBegShi@found
      \let\AtBegShi@found\relax
      \ifx\shipout\AtBegShi@Memoir@shipout
        \def\AtBegShi@found{\shipout}%
      \else\ifx\AtBeginShipoutOriginalShipout\AtBegShi@Memoir@shipout
        \def\AtBegShi@found{\AtBeginShipoutOriginalShipout}%
      \else\ifx\CROP@shipout\AtBegShi@Memoir@shipout
        \def\AtBegShi@found{\CROP@shipout}%
      \else\ifx\GPTorg@shipout\AtBegShi@Memoir@shipout
        \def\AtBegShi@found{\GPTorg@shipout}%
      \else\ifx\THBorg@shipout\AtBegShi@Memoir@shipout
        \def\AtBegShi@found{\THBorg@shipout}%
      \else\ifx\@EveryShipout@Org@Shipout\AtBegShi@Memoir@shipout
        \def\AtBegShi@found{\@EveryShipout@Org@Shipout}%
      \fi\fi\fi\fi\fi\fi
      \ifx\AtBegShi@found\relax
      \else
        \expandafter\endgroup
        \expandafter\def\AtBegShi@found{%
          \edef\AtBegShi@GroupLevel{\number\currentgrouplevel}%
          \afterassignment\mem@shipi
          \setbox\AtBeginShipoutBox=%
        }%
        \def\mem@shipi{%
          \ifnum\AtBegShi@GroupLevel=\currentgrouplevel
          \else
            \expandafter\aftergroup
          \fi
          \mem@shipii
        }%
        \let\mem@shipii\AtBegShi@Memoir@PatchX
        \@PackageInfoNoLine{atbegshi}{Class `memoir' patched}%
        \begingroup
      \fi
    \fi
  \fi
  \endgroup
  \let\AtBegShi@PatchMemoir\relax
}
\@ifclassloaded{memoir}{%
  \AtBegShi@PatchMemoir
}{%
  \AtBeginDocument{\AtBegShi@PatchMemoir}%
}
\AtBegShi@AtEnd%
⟨/package⟩
```

4 Test

4.1 Catcode checks for loading

879 ⟨*test1⟩

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

```
      \errmessage{%
        Character \the\count@\space
        with wrong catcode \the\catcode\count@\space
        instead of \number#3%
      }%
    \fi
  \ifnum\count@<#2\relax
    \advance\count@ 1 %
  \repeat
}
\def\space{ }
\expandafter\ifx\csname LoadCommand\endcsname\relax
  \def\LoadCommand{\input atbegshi.sty\relax}%
\fi
\def\Test{%
  \RangeCatcodeInvalid{0}{47}%
  \RangeCatcodeInvalid{58}{64}%
  \RangeCatcodeInvalid{91}{96}%
  \RangeCatcodeInvalid{123}{255}%
  \catcode`\@=12 %
  \catcode`\\=0 %
  \catcode`\%=14 %
  \LoadCommand
  \RangeCatcodeCheck{0}{36}{15}%
  \RangeCatcodeCheck{37}{37}{14}%
  \RangeCatcodeCheck{38}{47}{15}%
  \RangeCatcodeCheck{48}{57}{12}%
  \RangeCatcodeCheck{58}{63}{15}%
  \RangeCatcodeCheck{64}{64}{12}%
  \RangeCatcodeCheck{65}{90}{11}%
  \RangeCatcodeCheck{91}{91}{15}%
  \RangeCatcodeCheck{92}{92}{0}%
  \RangeCatcodeCheck{93}{96}{15}%
  \RangeCatcodeCheck{97}{122}{11}%
  \RangeCatcodeCheck{123}{255}{15}%
  \RestoreCatcodes
}
\Test
\csname @@end\endcsname
\end
```

⟨/test1⟩

⟨*test2⟩

```
\input atbegshi.sty\relax
\def\msg#{\immediate\write16}
\msg{File: atbegshi-test2.tex 2016/06/09 v1.18 Test file for plain-TeX}
\def\testmsg#1#2{%
  \msg{}%
  \msg{*** Test with box (#1), expected page output [#2]}% hash-ok
}

\newbox\voidbox
\def\void{\box\voidbox}
\begingroup
  \setbox\voidbox=\void
\endgroup

\count0=0\relax
\AtBeginShipout{%
  \global\advance\count0 by 1\relax
  \msg{* Inside \string\AtBeginShipout: [\the\count0]}%
}

```

```
\AtBeginShipoutFirst{%
  \msg{* Inside \string\AtBeginShipoutFirst}%
  Hello World%
}

\testmsg{\string\null}{1}
\shipout\null

\AtBeginShipoutFirst{%
  This is too late%
}

\testmsg{void}{}
\shipout\void

\testmsg{\string\copy255 (not void)}{2}
\setbox255\hbox{\vrule height 10bp width 10bp}
\shipout\copy255 %

\testmsg{\string\copy255 (again)}{3}
\shipout\copy255 %

\testmsg{\string\box255}{4}
\shipout\box255 %

\testmsg{\string\box255 (again)}{}
\shipout\box255 %

\testmsg{\string\hbox}{5}
\shipout\hbox{\vrule height 5bp width 20bp}

\testmsg{\string\vbox}{6}
\shipout\vbox{\hrule height 20bp width 5bp}

\testmsg{\string\null, voided by hook}{}
\def\VoidBox{%
  \begingroup
    \setbox\AtBeginShipoutBox=\box\AtBeginShipoutBox
  \endgroup
}
\AtBeginShipout{\VoidBox}
\shipout\null
\def\VoidBox{}

\msg{*** \string\begingroup}
\begingroup
  \testmsg{void}{}%
  \shipout\void
\msg{*** \string\endgroup}
\endgroup

\msg{*** \string\begingroup}
\begingroup
  \testmsg{void}{}%
  \shipout\void
  \testmsg{\string\null}{8}%
  \shipout\null
\msg{*** \string\endgroup}
\endgroup

\testmsg{output routine}{9}
Hello World
```

```
\vfill
\eject

\testmsg{\string\null\space(discarded)}{}
\AtBeginShipout{%
  \msg{* Inside \string\AtBeginShipout: DISCARD}%
  \AtBeginShipoutDiscard
}
\shipout\null

\end
```
⟨/test2⟩

⟨*test3⟩
```
\NeedsTeXFormat{LaTeX2e}
\ProvidesFile{atbegshi-test3.tex}[2016/06/09 v1.18 Test file for LaTeX]
\RequirePackage{color}
\pagecolor{yellow}
\documentclass[a5paper,showtrims]{memoir}
\usepackage{atbegshi}
\AtBeginShipout{%
  \setbox\AtBeginShipoutBox=\vbox{%
    \vbox to 0pt{%
      \kern-1.5in %
      \hbox to 0pt{%
        \kern-1.5in %
        \color{blue}%
        \rule{1in}{1in}%
        \hss
      }%
      \vss
    }%
    \hrule
    \hbox{\vrule\box\AtBeginShipoutBox\vrule}%
    \hrule
  }%
}
\usepackage{eso-pic}
\makeatletter
\@ifundefined{@EveryShipout@Init}{%
  \typeout{Test skipped}%
  \@@end
}{}
\@EveryShipout@Init
\let\@EveryShipout@Init\relax
\makeatother
\AddToShipoutPicture{%
  \hspace{.52\paperwidth}%
  \colorbox{cyan}{%
    \rule{0mm}{\paperheight}%
    \hspace{.48\paperwidth}%
  }%
}
```

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

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

5 Installation

5.1 Download

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

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

CTAN:macros/latex/contrib/oberdiek/atbegshi.pdf Documentation.

Bundle. All the packages of the bundle 'oberdiek' are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

CTAN:install/macros/latex/contrib/oberdiek.tds.zip

TDS refers to the standard "A Directory Structure for TEX Files" (CTAN:tds/tds.pdf). Directories with texmf in their name are usually organized this way.

5.2 Bundle installation

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

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

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

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

5.3 Package installation

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

```
tex atbegshi.dtx
```

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

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

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

5.4 Refresh file name databases

If your TEX distribution (teTEX, mikTEX, ...) relies on file name databases, you must refresh these. For example, teTEX users run texhash or mktexlsr.

[1] http://ctan.org/pkg/atbegshi

5.5 Some details for the interested

Unpacking with LaTeX. The .dtx chooses its action depending on the format:

plain TeX: Run docstrip and extract the files.

LaTeX: Generate the documentation.

If you insist on using LaTeX for docstrip (really, docstrip does not need LaTeX), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{atbegshi.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

Generating the documentation. You can use both the .dtx or the .drv to generate the documentation. The process can be configured by the configuration file ltxdoc.cfg. For instance, put this line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with pdfLaTeX:

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

6 Catalogue

The following XML file can be used as source for the TeX Catalogue. The elements caption and description are imported from the original XML file from the Catalogue. The name of the XML file in the Catalogue is atbegshi.xml.

```
⟨*catalogue⟩
<?xml version='1.0' encoding='us-ascii'?>
<!DOCTYPE entry SYSTEM 'catalogue.dtd'>
<entry datestamp='$Date$' modifier='$Author$' id='atbegshi'>
  <name>atbegshi</name>
  <caption>Execute stuff at \shipout time.</caption>
  <authorref id='auth:oberdiek'/>
  <copyright owner='Heiko Oberdiek' year='2007-2011'/>
  <license type='lppl1.3'/>
  <version number='1.18'/>
  <description>
    This package is a modern reimplementation of package
    <xref refid='everyshi'>everyshi</xref>, providing various commands
    to be executed before a <tt>\shipout</tt> command.  It makes use of
    e-TeX&amp;#x2019;s facilities if they are available.  The package may
    be used either with LaTeX or with plain TeX.
    <p/>
    The package is part of the <xref refid='oberdiek'>oberdiek</xref> bundle.
  </description>
  <documentation details='Package documentation'
      href='ctan:/macros/latex/contrib/oberdiek/atbegshi.pdf'/>
  <ctan file='true' path='/macros/latex/contrib/oberdiek/atbegshi.dtx'/>
  <miktex location='oberdiek'/>
  <texlive location='oberdiek'/>
  <install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip'/>
</entry>
⟨/catalogue⟩
```

7 History

[2007/04/17 v1.0]

- First version.

[2007/04/18 v1.1]

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

[2007/04/19 v1.2]

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

[2007/04/26 v1.3]

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

[2007/04/27 v1.4]

- Small optimizations.

[2007/06/06 v1.5]

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

[2007/09/09 v1.6]

- Catcode section rewritten.

[2008/07/18 v1.7]

- Documentation of `\AtBeginShipoutUpperLeft` fixed and extended.

[2008/07/19 v1.8]

- `\AtBeginShipoutUpperLeftForeground` added.

[2008/07/31 v1.9]

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

[2009/12/02 v1.10]

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

[2010/03/01 v1.11]

- Compatibility with ini-TEX except for \newbox.

[2010/03/25 v1.12]

- \AtBeginShipoutNext can now be used inside \AtBeginShipoutNext.

[2010/08/18 v1.13]

- Fixes for \AtBegShi@CheckDefinable.

[2010/12/02 v1.14]

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

[2011/01/30 v1.15]

- Already loaded package files are not input in plain TEX.

[2011/10/05 v1.16]

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

[2016/05/16 v1.17]

- Documentation updates.

[2016/06/09 v1.18]

- Update for \pdfhorign in new LuaTeX.

8 Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; plain numbers refer to the code lines where the entry is used.

Symbols

\# 882
\% 958
\@ 883, 956
\@@end 1101
\@EveryShipout@Init 1103, 1104
\@EveryShipout@Org@Shipout 605, 606, 845, 846
\@EveryShipout@Output . 670, 672, 716
\@EveryShipout@Shipout 692, 693
\@EveryShipout@Test 665, 675, 702, 705
\@PackageError 194, 338
\@PackageInfoNoLine 248, 648, 719, 864
\@PackageWarning 238, 300, 326
\@cclv 260, 577, 580, 590, 666, 669, 735, 738, 746, 750, 751, 755, 777, 778, 781, 783, 788, 789, 795, 798
\@ehc 341
\@ehd 196
\@firstofone 891, 894
\@gobble 888, 896
\@ifclassloaded 872
\@ifdefinable 201
\@ifpackageloaded 657, 726
\@ifundefined 1099
\@undefined 113, 188, 491, 493
\\ 957
\{ 880
\} 881

A

\AddToShipoutPicture 1106
\advance . . 392, 453, 921, 929, 944, 995
\afterassignment 223, 576, 619, 665, 702, 734, 853
\aftergroup 84, 230, 581, 625, 670, 708, 739, 859
\AtBeginDocument . 354, 660, 729, 875
\AtBeginShipout 2, 6, 44, 287, 994, 996, 1039, 1065, 1066, 1080
\AtBeginShipoutAddToBox 3, 279, 356, 540
\AtBeginShipoutAddToBoxForeground 415, 552
\AtBeginShipoutBox 45, 47, 220, 225, 237, 251, 257, 262, 325, 343, 357, 364, 374, 375, 379, 386, 387, 391, 392, 393, 407, 416, 423, 424, 425, 435, 436, 440, 447, 448, 452, 453, 454, 467, 620, 634, 637, 639, 642, 645, 703, 713, 715, 760, 762, 764, 765, 767, 770, 773, 803, 806, 809, 812, 815, 854, 1036, 1081, 1093
\AtBeginShipoutBoxDepth . . . 269, 273
\AtBeginShipoutBoxHeight . . . 268, 272
\AtBeginShipoutBoxWidth . 4, 267, 271
\AtBeginShipoutDiscard 3, 29, 205, 330, 1067
\AtBeginShipoutFirst 3, 282, 295, 302, 999, 1000, 1007
\AtBeginShipoutInit . . 3, 335, 352, 354
\AtBeginShipoutNext . . . 3, 14, 28, 291
\AtBeginShipoutOriginalShipout 4, 262, 344, 345, 603, 604, 679, 680, 837, 838
\AtBeginShipoutUpperLeft 4, 7, 15, 539
\AtBeginShipoutUpperLeftForeground 4, 551
\AtBegShi@AbortIfUndefined 563, 569, 570, 571, 572
\AtBegShi@AddHook 289, 293, 316
\AtBegShi@AddTo 297, 305, 317
\AtBegShi@AtEnd 150, 151, 167, 566, 877
\AtBegShi@BeginPicture 510, 526, 544, 556
\AtBegShi@CheckDefinable 182, 205, 287, 291, 295, 335, 344
\AtBegShi@Crop@@ship 592, 599
\AtBegShi@Crop@ship 579, 597
\AtBegShi@Crop@shiplist 585, 598
\AtBegShi@Crop@shipout 575, 601, 603, 605, 607, 609, 611
\AtBegShi@Discardedfalse 242, 249
\AtBegShi@Discardedtrue 208
\AtBegShi@EndPicture 520, 531, 546, 558
\AtBegShi@Everyshi@Output . 710, 712
\AtBegShi@Everyshi@shipout . 664, 677, 679, 681, 683, 685, 687, 692
\AtBegShi@Everyshi@Test 668, 675
\AtBegShi@First 255, 276
\AtBegShi@FirstDisabled 282, 299
\AtBegShi@found . . . 600, 602, 604, 606, 608, 610, 612, 614, 617, 676, 678, 680, 682, 684, 686, 688, 693, 697, 700, 819, 821, 824, 827, 830, 833, 834, 836, 838, 840, 842, 844, 846, 848, 851
\AtBegShi@GetBoxSize 257, 260, 266, 642, 770, 812
\AtBegShi@GroupLevel 222, 229, 618, 623, 701, 706, 852, 857
\AtBegShi@Hook 243, 284, 289
\AtBegShi@HookFirst 277, 279, 286, 297
\AtBegShi@HookNext 244, 246, 285, 293
\AtBegShi@horigin . 490, 495, 541, 553
\AtBegShi@Item 317, 319
\AtBegShi@Memoir@PatchAB 759, 822, 825
\AtBegShi@Memoir@PatchCD 802, 828, 831
\AtBegShi@Memoir@PatchX 822, 825, 828, 831, 863
\AtBegShi@Memoir@shipi 737, 818
\AtBegShi@Memoir@shipiiA . . 743, 820
\AtBegShi@Memoir@shipiiB . . 749, 823
\AtBegShi@Memoir@shipiiC . . 776, 826
\AtBegShi@Memoir@shipiiD . . 787, 829
\AtBegShi@Memoir@shipout 733, 835, 837, 839, 841, 843, 845
\AtBegShi@OrgProtect . . 240, 253, 256
\AtBegShi@Output 232, 234
\AtBegShi@PatchCrop 573, 655, 658, 660
\AtBegShi@PatchEveryshi 662, 724, 727, 729
\AtBegShi@PatchMemoir 731, 870, 873, 875
\AtBegShi@restore 358, 367, 377, 380, 396, 410, 417, 428, 438, 441, 457, 472
\AtBegShi@restorebox 385, 411, 446, 473
\AtBegShi@Shipout 218, 346
\AtBegShi@Test 223, 227
\AtBegShi@vorigin . 492, 496, 543, 555
\AtNextShipout 260

B

\baselineskip 403, 464, 588, 632
\begin 11, 50, 1115
\body 900, 904
\box 47, 251, 262, 590, 634, 644, 715, 751, 767, 772, 778, 783, 789, 795, 798, 809, 814, 988, 1021, 1022, 1024, 1025, 1036, 1093, 1117

C

\catcode 57, 58, 60, 61, 62, 63, 64, 65, 66, 67, 68, 88, 89, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 124, 125, 127, 128, 129, 133, 134, 135, 136, 137, 138, 139, 142, 143, 145, 146, 147, 148, 152, 154, 212, 213, 215, 216,

274, 275, 880, 881, 882, 883, 918, 927, 935, 939, 956, 957, 958
\circle 8
\color 16, 1086
\colorbox 1108
\copy 374, 406, 435, 469, 536, 1014, 1016, 1018, 1019
\count 993, 995, 996
\count@ 885, 914, 918, 920, 921, 925, 927, 928, 929, 933, 935, 938, 939, 943, 944
\countdef 885
\CROP@@ship 583, 599, 627, 636
\CROP@kernel 589, 633
\CROP@ship 576, 597, 619, 622
\CROP@shiplist . . . 594, 598, 629, 640
\CROP@shipout 593, 643, 681, 682, 839, 840
\csname 69, 76, 105, 121, 131, 169, 172, 183, 211, 241, 306, 340, 343, 351, 478, 481, 500, 503, 514, 565, 690, 884, 887, 890, 893, 948, 975, 1113, 1117
\currentgrouplevel 222, 229, 618, 623, 701, 706, 852, 857

D

\deadcycles 207
\dimen 383, 391, 392, 393, 444, 452, 453, 454, 468, 470
\documentclass 2, 37, 1078
\dp 269, 373, 387, 392, 402, 434, 448, 453, 463, 535

E

\E 275
\eject 1062
\empty 72, 73
\end 34, 54, 976, 1071, 1119
\endcsname 69, 76, 105, 121, 131, 169, 172, 183, 211, 241, 306, 340, 343, 351, 478, 481, 500, 503, 514, 565, 690, 884, 887, 890, 893, 948, 975, 1113, 1117
\endinput 84, 167
\endlinechar 59, 90, 126, 132, 144
\endpicture 521
\errmessage 937

F

\fill 25

G

\g@addto@macro 314
\gdef 244, 284, 285, 286, 348
\GPTorg@shipout 607, 608, 683, 684, 841, 842

H

\hbadness 360, 363, 419, 422
\hbox 45, 220, 364, 365, 394, 423, 426, 455, 527, 1015, 1027, 1028, 1084, 1093
\hfuzz 359, 362, 418, 421
\hrule 1031, 1092, 1094
\hspace 1107, 1110
\hss 1088
\ht 268, 372, 374, 386, 391, 401, 433, 435, 447, 452, 462, 534

I

\ifAtBegShi@Discarded 204, 247, 320, 323
\ifcase 185, 499, 833
\ifdim 228
\ifhbox 357, 416
\ifnum 229, 623, 706, 857, 920, 928, 935, 943
\ifpdf 489
\ifshowtrims 780, 791, 805
\ifvbox 379, 440, 764
\ifvoid 237, 325, 580, 637, 669, 713, 738, 750, 760, 777, 788, 803
\ifx 70, 73, 76, 105, 113, 116, 169, 172, 183, 185, 188, 211, 277, 306, 351, 478, 481, 491, 493, 500, 503, 514, 565, 597, 598, 599, 601, 603, 605, 607, 609, 611, 614, 675, 677, 679, 681, 683, 685, 687, 690, 692, 697, 818, 820, 823, 826, 829, 835, 837, 839, 841, 843, 845, 848, 884, 887, 890, 893, 948
\ignorespaces 518, 529
\immediate 78, 107, 980
\input 173, 482, 949, 979
\iterate 901, 903, 905

K

\kern 221, 408, 425, 436, 468, 470, 541, 543, 553, 555, 1083, 1085

L

\lastkern 228
\line 17, 18
\lineskip 404, 465, 586, 630
\lineskiplimit 405, 466, 587, 631
\LoadCommand 949, 959
\loop 899, 915, 926, 934
\ltx@cclv 715
\ltx@empty 277, 281
\ltx@IfUndefined 337
\ltx@ifundefined 258
\ltx@newif 204
\ltx@one 186, 189, 819
\ltx@zero . . . 191, 207, 309, 310, 365, 371, 372, 373, 374, 383, 391, 392, 393, 394, 400, 401, 402, 406, 426, 432, 433, 434, 435, 444, 452, 453, 454, 455, 461, 462, 463, 468, 469, 470, 527, 534, 535, 536, 821, 824, 827, 830

M

\makeatletter 1098
\makeatother 1105

\mem@oldshipout 611, 612, 687, 688, 744, 751, 753, 771, 778, 781, 783, 789, 792, 798, 813
\mem@shipi 734, 818, 853, 856
\mem@shipii 741, 820, 823, 826, 829, 861, 863
\MessageBreak 301, 327, 339
\msg 980, 981, 983, 984, 996, 1000, 1043, 1047, 1050, 1056, 1066

N

\NeedsTeXFormat 1074
\newbox 987
\newpage 13, 22, 27, 32, 52
\next 905, 907, 909
\nointerlineskip 794, 808
\null 1004, 1005, 1033, 1040, 1054, 1055, 1064, 1069, 1116
\number 222, 618, 701, 852, 940

P

\p@ 221
\PackageInfo 81
\pagecolor 1077
\paperheight 8, 17, 18, 1109
\paperwidth 8, 17, 18, 1107, 1110
\par 24
\pdfhorigin 491
\pdfvariable 491, 493
\pdfvorigin 493
\picture 512
\protect 240, 253, 256
\ProvidesFile 1075
\ProvidesPackage 74, 122
\put 8, 17, 18

R

\raise 374, 435
\RangeCatcodeCheck 932, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971
\RangeCatcodeInvalid 924, 952, 953, 954, 955
\repeat 899, 911, 922, 930, 945
\RequirePackage . . . 179, 180, 487, 1076
\RestoreCatcodes . . 913, 916, 917, 972
\rule 1087, 1109

S

\section 12, 1118
\setbox 45, 220, 225, 251, 364, 365, 393, 394, 423, 426, 454, 455, 527, 577, 620, 639, 666, 703, 715, 735, 762, 806, 854, 990, 1015, 1036, 1081
\shipout 345, 346, 601, 602, 677, 678, 835, 836, 1005, 1012, 1016, 1019, 1022, 1025, 1028, 1031, 1040, 1046, 1053, 1055, 1069, 1116, 1117, 1126, 1134
\space . . . 195, 339, 938, 939, 947, 1064
\special 46

T

\Test 951, 974
\testmsg 982, 1004, 1011, 1014, 1018, 1021, 1024, 1027, 1030, 1033, 1045, 1052, 1054, 1059, 1064
\THBorg@shipout 609, 610, 685, 686, 843, 844
\the . . . 132, 133, 134, 135, 136, 137, 138, 139, 152, 267, 268, 269, 310, 359, 360, 381, 382, 383, 386, 387, 418, 419, 442, 443, 444, 447, 448, 918, 938, 939, 996
\TMP@EnsureCode 149, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166
\TMP@RequirePackage 170, 176, 177, 479, 485
\toks 309, 310
\trimmarks 745, 754, 763, 781, 793, 807
\typeout 1100

U

\unhcopy 375, 424
\unitlength 516
\unvbox . . 407, 467, 746, 755, 765, 781
\usepackage 3, 4, 5, 38, 39, 1079, 1097, 1114

V

\vbadness 382, 390, 443, 451
\vbox 393, 454, 542, 554, 593, 639, 744, 753, 762, 781, 792, 806, 1030, 1031, 1081, 1082
\vfill 1061
\vfuzz 381, 389, 442, 450
\void 988, 990, 1012, 1046, 1053
\VoidBox 1034, 1039, 1041
\voidbox 987, 988, 990
\vrule 1015, 1028, 1093
\vspace 25
\vss 547, 559, 1090

W

\wd 267, 364, 371, 400, 423, 425, 432, 436, 461
\write 78, 107, 980

X

\X 274
\x 69, 70, 73, 77, 81, 83, 106, 111, 121, 130, 142

Z

\z@ 586, 587, 588