

The `ltshipout` package*

Frank Mittelbach

July 21, 2020

Contents

1	Introduction	1
1.1	Overloading the <code>\shipout</code> primitive	2
1.2	Provided hooks	3
1.3	Special commands for use inside the hooks	4
1.4	Information counters	5
1.5	Debugging shipout code	5
2	Emulating commands from other packages	5
2.1	Emulating <code>atbegshi</code>	6
2.2	Emulating <code>everyshi</code>	6
2.3	Emulating <code>atenddvi</code>	7
2.4	Emulating <code>everypage</code>	7
3	The Implementation	7
3.1	Debugging	7
3.2	Handling the end of job hook	15
4	Legacy \LaTeX 2ϵ interfaces	18
5	Package emulation for compatibility	18
5.1	Package <code>atbegshi</code> emulation	18
5.2	Package <code>everyshi</code> emulation	19
5.3	Package <code>atenddvi</code> emulation	19
	Index	20

1 Introduction

The code provides an interface to the `\shipout` primitive of \TeX which is called when a finished pages is finally “shipped out” to the target output file, e.g., the `.dvi` or `.pdf` file. A good portion of the code is based on ideas by Heiko Oberdiek implemented in his packages `atbegshi` and `atenddvi` even though the interfaces are somewhat different.¹

*This package has version v0.9a dated 2020/07/15, © \LaTeX Project.

¹Heiko’s interfaces are emulated by the kernel code, if a document requests his packages, so older documents will continue to work.

1.1 Overloading the `\shipout` primitive

`\shipout` With this implementation T_EX's `shipout` primitive is no longer available for direct use. Instead `\shipout` is running some (complicated) code that picks up the box to be shipped out regardless of how that is done, i.e., as a constructed `\vbox` or `\hbox` or as a box register.

It then stores it in a named box register. This box can then be manipulated through a set of hooks after which it is shipped out for real.

`\ShipoutBox` This box register is called `\ShipoutBox` (alternatively available via the L3 name `\l_`
`\l_shipout_box` `shipout_box`).

`\l_shipout_box_ht_dim`
`\l_shipout_box_dp_dim`
`\l_shipout_box_wd_dim`
`\l_shipout_box_ht_plus_dp_dim`

The `shipout` box dimensions are available in the L3 registers `\l_shipout_box_ht_dim`, etc. (there are no L^AT_EX 2_ε names).² These variables can be used inside the hook code for `shipout/before`, `shipout/foreground` and `shipout/background` if needed.

²Might need changing, but HO's version as strings is not really helpful I think).

1.2 Provided hooks

`shipout/before`
`shipout/foreground`
`shipout/background`
`shipout/firstpage`
`shipout/lastpage`

The code offers a number of hooks into which packages (or the user) can add code to support different use cases. These are:

shipout/before This hook is executed after the finished page has been stored in `\ShipoutBox` / `\l_shipout_box`. It can be used to alter that box content or to discard it completely (see `\DiscardShipoutBox` below).

shipout/background This hook adds a picture environment into the background of the page (with the (0,0) coordinate in the top-left corner using a `\unitlength` of `1pt`. It should therefore only receive `\put` commands or other commands suitable in a `picture` environment and the vertical coordinate values would normally be negative.

Todo: Again, mainly atbegshi compatibility. Not sure it is best to have to always use negative coordinates.

Technically this is implemented by adding a zero-sized `\hbox` as the very first item into the `\ShipoutBox` containing that `picture` environment. Thus the rest of the box content will overprint what ever is typeset by that hook.

shipout/foreground This hook adds a picture environment into the foreground of the page (with the (0,0) coordinate in the top-left corner using a `\unitlength` of `1pt`. Technically this is implemented by adding a zero-sized `\hbox` as the very last item into the `\ShipoutBox` and raising it up so that it still has its (0,0) point in the top-left corner. But being placed after the main box content it will be typeset later and thus overprints it (i.e., is in the foreground).

shipout/firstpage The material from this hook is executed only once at the very beginning of the first output page. It should only contain `\special` commands needed to direct post processors handling the `.dvi` or `.pdf` output.

Todo: not sure it has to be that restrictive.

In $\text{\LaTeX} 2_{\epsilon}$ that was already existing but implemented as a box register `\@begindvibox`.

Todo: drop or at least mark the obsolete code in latex.ltx

shipout/lastpage The corresponding hook to add `\specials` at the very end of the output file. It is only entered on the very last page.

It may not be possible for \LaTeX to correctly determine which page is the last one without several reruns. If this happens and the hook is non-empty then \LaTeX will add an extra page to place the material and also request a rerun to get the correct placement sorted out.

As mentioned above the hook `shipout/before` is executed first and can manipulate the prepared shipout box stored in `\ShipoutBox` or set things up for use in `\write` during the actual shipout. The other hooks are added inside `hboxes` to the box being shipped out in the following order:

<code>shipout/firstpage</code>	only on the first page
<code>shipout/background</code>	
<code><boxed content of \ShipoutBox></code>	
<code>shipout/foreground</code>	
<code>shipout/lastpage</code>	only on the last page

If any of the hooks has no code then that particular no box is added at that point.

In a document that doesn't produce pages, e.g., only makes `\typeouts`, none of the hooks are executed (as there is no `\shipout`) not even the `shipout/lastpage` hook.

`\AtBeginDvi`
`\AtEndDvi`

`\AtBeginDvi` is the existing L^AT_EX 2_ε interface to fill the `shipout/firstpage` hook. This is not really a good name as it is not just supporting `.dvi` but also `.pdf` output or `.dvi`.

`\AtEndDvi` is the counterpart that was not available in the kernel but only through the package `atenddvi`. It fills the `shipout/lastpage` hook.

Todo: better names? Any suggestions?

1.3 Special commands for use inside the hooks

`\DiscardShipoutBox`
`\shipout_discard_box:`

`\AddToHookNext {shipout/before} {...\DiscardShipoutBox...}`

The `\DiscardShipoutBox` declaration (L3 name `\shipout_discard_box:`) requests that on the next shipout the page box is thrown away instead of being shipped to the `.dvi` or `.pdf` file.

Typical applications wouldn't do this unconditionally, but have some processing logic that decides to use or not to use the page.

Note that if this declaration is used directly in the document it may depend on the placement to which page it applies, given that L^AT_EX output routine is called in an asynchronous manner!

Todo: Once we have a new mark mechanism available we can improve on that and make sure that the declaration applies to the page that contains it.

In the `atbegshi` package there are a number of additional commands for use inside the `shipout/before` hook. They should normally not be needed any more as one can instead simply add code to the hooks `shipout/before`, `shipout/background` or `shipout/foreground`.³ If `atbegshi` gets loaded then those commands become available as public functions with their original names as given below.

³If that assumption turns out to be wrong it would be trivial to change them to public functions (right now they are private).

1.4 Information counters

<code>\ReadOnlyShipoutCounter</code>	<code>\ifnum\ReadOnlyShipoutCounter=...</code>
<code>\g_shipout_readonly_int</code>	<code>\int_use:N \g_shipout_readonly_int % expl3 usage</code>

This integer holds the number of pages shipped out up to now (including the one to be shipped out when inside the output routine). More precisely, it is incremented only after it is clear that a page will be shipped out, i.e., after the `shipout/before` hook (because that might discard the page)!

Just like with the `page` counter its value is only accurate within the output routine. In the body of the document it may be off by one as the output routine is called asynchronously!

Also important: it *must not* be set, only read. There are no provisions to prevent that but if you do, chaos will be the result. To emphasize this fact it is not provided as a \LaTeX counter but as a \TeX counter (i.e., a command), so `\Alph{\ReadOnlyShipoutCounter}` etc, would not work.

<code>totalpages</code>	<code>\arabic{totalpages}</code>
<code>\g_shipout_totalpages_int</code>	<code>\int_use:N \g_shipout_totalpage_int % expl3 usage</code>

In contrast to `\ReadOnlyShipoutCounter`, the `totalpages` counter is a \LaTeX counter and incremented for each shipout attempt including those pages that are discarded for one or the other reason. Again `shipout/before` sees the counter before it is incremented).

Furthermore, while it is incremented for each page, its value is never used by \LaTeX . It can therefore be freely reset or changed by user code, for example, to additionally count a number of pages that are not build by \LaTeX but are added in a later part of the process, e.g., cover pages or picture pages made externally.

Important: as this is a page-related counter its value is only reliable inside the output routine!

1.5 Debugging shipout code

<code>\DebugShipoutOn</code>	<code>\DebugShipoutOn</code>
<code>\DebugShipoutOff</code>	
<code>\shipout_debug_on:</code>	Turn the debugging of shipout code on or off. This displays changes made to the shipout data structures.
<code>\shipout_debug_off:</code>	

Todo: This needs some rationalizing and will probably not stay this way.

2 Emulating commands from other packages

The packages in this section are no longer necessary but as they are used in other packages they are emulated when they are loaded via `\usepackage` or `\RequirePackage`.

2.1 Emulating atbegshi

<code>\AtBeginShipoutUpperLeft</code>	<code>\AddToHook {shipout/before}</code>
<code>\AtBeginShipoutUpperLeftForeground</code>	<code>{... \AtBeginShipoutUpperLeft{<code>}...}</code>

This adds a `picture` environment into the background of the shipout box expecting `<code>` to contain `picture` commands. The same effect can be obtained by simply using kernel features as follows:

```
\AddToHook{shipout/background}{<code>}
```

There is one technical difference: if `\AtBeginShipoutUpperLeft` is used several times each invocation is put into its own box inside the shipout box whereas all `<code>` going into `shipout/background` ends up all in the same box in the order it is added or sorted based on the rules for the hook chunks.

`\AtBeginShipoutUpperLeftForeground` is similar with the difference that the `picture` environment is placed in the foreground. To model it with the kernel functions use the hook `shipout/foreground` instead.

<code>\AtBeginShipoutAddToBox</code>	<code>\AddToHook {shipout/before} {... \AtBeginShipoutAddToBox{<code>}...}</code>
<code>\AtBeginShipoutAddToBoxForeground</code>	

These work like `\AtBeginShipoutUpperLeft` and `\AtBeginShipoutUpperLeftForeground` with the difference that `<code>` is directly placed into an `\hbox` inside the shipout box and not surrounded by a `picture` environment.

To emulate them using `shipout/background` or `shipout/foreground` you may have to wrap `<code>` into a `\put` statement but if the code is not doing any typesetting just adding it to the hook should be sufficient.

<code>\AtBeginShipoutBox</code>	This is the name of the shipout box as <code>atbegshi</code> knows it.
---------------------------------	--

<code>\AtBeginShipoutInit</code>	By default <code>atbegshi</code> delayed its action until <code>\beg{document}</code> . This command was forcing it in an earlier place. With the new concept it does nothing.
----------------------------------	--

<code>\AtBeginShipout</code>	<code>\AtBeginShipout{<code>} ≡ \AddToHook{shipout/before}{<code>}</code>
<code>\AtBeginShipoutNext</code>	<code>\AtBeginShipoutNext{<code>} ≡ \AddToHookNext{shipout/before}{<code>}</code>

This is equivalent to filling the `shipout/before` hook by either using `\AddToHook` or `\AddToHookNext`, respectively.

<code>\AtBeginShipoutFirst</code>	The <code>atbegshi</code> names for <code>\AtBeginDvi</code> and <code>\DiscardShipoutBox</code> .
<code>\AtBeginShipoutDiscard</code>	

2.2 Emulating everyshi

<code>\EveryShipout</code>	<code>\EveryShipout{<code>} ≡ \AddToHook{shipout/before}{<code>}</code>
----------------------------	---

<code>\AtNextShipout</code>	<code>\AtNextShipout{<code>} ≡ \AddToHookNext{shipout/before}{<code>}</code>
-----------------------------	--

2.3 Emulating atenddvi

The atenddvi package implemented only a single command: `\AtEndDvi` and that is now available out of the box.

2.4 Emulating everypage

This page takes over the original `\@begindvi` hook and replaces it. It should be all covered by the hooks offered here (details need checking) and thus could simply use the provided hooks rather than defining its own.

3 The Implementation

```
1 <*2kernel>
2 <@@=shipout>
3 \ExplSyntaxOn
```

3.1 Debugging

```
\g__shipout_debug_bool Holds the current debugging state.
4 \bool_new:N \g__shipout_debug_bool
(End definition for \g__shipout_debug_bool.)
```

```
\shipout_debug_on: Turns debugging on and off by redefining \__shipout_debug:n.
\shipout_debug_off:
  \__shipout_debug:n
\__shipout_debug_gset:
5 \cs_new_eq:NN \__shipout_debug:n \use_none:n
6 \cs_new_protected:Npn \shipout_debug_on:
7 {
8   \bool_gset_true:N \g__shipout_debug_bool
9   \__shipout_debug_gset:
10 }
11 \cs_new_protected:Npn \shipout_debug_off:
12 {
13   \bool_gset_false:N \g__shipout_debug_bool
14   \__shipout_debug_gset:
15 }
16 \cs_new_protected:Npn \__shipout_debug_gset:
17 {
18   \cs_gset_protected:Npx \__shipout_debug:n ##1
19   { \bool_if:NT \g__shipout_debug_bool {##1} }
20 }
(End definition for \shipout_debug_on: and others. These functions are documented on page 5.)
```

```
shipout/before Declaring all hooks for the shipout code.
shipout/foreground
shipout/background
shipout/firstpage
shipout/lastpage
21 \hook_new:n{shipout/before}
22 \hook_new:n{shipout/foreground}
23 \hook_new:n{shipout/background}
24 \hook_new:n{shipout/firstpage}
25 \hook_new:n{shipout/lastpage}
(End definition for shipout/before and others. These functions are documented on page 3.)
```

`\ShipoutBox` The box filled with the page to be shipped out (both L3 and L^AT_EX 2_ε name).
`\l_shipout_box`

```
26 \box_new:N \l_shipout_box
27 \cs_set_eq:NN \ShipoutBox \l_shipout_box
```

(End definition for `\ShipoutBox` and `\l_shipout_box`. These functions are documented on page 2.)

`__shipout_execute:` This is going to be the code run by `\shipout`. The code follows closely the ideas from atbegshi, so not documenting that here for now.

```
28 \cs_set:Npn \__shipout_execute: {
29   \tl_set:Nx \l__shipout_group_level_tl
30     { \int_value:w \tex_currentgrouplevel:D }
31   \tex_afterassignment:D \__shipout_execute_test_level:
32   \tex_setbox:D \l_shipout_box
33 }
```

(End definition for `__shipout_execute:.`)

`\shipout` Overloading the `\shipout` primitive:

```
34 \cs_gset_eq:NN \shipout \__shipout_execute:
```

(End definition for `\shipout`. This function is documented on page 2.)

`\l__shipout_group_level_tl` Helper token list to record the group level at which `__shipout_execute:` is encountered.

```
35 \tl_new:N \l__shipout_group_level_tl
```

(End definition for `\l__shipout_group_level_tl`.)

`_shipout_execute_test_level:` If the group level has changed then we are still constructing `\l_shipout_box` and to continue we need to wait until the current group has finished, hence the `\tex_aftergroup:D`.

```
36 \cs_new:Npn \_shipout_execute_test_level: {
37   \int_compare:nNnT
38     \l__shipout_group_level_tl < \tex_currentgrouplevel:D
39     \tex_aftergroup:D
40   \__shipout_execute_cont:
41 }
```

(End definition for `_shipout_execute_test_level:.`)

`__shipout_execute_cont:` When we have reached this point the shipout box has been processed and is available in `\l_shipout_box` and ready for real ship out (perhaps)..

First we quickly check if it is void (can't happen in the standard L^AT_EX output routine but `\shipout` might be called from a package that has some special processing logic). If it is void we aren't shipping anything out and processing ends.⁴

```
42 \cs_new:Npn \__shipout_execute_cont: {
43   \box_if_empty:NTF \l_shipout_box
44     { \PackageWarning{!tshipout}{Ignoring~ void~ shipout~ box} }
45   {
```

Otherwise we assume that we will ship something and prepare for final adjustments (in particular setting the state of `\protect` while we are running the hook code).

```
46   \bool_gset_false:N \g__shipout_discard_bool
47   \set@typeset@protect
```

⁴In that case we don't reset the deadcycles, that would be up to the OR processing logic to do.

We also store the current shipout box dimension in registers, so that they can be used in the hook code.⁵

```
48     \_shipout_get_box_size:N \l_shipout_box
```

Then we execute the `shipout/before` hook.

```
49     \hook_use:n {shipout/before}
```

In `\g_shipout_totalpages_int` we count all shipout attempts so we increment that counter already here (the other one is incremented later when we know for sure that we do a `\shipout`).

We increment it after running the above hook so that the values for `\g_shipout_totalpages_int` and `\` are in sync while the hook is executed (in the case that `totalpages` isn't manually altered or through discarding pages that is).

```
50     \int_gincr:N \g_shipout_totalpages_int
```

The above hook might contain code that requests the page to be discarded so we now test for it.

```
51     \bool_if:NTF \g_shipout_discard_bool
52     { \PackageInfo{ltshipout}{Completed~ page~ discarded}
53       \bool_gset_false:N \g_shipout_discard_bool
```

As we are discarding the page box and not shipping anything out, we need to do some house cleaning and reset TeX's deadcycles so that it doesn't complain about too many calls to the OR without any shipout.

```
54     \tex_deadcycles:D \c_zero_int
```

Todo: In atbegshi the box was dropped but is that actually needed? Or the resetting of `\protect` to its kernel value?

```
55 %     \group_begin:
56 %     \box_set_eq_drop:NN \l_shipout_box \l_shipout_box
57 %     \group_end:
58 %     \cs_set_eq:NN \protect \exp_not:N
59     }
```

Even if there was no explicit request to discard the box it is possible that the code for the hook `shipout/before` has voided the box (by mistake or deliberately). We therefore test once more but this time make it a warning, because the best practice way is to use the request mechanism.

```
60     { \box_if_empty:NTF \l_shipout_box
61       { \PackageWarning{ltshipout}{
62         Shipout~ box~ was~ voided~ by~ hook, \MessageBreak
63         ignoring~ shipout~ box }
64     }
```

Finally, if the box is still non-empty we are nearly ready to ship it out. First we increment the total page counter so that we can later test if we have reached the final page according to our available information.⁶

```
65     {
66       \int_gincr:N \g_shipout_readonly_int
67       \_shipout_debug:n {
```

⁵This is not really necessary as the code could access them via `\box_ht:N`, etc., but it is perhaps convenient.

⁶Doing that earlier would be wrong because we might end up with the last page counted but discard and then we have no place to add the final objects into the output file.

```

68         \typeout{Absolute~ page~ =~ \int_use:N \g_shipout_readonly_int
69             \space (target:- \@abspage@last)}
70     }

```

Then we store the box sizes again (as they may have changed) and then look at the hooks `shipout/foreground` and `shipout/background`. If either or both are non-empty we add a `picture` environment to the box (in the foreground and or in the background) and execute the hook code inside that environment.

```

71         \__shipout_get_box_size:N \l_shipout_box
72         \@kernel@before@shipout@foreground
73         \hook_if_empty:nF {shipout/foreground}
74             { \__shipout_add_foreground_picture:n
75                 { \hook_use:n {shipout/foreground} } }
76         \hook_if_empty:nF {shipout/background}
77             { \__shipout_add_background_picture:n
78                 { \hook_use:n {shipout/background} } }

```

We then run `__shipout_execute_firstpage_hook`: that adds the content of the hook `shipout/firstpage` to the start of the first page (if non-empty). It is then redefined to do nothing on later pages.

```

79         \__shipout_execute_firstpage_hook:

```

The we check if we have to add the `shipout/lastpage` hook because we have reached the last page. This test will be false for all but one (and hopefully the correct) page.

```

80         \int_compare:nNnT \@abspage@last = \g_shipout_readonly_int
81             { \hook_if_empty:nF {shipout/lastpage}
82                 { \__shipout_debug:n { \typeout{Executing~ lastpage~ hook~
83                     on~ page~ \int_use:N \g_shipout_readonly_int } }
84                     \__shipout_add_foreground_box:n { \UseHook{shipout/lastpage}
85                         \@kernel@after@shipout@lastpage }
86                 }
87             \bool_gset_true:N \g__shipout_lastpage_handled_bool
88         }

```

Finally we run the actual `TEX` primitive for `shipout`. As that will expand delayed `\write` statements inside the page in which protected commands should not expand we first change `\protect` to the appropriate definition for that case.

```

89         \cs_set_eq:NN \protect \exp_not:N
90         \tex_shipout:D \box_use:N \l_shipout_box
91     }
92 }
93 }
94 }

```

(End definition for `__shipout_execute_cont:`.)

```

\@kernel@after@shipout@lastpage
\@kernel@before@shipout@foreground
95 \let\@kernel@after@shipout@lastpage\@empty
96 \let\@kernel@before@shipout@foreground\@empty

```

(End definition for `\@kernel@after@shipout@lastpage` and `\@kernel@before@shipout@foreground`. These functions are documented on page ??.)

`__shipout_execute_firstpage_hook`: This command adds any specials into a box and adds that to the very beginning of the first box shipped out. After that we redefine it to do nothing on later pages.

```

97 \cs_new:Npn \__shipout_execute_firstpage_hook: {

```

Adding something to the beginning means adding it to the background as that layer is done first in the output. Of course that is only needed if the hook actually contains anything.

```

98 \hook_if_empty:nF {shipout/firstpage}
99   { \__shipout_add_background_box:n { \UseHook{shipout/firstpage} } }

```

Once we are here we change the definition to do nothing next time and we also change the command used to implement `\AtBeginDvi` to become a warning and not add further material to a hook that is never used again.

```

100 \cs_gset_eq:NN \__shipout_execute_firstpage_hook: \prg_do_nothing:
101 \cs_gset:Npn \__shipout_add_firstpage_material:Nn ##1 ##2 {
102   \PackageWarning{!tshipout}{
103     First~ page~ is~ already~ shipped~ out,~ ignoring\MessageBreak
104     \string##1 }
105   }
106 }

```

(End definition for `__shipout_execute_firstpage_hook:.`)

`\g__shipout_lastpage_handled_bool` A boolean to signal if we have already handled the `shipout/lastpage` hook.

```

107 \bool_new:N \g__shipout_lastpage_handled_bool

```

(End definition for `\g__shipout_lastpage_handled_bool.`)

`__shipout_add_firstpage_material:Nn` This command adds material to the `shipout/firstpage` hook. It is used in `\AtBeginDvi`, etc. The first argument is the command through which it is called. Initially this is ignored but once we are passed the first page it can be used to generate a warning message mentioning the right user command.

```

108 \cs_new:Npn \__shipout_add_firstpage_material:Nn #1#2 {
109   \AddToHook{shipout/firstpage}{#2}
110 }

```

(End definition for `__shipout_add_firstpage_material:Nn.`)

`__shipout_get_box_size:N` Store the box dimensions in `dimen` registers.

Todo: This could/should perhaps be generalized to set height depth and width given an arbitrary box.

```

111 \cs_new:Npn \__shipout_get_box_size:N #1 {
112   \dim_set:Nn \l_shipout_box_ht_dim { \box_ht:N #1 }
113   \dim_set:Nn \l_shipout_box_dp_dim { \box_dp:N #1 }
114   \dim_set:Nn \l_shipout_box_wd_dim { \box_wd:N #1 }
115   \dim_set:Nn \l_shipout_box_ht_plus_dp_dim { \l_shipout_box_ht_dim +
116                                             \l_shipout_box_dp_dim }
117 }

```

(End definition for `__shipout_get_box_size:N.`)

`\l_shipout_box_ht_dim` And here are the variables set by `__shipout_get_box_size:N`.

```

\l_shipout_box_dp_dim 118 \dim_new:N \l_shipout_box_ht_dim
\l_shipout_box_wd_dim 119 \dim_new:N \l_shipout_box_dp_dim
\l_shipout_box_ht_plus_dp_dim 120 \dim_new:N \l_shipout_box_wd_dim
121 \dim_new:N \l_shipout_box_ht_plus_dp_dim

```

(End definition for `\l_shipout_box_ht_dim` and others. These functions are documented on page 2.)

`\g__shipout_discard_bool` Indicate whether or not the current page box should be discarded

```
122 \bool_new:N \g__shipout_discard_bool
```

(End definition for `\g__shipout_discard_bool`.)

`\l__shipout_tmp_box` We need a box for the background and foreground material and a token register to
`\l__shipout_saved_badness_tl` remember badness settings as we disable them during the buildup below.

```
123 \box_new:N \l__shipout_tmp_box
```

```
124 \tl_new:N \l__shipout_saved_badness_tl
```

(End definition for `\l__shipout_tmp_box` and `\l__shipout_saved_badness_tl`.)

`__shipout_add_background_box:n` In standard L^AT_EX the shipout box is always a `\vbox` but here we allow for other usage as well, in case some package has its own output routine.

```
125 \cs_new:Npn \__shipout_add_background_box:n #1
```

```
126 { \__shipout_get_box_size:N \l__shipout_box
```

But we start testing for a vertical box as that should be the normal case.

```
127 \box_if_vertical:NTF \l__shipout_box
```

```
128 {
```

Save current values of `\vfuzz` and `\vbadness` then change them to allow box manipulations without warnings.

```
129 \tl_set:Nx \l__shipout_saved_badness_tl
```

```
130 { \vfuzz=\the\vfuzz\relax
```

```
131 \vbadness=\the\vbadness\relax }
```

```
132 \vfuzz=\c_max_dim
```

```
133 \vbadness=\c_max_int
```

Then we reconstruct `\l__shipout_box` ...

```
134 \vbox_set_to_ht:Nnn \l__shipout_box \l__shipout_box_ht_plus_dp_dim
```

```
135 {
```

... the material in `#1` is placed into a horizontal box with zero dimensions.

```
136 \hbox_set:Nn \l__shipout_tmp_box
```

```
137 { \l__shipout_saved_badness_tl #1 }
```

```
138 \box_set_wd:Nn \l__shipout_tmp_box \c_zero_dim
```

```
139 \box_set_ht:Nn \l__shipout_tmp_box \c_zero_dim
```

```
140 \box_set_dp:Nn \l__shipout_tmp_box \c_zero_dim
```

The we typeset that box followed by whatever was in `\l__shipout_box` before (unpacked).

```
141 \skip_zero:N \baselineskip
```

```
142 \skip_zero:N \lineskip
```

```
143 \skip_zero:N \lineskiplimit
```

```
144 \box_use:N \l__shipout_tmp_box
```

```
145 \vbox_unpack:N \l__shipout_box
```

The `\kern` ensures that the box has no depth which is afterwards explicitly corrected.

```
146 \kern \c_zero_dim
```

```
147 }
```

```
148 \box_set_ht:Nn \l__shipout_box \l__shipout_box_ht_dim
```

```
149 \box_set_dp:Nn \l__shipout_box \l__shipout_box_dp_dim
```

Todo: The whole boxing maneuver looks a bit like overkill to me, but for the moment I leave.

```

150     \l__shipout_saved_badness_tl
151   }
152   {

```

A horizontal box is handled in a similar way. The last case would be a void box in which case we do nothing hence the missing F branch.

```

153     \box_if_horizontal:NT \l_shipout_box
154   {
155     \tl_set:Nx \l__shipout_saved_badness_tl
156     { \hfuzz=\the\hfuzz\relax
157       \hbadness=\the\hbadness\relax }
158     \hfuzz=\c_max_dim
159     \hbadness=\c_max_int
160     \hbox_set_to_wd:Nnn \l_shipout_box \l_shipout_box_wd_dim
161     {
162       \hbox_set:Nn \l__shipout_tmp_box
163       { \l__shipout_saved_badness_tl #1 }
164       \box_set_wd:Nn \l__shipout_tmp_box \c_zero_dim
165       \box_set_ht:Nn \l__shipout_tmp_box \c_zero_dim
166       \box_set_dp:Nn \l__shipout_tmp_box \c_zero_dim
167       \box_move_up:nn
168       \l_shipout_box_ht_dim
169       { \box_use:N \l__shipout_tmp_box }
170       \hbox_unpack:N \l_shipout_box
171     }
172     \l__shipout_saved_badness_tl
173   }
174 }
175 }

```

(End definition for `__shipout_add_background_box:n`.)

`__shipout_add_foreground_box:n` Foreground boxes are done in the same way, only the order and placement of boxes has to be done differently.

```

176 \cs_new:Npn \__shipout_add_foreground_box:n #1
177 {
178   \box_if_vertical:NTP \l_shipout_box
179   {
180     \tl_set:Nx \l__shipout_saved_badness_tl
181     { \vfuzz=\the\vfuzz\relax
182       \vbadness=\the\vbadness\relax }
183     \vfuzz=\c_max_dim
184     \vbadness=\c_max_int
185     \vbox_set_to_ht:Nnn \l_shipout_box \l_shipout_box_ht_plus_dp_dim
186     {
187       \hbox_set:Nn \l__shipout_tmp_box
188       { \l__shipout_saved_badness_tl #1 }
189       \box_set_wd:Nn \l__shipout_tmp_box \c_zero_dim
190       \box_set_ht:Nn \l__shipout_tmp_box \c_zero_dim
191       \box_set_dp:Nn \l__shipout_tmp_box \c_zero_dim
192       \skip_zero:N \baselineskip
193       \skip_zero:N \lineskip
194       \skip_zero:N \lineskiplimit
195       \vbox_unpack:N \l_shipout_box

```

```

196         \kern -\l_shipout_box_ht_plus_dp_dim
197         \box_use:N \l__shipout_tmp_box
198         \kern \l_shipout_box_ht_plus_dp_dim
199     }
200     \l__shipout_saved_badness_tl
201     \box_set_ht:Nn \l_shipout_box \l_shipout_box_ht_dim
202     \box_set_dp:Nn \l_shipout_box \l_shipout_box_dp_dim
203 }
204 {
205     \box_if_horizontal:NT \l_shipout_box
206     {
207         \tl_set:Nx \l__shipout_saved_badness_tl
208         { \hfuzz=\the\hfuzz\relax
209           \hbadness=\the\hbadness\relax }
210         \hfuzz=\c_max_dim
211         \hbadness=\c_max_int
212         \hbox_set_to_wd:Nnn \l_shipout_box \l_shipout_box_wd_dim
213         {
214             \hbox_unpack:N \l_shipout_box
215             \kern -\box_wd:N \l_shipout_box
216             \hbox_set:Nn \l__shipout_tmp_box
217             { \l__shipout_saved_badness_tl #1 }
218             \box_set_wd:Nn \l__shipout_tmp_box \c_zero_dim
219             \box_set_ht:Nn \l__shipout_tmp_box \c_zero_dim
220             \box_set_dp:Nn \l__shipout_tmp_box \c_zero_dim
221             \box_move_up:nn { \box_ht:N \l_shipout_box }
222             { \box_use:N \l__shipout_tmp_box }
223             \kern \box_wd:N \l_shipout_box
224         }%
225         \l__shipout_saved_badness_tl
226     }
227 }
228 }

```

(End definition for `__shipout_add_foreground_box:n`.)

`\c__shipout_horigin_tl` Two constants holding the offset of the top-left with respect to the media box.
`\c__shipout_vorigin_tl` Setting the constants this way is courtesy of Bruno.

```

229 \tl_gput_right:Nn \@kernel@after@begindocument {
230     \tl_const:Nx \c__shipout_horigin_tl
231     {
232         \cs_if_exist_use:NTF \pdfvariable { horigin }
233         { \cs_if_exist_use:NF \pdfhorigin { 1in } }
234     }
235     \tl_const:Nx \c__shipout_vorigin_tl
236     {
237         \cs_if_exist_use:NTF \pdfvariable { vorigin }
238         { \cs_if_exist_use:NF \pdfvorigin { 1in } }
239     }
240 }

```

(End definition for `\c__shipout_horigin_tl` and `\c__shipout_vorigin_tl`.)

`__shipout_picture_overlay:n` Put the argument into a picture environment that doesn't take up any size and uses 1pt for `\unitlength`.

Todo: Could perhaps be generalized as it might be useful elsewhere. For not it is not.

```

241 \cs_new:Npn \__shipout_picture_overlay:n #1 {
242   \kern -\c__shipout_horigin_tl \scan_stop:
243   \vbox_to_zero:n {
244     \kern -\c__shipout_vorigin_tl \scan_stop:
245     \unitlength 1pt \scan_stop:
246     \hbox_set_to_wd:Nnn \l__shipout_tmp_box \c_zero_dim { \ignorespaces #1 }
247 %   \box_set_wd:Nn \l__shipout_tmp_box \c_zero_dim
248     \box_set_ht:Nn \l__shipout_tmp_box \c_zero_dim
249     \box_set_dp:Nn \l__shipout_tmp_box \c_zero_dim
250     \box_use:N \l__shipout_tmp_box
251     \tex_vss:D
252   }
253 }

```

(End definition for __shipout_picture_overlay:n.)

`__shipout_add_background_picture:n` Put a `picture` env in the background of the shipout box with its reference point in the top-left corner.

```

254 \cs_new:Npn \__shipout_add_background_picture:n #1 {
255   \__shipout_add_background_box:n { \__shipout_picture_overlay:n {#1} }
256 }

```

(End definition for __shipout_add_background_picture:n.)

`__shipout_add_foreground_picture:n` Put a `picture` env in the foreground of the shipout box with its reference point in the top-left corner.

```

257 \cs_new:Npn \__shipout_add_foreground_picture:n #1 {
258   \__shipout_add_foreground_box:n { \__shipout_picture_overlay:n {#1} }
259 }

```

(End definition for __shipout_add_foreground_picture:n.)

`\shipout_discard:` Request that the next shipout box should be discarded. At the moment this is just setting a boolean, but we may want to augment this behavior that the position of the call is taken into account (in case L^AT_EX looks ahead and is not using the position for on the next page).

```

260 \cs_new:Npn \shipout_discard: {
261   \bool_gset_true:N \g__shipout_discard_bool
262 }

```

(End definition for \shipout_discard:. This function is documented on page ??.)

3.2 Handling the end of job hook

At the moment this is partly solved by using the existing hooks. But rather than putting the code into these hooks it should be moved to the right place directly as we shouldn't prefill hooks with material unless it needs to interact with other code.

`\g_shipout_readonly_int` We count every shipout activity that makes a page (but not those that are discarded) in order to know how many pages got produced.

`\ReadonlyShipoutCounter`

```

263 \int_new:N \g_shipout_readonly_int

```

For L^AT_εX 2_ε it is available as a command (i.e., a T_εX counter only).

```
264 \cs_new_eq:NN \ReadOnlyShipoutCounter \g_shipout_readonly_int
```

(End definition for `\g_shipout_readonly_int` and `\ReadOnlyShipoutCounter`. These functions are documented on page 5.)

`\g_shipout_totalpages_int` We count every shipout attempt (even those that are discarded) in tis counter. It is not used in the code but may get used in user code.

```
265 \int_new:N \g_shipout_totalpages_int
```

For L^AT_εX 2_ε this is offered as a L^AT_εX counter so can be easily typeset inside the output routine to display things like “`\thepage/\thetotalpages`”, etc.

```
266 \cs_new_eq:NN \c@totalpages \g_shipout_totalpages_int
```

```
267 \cs_new:Npn \thetotalpages { \arabic{totalpages} }
```

(End definition for `\g_shipout_totalpages_int` and `\c@totalpages`. These functions are documented on page 5.)

`\@abspage@last` In `\@abspage@last` record the number of pages from the last run. This is written to the `.aux` and this way made available to the next run. In case there is no `.aux` file or the statement is missing from it we initialize it with the largest possible number in T_εX. We use this as the default because then we are inserting the `shipout/lastpage` on the last page (or after the last page but not on page 1 for a multipage document).

```
268 \xdef\@abspage@last{\number\maxdimen}
```

(End definition for `\@abspage@last`. This function is documented on page ??.)

`\enddocument` Instead of using the hooks `enddocument` and `enddocument/afterlastpage` we add this code to private kernel hooks to be 100% when it is executed and to avoid cluttering the hooks with data that is always there.

Inside `\enddocument` there is a `\clearpage`. Just before that we execute this code here. There is a good change that we are on the last page. Therefore, if we don't know the value from the last run, we assume that the current page is the right one. So we set `\@abspage@last` and as a result the next shipout will run the `shipout/lastpage` code. Of course, if there are floats that still need a placement this guess will be wrong but then rerunning the document will give us the correct value next time around.

```
\@kernel@after@enddocument 269 \g@addto@macro \@kernel@after@enddocument {
270   \int_compare:nNnT \@abspage@last = \maxdimen
271   {
```

We use L^AT_εX 2_ε coding as `\@abspage@last` is not an L3 name.

```
272   \xdef\@abspage@last{ \int_eval:n {\g_shipout_readonly_int + 1} }
273   }
274 }
```

`\@kernel@after@enddocument@afterlastpage`

Once the `\clearpage` has done its work inside `\enddocument` we know for sure how many pages this document has, so we record that in the `.aux` file for the next run.

```
275 \g@addto@macro \@kernel@after@enddocument@afterlastpage {
```


There is one special case: If no output is produced then there is no point in a) recording the number as 0 will never match the page number of a real page and b) adding an extra page to ran the `shipout/lastpage` is pointless as well (as it would remain forever). So we test for this and run the code only if there have been pages.

```
276 \int_compare:nNnF \g_shipout_readonly_int = 0
277 {
```

This ends up in the `.aux` so we use L^AT_EX 2_ε names here.

Todo: this needs an interface for `\nofiles` in `expl3`, doesn't at the moment!

```
278 \if@filesw
279 \iow_now:Nx \@auxout {
280 \gdef\string\@abspage@last {\int_use:N \g_shipout_readonly_int}}
281 \fi
```

But we may have guessed wrongly earlier and we still have to run the `shipout/lastpage` even though there is no page to place it into. If that is the case we make a trivial extra page and put it there. This temporary page will then vanish again on the next run but helps to keep pdf viewers happy.

```
282 \bool_if:NF \g__shipout_lastpage_handled_bool
283 { \tex_shipout:D\vbox to\textheight{
284 \hbox{\UseHook{shipout/lastpage} \@kernel@after@shipout@lastpage }
285 }
```

This extra page could be totally empty except for the hook content, but to help the user understanding why it is there we put some text into it.

```
285 \__shipout_excuse_extra_page:
286 \null
287 }
```

At this point we should also signal to L^AT_EX that a rerun is necessary so that an appropriate message can be shown on the terminal.

Todo: integrate warning “Rerun ...”

```
288 }
289 }
290 }
```

(End definition for `\enddocument`, `\@kernel@after@enddocument`, and `\@kernel@after@enddocument@afterlastpage`. These functions are documented on page ??.)

`__shipout_excuse_extra_page:` Say mea culpa ...

```
291 \cs_new:Npn \__shipout_excuse_extra_page: {
292 \vfil
293 \begin{center}
294 \bfseries Temporary~ page!
295 \end{center}
296 \LaTeX{}~ was~ unable~ to~ guess~ the~ total~ number~ of~ pages~
297 correctly.~ ~ As~ there~ was~ some~ unprocessed~ data~ that~
298 should~ have~ been~ added~ to~ the~ final~ page~ this~ extra~
299 page~ has~ been~ added~ to~ receive~ it.
300 \par
301 If~ you~ rerun~ the~ document~ (without~ altering~ it)~ this~
302 surplus~ page~ will~ go~ away,~ because~ \LaTeX{}~ now~ knows~
303 how~ many~ pages~ to~ expect~ for~ this~ document.
304 \vfil
305 }
```

(End definition for `_shipout_excuse_extra_page:.`)

4 Legacy L^AT_EX 2_ε interfaces

`\DiscardShipoutBox` Request that the next shipout box is to be discarded.

```
306 \cs_new_eq:NN \DiscardShipoutBox \shipout_discard:
```

(End definition for `\DiscardShipoutBox`. This function is documented on page 4.)

`\AtBeginDvi`

```
307 \renewcommand \AtBeginDvi {\_shipout_add_firstpage_material:Nn \AtBeginDvi}
```

(End definition for `\AtBeginDvi`. This function is documented on page 4.)

`\DebugShipoutOn`

`\DebugShipoutOff`

```
308 \cs_new_eq:NN \DebugShipoutOn \shipout_debug_on:
```

```
309 \cs_new_eq:NN \DebugShipoutOff \shipout_debug_off:
```

(End definition for `\DebugShipoutOn` and `\DebugShipoutOff`. These functions are documented on page 5.)

5 Package emulation for compatibility

5.1 Package `atbegshi` emulation

`\AtBeginShipoutBox`

```
310 \cs_new_eq:NN \AtBeginShipoutBox \ShipoutBox
```

(End definition for `\AtBeginShipoutBox`. This function is documented on page 6.)

`\AtBeginShipoutInit` Compatibility only, we aren't delaying ...

```
311 \cs_set_eq:NN \AtBeginShipoutInit \@empty
```

(End definition for `\AtBeginShipoutInit`. This function is documented on page 6.)

`\AtBeginShipout` Filling hooks

`\AtBeginShipoutNext`

```
312 \newcommand \AtBeginShipout {\AddToHook{shipout/before}}
```

```
313 \newcommand \AtBeginShipoutNext {\AddToHookNext{shipout/before}}
```

(End definition for `\AtBeginShipout` and `\AtBeginShipoutNext`. These functions are documented on page 6.)

`\AtBeginShipoutFirst` Slightly more complex as we need to know the name of the command under which the `shipout/firstpage` hook is filled.

```
314 \newcommand \AtBeginShipoutFirst {\_shipout_add_firstpage_material:Nn \AtBeginShipoutFirst}
```

(End definition for `\AtBeginShipoutFirst`. This function is documented on page 6.)

This is somewhat different from the original where `\ShipoutBoxHeight` etc. only holds the `\the\ht<box>` value. This may have some implications in some use cases and if that is a problem then it might need changing.

```
315 \cs_new:Npn \ShipoutBoxHeight { \dim_use:N \l_shipout_box_ht_dim }
```

```
316 \cs_new:Npn \ShipoutBoxDepth { \dim_use:N \l_shipout_box_dp_dim }
```

```
317 \cs_new:Npn \ShipoutBoxWidth { \dim_use:N \l_shipout_box_wd_dim }
```

`\AtBeginShipoutDiscard` Just a different name.
`318 \cs_new_eq:NN \AtBeginShipoutDiscard \DiscardShipoutBox`
(End definition for \AtBeginShipoutDiscard. This function is documented on page 6.)

`\AtBeginShipoutAddToBox` We don't expose them.
`\AtBeginShipoutAddToBoxForeground` `319 \cs_new_eq:NN \AtBeginShipoutAddToBox __shipout_add_background_box:n`
`\AtBeginShipoutUpperLeft` `320 \cs_new_eq:NN \AtBeginShipoutAddToBoxForeground __shipout_add_foreground_box:n`
`\AtBeginShipoutUpperLeftForeground` `321 \cs_new_eq:NN\AtBeginShipoutUpperLeft __shipout_add_background_picture:n`
`322 \cs_new_eq:NN\AtBeginShipoutUpperLeftForeground __shipout_add_foreground_picture:n`
(End definition for \AtBeginShipoutAddToBox and others. These functions are documented on page 6.)

We prevent the package atbegshi from loading:
`323 \expandafter\cs_set_eq:NN\csname ver@atbegshi.sty\endcsname\fmtversion`
hyperref code (and ltxcmds) doesn't understand 2020-10-01 and thinks it is before 1994, so for now ...
`324 \@namedef {ver@atbegshi.sty}{2020/10/01}`

5.2 Package everyshi emulation

`\EveryShipout` This package has only two public commands to simulating it is easy:
`\AtNextShipout` `325 \cs_new_eq:NN\EveryShipout\AtBeginShipout`
`326 \cs_new_eq:NN\AtNextShipout\AtBeginShipoutNext`
(End definition for \EveryShipout and \AtNextShipout. These functions are documented on page 6.)
`327 \expandafter\cs_set_eq:NN\csname ver@everyshi.sty\endcsname\fmtversion`
`328 \@namedef {ver@everyshi.sty}{2020/10/01}`

5.3 Package atenddvi emulation

`\AtEndDvi` This package has only one public command to simulating it is easy:
`329 \cs_new:Npn \AtEndDvi {\AddToHook{shipout/lastpage}}`
(End definition for \AtEndDvi. This function is documented on page 4.)
`330 \expandafter\cs_set_eq:NN\csname ver@atenddvi.sty\endcsname\fmtversion`
`331 \@namedef {ver@atenddvi.sty}{2020/10/01}`
`332 \ExplSyntaxOff`
`333 </2kernel>`

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

- A**
- `\AddToHook` 5, 5, 5, 5, 109, 312, 329
 - `\AddToHookNext` 3, 5, 5, 313
 - `\Alph` 4
 - `\arabic` 4, 267
 - `\AtBeginDvi` 3, 5, 10, 10, 307
 - `\AtBeginShipout` 5, 312, 325
 - `\AtBeginShipoutAddToBox` 5, 319
 - `\AtBeginShipoutAddToBoxForeground` 5, 319
 - `\AtBeginShipoutBox` 5, 310
 - `\AtBeginShipoutDiscard` 5, 318
 - `\AtBeginShipoutFirst` 5, 314
 - `\AtBeginShipoutInit` 5, 311
 - `\AtBeginShipoutNext` 5, 312, 326
 - `\AtBeginShipoutUpperLeft` 5, 5, 319
 - `\AtBeginShipoutUpperLeftForeground` .
..... 5, 5, 319
 - `\AtEndDvi` 3, 6, 329
 - `\AtNextShipout` 5, 325
- B**
- `\baselineskip` 141, 192
 - `\begin` 293
 - `\bfseries` 294
 - bool commands:
 - `\bool_gset_false:N` 13, 46, 53
 - `\bool_gset_true:N` 8, 87, 261
 - `\bool_if:NTF` 19, 51, 282
 - `\bool_new:N` 4, 107, 122
 - box commands:
 - `\box_dp:N` 113
 - `\box_ht:N` 8, 112, 221
 - `\box_if_empty:NTF` 43, 60
 - `\box_if_horizontal:NTF` 153, 205
 - `\box_if_vertical:NTF` 127, 178
 - `\box_move_up:nn` 167, 221
 - `\box_new:N` 26, 123
 - `\box_set_dp:Nn`
..... 140, 149, 166, 191, 202, 220, 249
 - `\box_set_eq_drop:NN` 56
 - `\box_set_ht:Nn`
..... 139, 148, 165, 190, 201, 219, 248
 - `\box_set_wd:Nn` 138, 164, 189, 218, 247
 - `\box_use:N` . 90, 144, 169, 197, 222, 250
 - `\box_wd:N` 114, 215, 223
 - `\l_shipout_box` 1,
2, 7, 7, 11, 11, 26, 32, 43, 48, 56,
60, 71, 90, 126, 127, 134, 145, 148,
149, 153, 160, 170, 178, 185, 195,
201, 202, 205, 212, 214, 215, 221, 223
- C**
- `\clearpage` 15, 15
 - cs commands:
 - `\cs_gset:Npn` 101
 - `\cs_gset_eq:NN` 34, 100
 - `\cs_gset_protected:Npx` 18
 - `\cs_if_exist_use:NTF` 232, 233, 237, 238
 - `\cs_new:Npn` 36,
42, 97, 108, 111, 125, 176, 241, 254,
257, 260, 267, 291, 315, 316, 317, 329
 - `\cs_new_eq:NN`
..... 5, 264, 266, 306, 308, 309,
310, 318, 319, 320, 321, 322, 325, 326
 - `\cs_new_protected:Npn` 6, 11, 16
 - `\cs_set:Npn` 28
 - `\cs_set_eq:NN`
..... 27, 58, 89, 311, 323, 327, 330
 - `\csname` 323, 327, 330
- D**
- `\DebugShipoutOff` 4, 308
 - `\DebugShipoutOn` 4, 308
 - dim commands:
 - `\dim_new:N` 118, 119, 120, 121
 - `\dim_set:Nn` 112, 113, 114, 115
 - `\dim_use:N` 315, 316, 317
 - `\c_max_dim` 132, 158, 183, 210
 - `\c_zero_dim` 138, 139,
140, 146, 164, 165, 166, 189, 190,
191, 218, 219, 220, 246, 247, 248, 249
 - `\DiscardShipoutBox` 2, 3, 5, 306, 318
- E**
- `\end` 295
 - `\endcsname` 323, 327, 330
 - `\enddocument` 15, 15, 269
 - `\EveryShipout` 5, 325
 - exp commands:
 - `\exp_not:N` 58, 89
 - `\expandafter` 323, 327, 330
 - `\ExplSyntaxOff` 332
 - `\ExplSyntaxOn` 3
- F**
- `\fi` 281
 - `\fmtversion` 323, 327, 330

G	
<code>\gdef</code>	280
group commands:	
<code>\group_begin:</code>	55
<code>\group_end:</code>	57
H	
<code>\hbadness</code>	157, 159, 209, 211
<code>\hbox</code>	1, 2, 5, 284
hbox commands:	
<code>\hbox_set:Nn</code>	136, 162, 187, 216
<code>\hbox_set_to_wd:Nnn</code>	160, 212, 246
<code>\hbox_unpack:N</code>	170, 214
<code>\hfuzz</code>	156, 158, 208, 210
hook commands:	
<code>\hook_if_empty:nTF</code>	73, 76, 81, 98
<code>\hook_new:n</code>	21, 22, 23, 24, 25
<code>\hook_use:n</code>	49, 75, 78
I	
<code>\ifnum</code>	4
<code>\ignorespaces</code>	246
int commands:	
<code>\int_compare:nNnTF</code> ..	37, 80, 270, 276
<code>\int_eval:n</code>	272
<code>\int_gincr:N</code>	50, 66
<code>\int_new:N</code>	263, 265
<code>\int_use:N</code>	4, 4, 68, 83, 280
<code>\int_value:w</code>	30
<code>\c_max_int</code>	133, 159, 184, 211
<code>\c_zero_int</code>	54
iow commands:	
<code>\iow_now:Nn</code>	279
K	
<code>\kern</code> .	11, 146, 196, 198, 215, 223, 242, 244
L	
<code>\LaTeX</code>	296, 302
<code>\let</code>	95, 96
<code>\lineskip</code>	142, 193
<code>\lineskiplimit</code>	143, 194
M	
<code>\maxdimen</code>	268, 270
<code>\MessageBreak</code>	62, 103
N	
<code>\newcommand</code>	312, 313, 314
<code>\nofiles</code>	16
<code>\null</code>	286
<code>\number</code>	268
P	
<code>\PackageInfo</code>	52
<code>\PackageWarning</code>	44, 61, 102
<code>\par</code>	300
<code>\pdfhorigin</code>	233
<code>\pdfvariable</code>	232, 237
<code>\pdfvorigin</code>	238
prg commands:	
<code>\prg_do_nothing:</code>	100
<code>\protect</code>	7, 8, 9, 58, 89
<code>\put</code>	2, 5
R	
<code>\ReadOnlyShipoutCounter</code>	4
<code>\ReadonlyShipoutCounter</code>	4, 4, 263
<code>\relax</code> 130, 131, 156, 157, 181, 182, 208, 209	
<code>\renewcommand</code>	307
<code>\RequirePackage</code>	4
S	
scan commands:	
<code>\scan_stop:</code>	242, 244, 245
<code>\shipout</code>	1, 1, 3, 7, 7, 7, 8, 34
shipout commands:	
<code>\l_shipout_box_dp_dim</code>	
.....	1, 113, 116, 118, 149, 202, 316
<code>\l_shipout_box_ht_dim</code>	
.....	1, 112, 115, 118, 148, 168, 201, 315
<code>\l_shipout_box_ht_plus_dp_dim</code> ..	
.....	1, 115, 118, 134, 185, 196, 198
<code>\l_shipout_box_wd_dim</code>	
.....	1, 114, 118, 160, 212, 317
<code>\shipout_debug_off:</code>	4, 5, 309
<code>\shipout_debug_on:</code>	4, 5, 308
<code>\shipout_discard:</code>	260, 306
<code>\shipout_discard_box:</code>	3
<code>\g_shipout_readonly_int</code>	
.....	4, 66, 68, 80, 83, 263, 272, 276, 280
<code>\g_shipout_totalpage_int</code>	4
<code>\g_shipout_totalpages_int</code> 4, 8, 50, 265	
shipout internal commands:	
<code>__shipout_add_background_box:n</code> .	
.....	99, 125, 255, 319
<code>__shipout_add_background_-</code>	
<code>picture:n</code>	77, 254, 321
<code>__shipout_add_firstpage_-</code>	
<code>material:Nn</code>	101, 108, 307, 314
<code>__shipout_add_foreground_box:n</code> .	
.....	84, 176, 258, 320
<code>__shipout_add_foreground_-</code>	
<code>picture:n</code>	74, 257, 322
<code>__shipout_debug:n</code>	5, 6, 67, 82
<code>\g__shipout_debug_bool</code> ...	4, 8, 13, 19
<code>__shipout_debug_gset:</code>	5
<code>\g__shipout_discard_bool</code>	
.....	46, 51, 53, 122, 261

