

The `ltshipout` package*

Frank Mittelbach, L^AT_EX Project Team

January 9, 2021

Contents

1	Introduction	1
1.1	Overloading the <code>\shipout</code> primitive	2
1.2	Provided hooks	3
1.3	Legacy L ^A T _E X commands	4
1.4	Special commands for use inside the hooks	4
1.5	Information counters	5
1.6	Debugging shipout code	5
2	Emulating commands from other packages	5
2.1	Emulating <code>atbegshi</code>	6
2.2	Emulating <code>everyshi</code>	7
2.3	Emulating <code>atenddvi</code>	7
2.4	Emulating <code>everypage</code>	7
3	The Implementation	7
3.1	Debugging	8
3.2	Handling the end of job hook	17
4	Legacy L^AT_EX 2_ε interfaces	20
5	Internal commands needed elsewhere	20
6	Package emulation for compatibility	22
6.1	Package <code>atenddvi</code> emulation	22
6.2	Package <code>atbegshi</code> emulation	22
6.3	Package <code>everyshi</code> emulation	24
	Index	25

*This package has version v1.0d dated 2020/12/06, © L^AT_EX Project.

1 Introduction

The code provides an interface to the `\shipout` primitive of T_EX 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

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`
`\l_shipout_box`

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

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

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.

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 that is shipped out (i.e., not discarded at the last minute). It should only contain `\special` or similar commands needed to direct post processors handling the `.dvi` or `.pdf` output.³

shipout/lastpage The corresponding hook to add `\specials` at the very end of the output file. It is only executed on the very last page of the output file — or rather on the page that `LATEX` believes is the last one.

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:

³In `LATEX 2ε` that was already existing, but implemented using a box register with the name `\@begindvibox`.

<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.

1.3 Legacy L^AT_EX commands

`\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 `.xdv`.

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

As these two wrappers have been available for a long time we continue offering them. However, for new code we suggest using the high-level hook management commands directly instead of “randomly-named” wrappers. This will lead to code that is easier to understand and to maintain. For this reason we do not provide any other wrapper commands for the above hooks in the kernel.

1.4 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`.⁴ If `atbegshi` gets loaded then those commands become available as public functions with their original names as given below.

⁴If that assumption turns out to be wrong it would be trivial to change them to public functions (right now they are private).

1.5 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!

<code>\PreviousTotalPages</code>	<code>\thetotalpages/\PreviousTotalPages</code>
----------------------------------	---

Command that expands to the number of total pages from the previous run. If there was no previous run or if used in the preamble it expands to 0. Note that this is a command and not a counter, so in order to display the number in, say, Roman numerals you have to assign its value to a counter and then use `\Roman` on that counter.

1.6 Debugging shipout code

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

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

2 Emulating commands from other packages

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

Please note that the emulation only happens if the package is explicitly requested, i.e., the commands documented below are not automatically available in the L^AT_EX kernel! If you write a new package we suggest to use the appropriate kernel hooks directly instead of loading the emulation.

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>\begin{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

The everyshi package is providing commands to run arbitrary code just before the shipout starts. One point of difference: in the new shipout hooks the page is available as `\ShipoutBox` for inspection of change, one should not manipulate box 255 directly inside `shipout/before`, so old code doing this would change to use `\ShipoutBox` instead of 255 or `\@cclv`.

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

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

However, most use cases for everyshi are attempts to put some picture or text into the background or foreground of the page and that can be done today simply by using the `shipout/background` and `shipout/foreground` hooks without any need to coding.

2.3 Emulating atenddvi

The atenddvi package implemented only a single command: `\AtEndDvi` and that is now available out of the box so the emulation makes the package a no-op.

2.4 Emulating everypage

This package patched the original `\@beginDvi` hook and replaced it with its own version. Its functionality is now covered by the hooks offered by the kernel so that there is no need for such patching any longer.

`\AddEverypageHook` `\AddEverypageHook{<code>} ≡`
`\AddToHook{shipout/background}{\put(1in,-1in){<code>}}`

`\AddEverypageHook` is adding something into the background of every page at a position of 1in to the right and 1in down from the top left corner of the page. By using the kernel hook directly you can put your material directly to the right place, i.e., use other coordinates in the `\put` statement above.

`\AddThispageHook` `\AddThispageHook{<code>} ≡`
`\AddToHookNext{shipout/background}{\put(1in,-1in){<code>}}`

The `\AddThispageHook` wrapper is similar but uses `\AddToHookNext`.

3 The Implementation

1 `<@@=shipout>`

At the moment the whole module rolls back in one go, but if we make any modifications in later releases this will then need splitting.

2 `<*2kernel | latexrelease>`

3 `<latexrelease>\IncludeInRelease{2020/10/01}%`

4 `<latexrelease> \{shipout\}{Hook management (shipout)}%`

5 `\ExplSyntaxOn`

3.1 Debugging

`\g__shipout_debug_bool` Holds the current debugging state.

```

6 \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:`

```

7 \cs_new_eq:NN \__shipout_debug:n \use_none:n
8 \cs_new_protected:Npn \shipout_debug_on:
9   {
10    \bool_gset_true:N \g__shipout_debug_bool
11    \__shipout_debug_gset:
12   }
13 \cs_new_protected:Npn \shipout_debug_off:
14   {
15    \bool_gset_false:N \g__shipout_debug_bool
16    \__shipout_debug_gset:
17   }
18 \cs_new_protected:Npn \__shipout_debug_gset:
19   {
20    \cs_gset_protected:Npx \__shipout_debug:n ##1
21    { \bool_if:NT \g__shipout_debug_bool {##1} }
22   }

```

(End definition for \shipout_debug_on: and others. These functions are documented on page 5.)

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

`\l_shipout_box`

```

23 \box_new:N \l_shipout_box
24 \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.

```

25 \cs_set_protected:Npn \__shipout_execute: {
26   \tl_set:Nx \l__shipout_group_level_tl
27     { \int_value:w \tex_currentgrouplevel:D }
28   \tex_afterassignment:D \__shipout_execute_test_level:
29   \tex_setbox:D \l_shipout_box
30 }

```

(End definition for __shipout_execute:.)

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

```

31 \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.

```

32 \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`.

```

33 \cs_new:Npn \_shipout_execute_test_level: {
34   \int_compare:nNt
35     \l_shipout_group_level_tl < \tex_currentgrouplevel:D
36     \tex_aftergroup:D
37   \_shipout_execute_cont:
38 }

```

(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.⁵

```

39 \cs_new:Npn \_shipout_execute_cont: {
40   \box_if_empty:NTF \l_shipout_box
41     { \PackageWarning{ltshipout}{Ignoring~ void~ shipout~ box} }
42     {

```

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). We also save the current `\protect` state to restore it later.

```

43     \bool_gset_false:N \g_shipout_discard_bool
44     \cs_set_eq:NN \_shipout_saved_protect: \protect
45     \set@typeset@protect

```

We also store the current shipout box dimension in registers, so that they can be used in the hook code.⁶

```

46     \_shipout_get_box_size:N \l_shipout_box

```

Then we execute the `shipout/before` hook.

```

47     \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 `\g_shipout_readonly_int` are in sync while the hook is executed (in the case that `totalpages` isn't manually altered or through discarding pages that is).

```

48     \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.

```

49     \bool_if:NTF \g_shipout_discard_bool
50       { \PackageInfo{ltshipout}{Completed~ page~ discarded}
51         \bool_gset_false:N \g_shipout_discard_bool

```

⁵In that case we don't reset the deadcycles, that would be up to the OR processing logic to do.

⁶This is not really necessary as the code could access them via `\box_ht:N`, etc., but it is perhaps convenient.

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

```
52         \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?

```
53 %         \group_begin:
54 %         \box_set_eq_drop:NN \l_shipout_box \l_shipout_box
55 %         \group_end:
56 %         \cs_set_eq:NN \protect \exp_not:N
57     }
```

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.

```
58     { \box_if_empty:NTF \l_shipout_box
59       { \PackageWarning{!tshipout}{
60         Shipout~ box~ was~ voided~ by~ hook,\MessageBreak
61         ignoring~ shipout~ box }
62     }
```

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.⁷

```
63     {
64       \int_gincr:N \g_shipout_readonly_int
65       \__shipout_debug:n {
66         \typeout{Absolute~ page~ =~ \int_use:N \g_shipout_readonly_int
67           \space (target:~ \@abspage@last)}
68     }
```

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.

```
69         \__shipout_get_box_size:N \l_shipout_box
70         \hook_if_empty:nF {shipout/foreground}
71         { \__shipout_add_foreground_picture:n
72           { \hook_use:n {shipout/foreground} } }
```

If there is no user hook, there might still code in the kernel hook.

```
73         \bool_lazy_and:nnF
74         { \hook_if_empty_p:n {shipout/background} }
75         { \tl_if_empty_p:N \@kernel@before@shipout@background }
76         { \__shipout_add_background_picture:n
77           { \@kernel@before@shipout@background
78             \hook_use:n {shipout/background} } }
```

⁷Doing 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.

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 or the corresponding kernel 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 { \bool_lazy_and:nnF
82 { \hook_if_empty_p:n {shipout/lastpage} }
83 { \tl_if_empty_p:N \@kernel@after@shipout@lastpage }
84 { \__shipout_debug:n { \typeout{Executing~ lastpage~ hook~
85 on~ page~ \int_use:N \g_shipout_readonly_int } } }
86 \__shipout_add_foreground_box:n { \UseHook{shipout/lastpage}
87 \@kernel@after@shipout@lastpage }
88 }
89 \bool_gset_true:N \g__shipout_lastpage_handled_bool
90 }
```

Finally we run the actual T_EX 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.

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

Restore the value of `\protect` in case `\shipout` is called outside of the output routine (where it is automatically restored because of the implicit group).

```
95 \cs_set_eq:NN \protect \__shipout_saved_protect:
96 }
97 }
```

(End definition for __shipout_execute_cont:.)

`__shipout_saved_protect`: Remember the current `\protect` state.

```
98 \cs_new_eq:NN \__shipout_saved_protect: \protect
```

(End definition for __shipout_saved_protect:.)

`shipout/before` Declaring all hooks for the shipout code.

```
99 \hook_new:n{shipout/before}
100 \hook_new:n{shipout/foreground}
101 \hook_new:n{shipout/background}
102 \hook_new:n{shipout/firstpage}
103 \hook_new:n{shipout/lastpage}
```

(End definition for shipout/before and others. These functions are documented on page 3.)

`\@kernel@after@shipout@lastpage` And here are the internal kernel hooks going before or after the public ones where needed.

```
\@kernel@before@shipout@background 104 \let\@kernel@after@shipout@lastpage\empty
105 \let\@kernel@before@shipout@background\empty
```

(End definition for \@kernel@after@shipout@lastpage and \@kernel@before@shipout@background. 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.

```
106 \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.

```
107   \hook_if_empty:nF {shipout/firstpage}
108     { \_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.

```
109   \cs_gset_eq:NN \_shipout_execute_firstpage_hook: \prg_do_nothing:
110   \cs_gset:Npn \_shipout_add_firstpage_material:Nn ##1 ##2 {
111     \@latex@warning{
112       First~ page~ is~ already~ shipped~ out,~ ignoring\MessageBreak
113       \string##1 }
114   }
115 }
```

(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.

```
116 \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.

```
117 \cs_new:Npn \_shipout_add_firstpage_material:Nn #1#2 {
118   \AddToHook{shipout/firstpage}{#2}
119 }
```

(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.

```
120 \cs_new:Npn \_shipout_get_box_size:N #1 {
121   \dim_set:Nn \l_shipout_box_ht_dim { \box_ht:N #1 }
122   \dim_set:Nn \l_shipout_box_dp_dim { \box_dp:N #1 }
123   \dim_set:Nn \l_shipout_box_wd_dim { \box_wd:N #1 }
124   \dim_set:Nn \l_shipout_box_ht_plus_dp_dim { \l_shipout_box_ht_dim +
125                                             \l_shipout_box_dp_dim }
126 }
```

(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 127 \dim_new:N \l_shipout_box_ht_dim  
\l_shipout_box_wd_dim 128 \dim_new:N \l_shipout_box_dp_dim  
\l_shipout_box_ht_plus_dp_dim 129 \dim_new:N \l_shipout_box_wd_dim  
130 \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

```
131 \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.

```
132 \box_new:N \l_shipout_tmp_box  
133 \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 are allow for other usage as well, in case some package has its own output routine.

```
134 \cs_new:Npn \_shipout_add_background_box:n #1  
135 { \_shipout_get_box_size:N \l_shipout_box
```

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

```
136 \box_if_vertical:NTF \l_shipout_box  
137 {
```

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

```
138 \tl_set:Nx \l_shipout_saved_badness_tl  
139 { \vfuzz=\the\vfuzz\relax  
140 \vbadness=\the\vbadness\relax }  
141 \vfuzz=\c_max_dim  
142 \vbadness=\c_max_int
```

Then we reconstruct `\l_shipout_box` ...

```
143 \vbox_set_to_ht:Nnn \l_shipout_box \l_shipout_box_ht_plus_dp_dim  
144 {
```

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

```
145 \hbox_set:Nn \l_shipout_tmp_box  
146 { \l_shipout_saved_badness_tl #1 }  
147 \box_set_wd:Nn \l_shipout_tmp_box \c_zero_dim  
148 \box_set_ht:Nn \l_shipout_tmp_box \c_zero_dim  
149 \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).

```
150 \skip_zero:N \baselineskip  
151 \skip_zero:N \lineskip  
152 \skip_zero:N \lineskiplimit  
153 \box_use:N \l_shipout_tmp_box  
154 \vbox_unpack:N \l_shipout_box
```

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

```

155     \kern \c_zero_dim
156   }
157   \box_set_ht:Nn \l_shipout_box \l_shipout_box_ht_dim
158   \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.

```

159   \l__shipout_saved_badness_tl
160 }
161 {

```

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.

```

162   \box_if_horizontal:NT \l_shipout_box
163   {
164     \tl_set:Nx \l__shipout_saved_badness_tl
165     { \hfuzz=\the\hfuzz\relax
166       \hbadness=\the\hbadness\relax }
167     \hfuzz=\c_max_dim
168     \hbadness=\c_max_int
169     \hbox_set_to_wd:Nnn \l_shipout_box \l_shipout_box_wd_dim
170     {
171       \hbox_set:Nn \l__shipout_tmp_box
172       { \l__shipout_saved_badness_tl #1 }
173       \box_set_wd:Nn \l__shipout_tmp_box \c_zero_dim
174       \box_set_ht:Nn \l__shipout_tmp_box \c_zero_dim
175       \box_set_dp:Nn \l__shipout_tmp_box \c_zero_dim
176       \box_move_up:nn
177         \l_shipout_box_ht_dim
178       { \box_use:N \l__shipout_tmp_box }
179       \hbox_unpack:N \l_shipout_box
180     }
181     \l__shipout_saved_badness_tl
182   }
183 }
184 }

```

(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.

```

185 \cs_new:Npn \__shipout_add_foreground_box:n #1
186 {
187   \box_if_vertical:NTF \l_shipout_box
188   {
189     \tl_set:Nx \l__shipout_saved_badness_tl
190     { \vfuzz=\the\vfuzz\relax
191       \vbadness=\the\vbadness\relax }
192     \vfuzz=\c_max_dim
193     \vbadness=\c_max_int
194     \vbox_set_to_ht:Nnn \l_shipout_box \l_shipout_box_ht_plus_dp_dim
195     {

```

```

196         \hbox_set:Nn \l__shipout_tmp_box
197             { \l__shipout_saved_badness_tl #1 }
198         \box_set_wd:Nn \l__shipout_tmp_box \c_zero_dim
199         \box_set_ht:Nn \l__shipout_tmp_box \c_zero_dim
200         \box_set_dp:Nn \l__shipout_tmp_box \c_zero_dim
201         \skip_zero:N \baselineskip
202         \skip_zero:N \lineskip
203         \skip_zero:N \lineskiplimit
204         \vbox_unpack:N \l__shipout_box
205         \kern -\l__shipout_box_ht_plus_dp_dim
206         \box_use:N \l__shipout_tmp_box
207         \kern \l__shipout_box_ht_plus_dp_dim
208     }
209     \l__shipout_saved_badness_tl
210     \box_set_ht:Nn \l__shipout_box \l__shipout_box_ht_dim
211     \box_set_dp:Nn \l__shipout_box \l__shipout_box_dp_dim
212 }
213 {
214     \box_if_horizontal:NT \l__shipout_box
215     {
216         \tl_set:Nx \l__shipout_saved_badness_tl
217             { \hfuzz=\the\hfuzz\relax
218               \hbadness=\the\hbadness\relax }
219         \hfuzz=\c_max_dim
220         \hbadness=\c_max_int
221         \hbox_set_to_wd:Nnn \l__shipout_box \l__shipout_box_wd_dim
222             {
223                 \hbox_unpack:N \l__shipout_box
224                 \kern -\box_wd:N \l__shipout_box
225                 \hbox_set:Nn \l__shipout_tmp_box
226                     { \l__shipout_saved_badness_tl #1 }
227                 \box_set_wd:Nn \l__shipout_tmp_box \c_zero_dim
228                 \box_set_ht:Nn \l__shipout_tmp_box \c_zero_dim
229                 \box_set_dp:Nn \l__shipout_tmp_box \c_zero_dim
230                 \box_move_up:nn { \box_ht:N \l__shipout_box }
231                     { \box_use:N \l__shipout_tmp_box }
232                 \kern \box_wd:N \l__shipout_box
233             }%
234         \l__shipout_saved_badness_tl
235     }
236 }
237 }

```

(End definition for `__shipout_add_foreground_box:n`.)

```

\__shipout_init_page_origins:
\c__shipout_horigin_tl
\c__shipout_vorigin_tl

```

Two constants holding the offset of the top-left with respect to the media box.

Setting the constants this way is courtesy of Bruno.

We delay setting the constants to the last possible place as there might be updates in the preamble or even in the `begindocument` hook that affects their setup.

```

238 \cs_new:Npn \__shipout_init_page_origins: {
239     \tl_const:Nx \c__shipout_horigin_tl
240     {
241         \cs_if_exist_use:NTF \pdfvariable { horigin }
242         { \cs_if_exist_use:NF \pdfhorigin { 1in } }

```

```

243     }
244     \tl_const:Nx \c__shipout_vorigin_tl
245     {
246         \cs_if_exist_use:NTF \pdfvariable { vorigin }
247         { \cs_if_exist_use:NF \pdfvorigin { 1in } }
248     }

```

After the constants have been set there is no need to execute this command again, in fact it would raise an error, so we redefine it to do nothing.

```

249     \cs_gset_eq:NN \__shipout_init_page_origins: \prg_do_nothing:
250 }

```

(End definition for `__shipout_init_page_origins:`, `\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 now it is not.

```

251 \cs_new:Npn \__shipout_picture_overlay:n #1 {

```

The very first time this is executed we have to initialize (and freeze) the origins.

```

252     \__shipout_init_page_origins:
253     \kern -\c__shipout_horigin_tl \scan_stop:
254     \vbox_to_zero:n {
255         \kern -\c__shipout_vorigin_tl \scan_stop:
256         \unitlength 1pt \scan_stop:

```

This mimics a simple zero-sized `picture` environment. The `\hss` is needed in case there is horizontal material (without using `\put` with a positive width).

```

257         \hbox_set_to_wd:Nnn \l__shipout_tmp_box \c_zero_dim
258         { \ignorespaces #1 \hss }
259         \box_set_ht:Nn \l__shipout_tmp_box \c_zero_dim
260         \box_set_dp:Nn \l__shipout_tmp_box \c_zero_dim
261         \box_use:N \l__shipout_tmp_box
262         \tex_vss:D
263     }
264 }

```

(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.

```

265 \cs_new:Npn \__shipout_add_background_picture:n #1 {
266     \__shipout_add_background_box:n { \__shipout_picture_overlay:n {#1} }
267 }

```

(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.

```

268 \cs_new:Npn \__shipout_add_foreground_picture:n #1 {
269     \__shipout_add_foreground_box:n { \__shipout_picture_overlay:n {#1} }
270 }

```


(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 \LaTeX looks ahead and is not using the position for on the next page).

```
271 \cs_new_protected:Npn \shipout_discard: {  
272   \bool_gset_true:N \g__shipout_discard_bool  
273 }
```

(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
`\ReadonlyShipoutCounter` order to know how many pages got produced.

```
274 \int_new:N \g_shipout_readonly_int
```

For $\LaTeX 2_\epsilon$ it is available as a command (i.e., a \TeX counter only).

```
275 \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 this counter. It is not
`\c@totalpages` used in the code but may get used in user code.

```
276 \int_new:N \g_shipout_totalpages_int
```

For $\LaTeX 2_\epsilon$ this is offered as a \LaTeX counter so can be easily typeset inside the output routine to display things like “`\thepage/\thetotalpages`”, etc.

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

```
278 \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 \TeX . 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.

```
279 \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% sure 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 chance 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 280 \g@addto@macro \@kernel@after@enddocument {
281   \int_compare:nNnT \@abspage@last = \maxdimen
282   {
We use LATEX 2ε coding as \@abspage@last is not an L3 name.
283   \xdef\@abspage@last{ \int_eval:n {\g_shipout_readonly_int + 1} }
284   }
285 }
```

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.

```
\@kernel@after@enddocument@afterlastpage 286 \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 run 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.
287   \int_compare:nNnF \g_shipout_readonly_int = 0
288   {
```

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!

```
289   \if@filesw
290     \iow_now:Nx \@auxout {
291       \gdef\string\@abspage@last {\int_use:N \g_shipout_readonly_int}}
292   \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.

```
293   \bool_if:NF \g__shipout_lastpage_handled_bool
294   {
```

However, making this extra page in case the hook is actually empty would be forcing a rerun without any reason, so we check that condition and also check if `\@kernel@after@shipout@lastpage` contains any code. If both are empty we omit the page generation.

```
295     \bool_lazy_and:nnF
296     { \hook_if_empty_p:n {shipout/lastpage} }
297     { \tl_if_empty_p:N \@kernel@after@shipout@lastpage }
```

```

298     {
299         \tex_shipout:D\vbox to\textheight
300         {
301             \hbox:n { \UseHook{shipout/lastpage}
302                 \@kernel@after@shipout@lastpage }

```

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.

```

303         \_shipout_excuse_extra_page:
304         \null
305     }

```

At this point we also signal to L^AT_EX's endgame that a rerun is necessary so that an appropriate message can be shown on the terminal. We do this by simply defining a command used as a flag and tested in `\enddocument`.

```

306         \cs_gset_eq:NN \@extra@page@added \relax
307     }
308 }
309 }
310 }

```

(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 ...

```

311 \cs_new:Npn \_shipout_excuse_extra_page: {
312     \vfil
313     \begin{center}
314         \bfseries Temporary~ page!
315     \end{center}
316     \LaTeX{}~ was~ unable~ to~ guess~ the~ total~ number~ of~ pages~
317     correctly.~ ~ As~ there~ was~ some~ unprocessed~ data~ that~
318     should~ have~ been~ added~ to~ the~ final~ page~ this~ extra~
319     page~ has~ been~ added~ to~ receive~ it.
320     \par
321     If~ you~ rerun~ the~ document~ (without~ altering~ it)~ this~
322     surplus~ page~ will~ go~ away,~ because~ \LaTeX{}~ now~ knows~
323     how~ many~ pages~ to~ expect~ for~ this~ document.
324     \vfil
325 }

```

(End definition for `_shipout_excuse_extra_page:.`)

`\PreviousTotalPages`

In the preamble before the aux file was read `\PreviousTotalPages` is always zero.

`\@kernel@before@begindocument`

```

326 \def\PreviousTotalPages{0}

```

In the aux file there should be an update for `\@abspage@last` recording the number of pages from the previous run. If not that macro holds the value of `\maxdimen`. So we test for it and update `\PreviousTotalPages` if there was a real value. This should happen just before the `begindocument` hook is executed so that the value can be used inside that hook.

```

327 \g@addto@macro\@kernel@before@begindocument
328     {\ifnum\@abspage@last<\maxdimen
329         \xdef\PreviousTotalPages{\@abspage@last}\fi}

```

(End definition for `\PreviousTotalPages` and `\@kernel@before@begindocument`. These functions are documented on page 5.)

4 Legacy L^AT_EX 2_ε interfaces

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

```
330 \cs_new_eq:NN \DiscardShipoutBox \shipout_discard:
```

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

\AtBeginDvi If we roll forward from an earlier kernel \AtBeginDvi is defined so we better not use \cs_new_protected:Npn here.

```
331 \cs_set_protected:Npn \AtBeginDvi {\_shipout_add_firstpage_material:Nn \AtBeginDvi}
```

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

\DebugShipoutsOn
\DebugShipoutsOff

```
332 \cs_new_eq:NN \DebugShipoutsOn \shipout_debug_on:
```

```
333 \cs_new_eq:NN \DebugShipoutsOff \shipout_debug_off:
```

(End definition for \DebugShipoutsOn and \DebugShipoutsOff. These functions are documented on page 5.)

5 Internal commands needed elsewhere

These internal commands use double and triple @ signs so we need to stop getting them translated to the module name.

```
334 <@@=>
```

Some internals needed elsewhere.

```
335 \cs_set_eq:NN \@expl@@@shipout@add@firstpage@material@@Nn
```

```
336 \_shipout_add_firstpage_material:Nn
```

```
337 \cs_set_eq:NN \@expl@@@shipout@add@background@box@@n
```

```
338 \_shipout_add_background_box:n
```

```
339 \cs_set_eq:NN \@expl@@@shipout@add@foreground@box@@n
```

```
340 \_shipout_add_foreground_box:n
```

```
341 \cs_set_eq:NN \@expl@@@shipout@add@background@picture@@n
```

```
342 \_shipout_add_background_picture:n
```

```
343 \cs_set_eq:NN \@expl@@@shipout@add@foreground@picture@@n
```

```
344 \_shipout_add_foreground_picture:n
```

(End definition for \@expl@@@shipout@add@firstpage@material@@Nn and others. These functions are documented on page ??.)

```
345 \ExplSyntaxOff
```

```
346 </2kernel | latexrelease>
```

```
347 <latexrelease>\EndIncludeInRelease
```

Rolling back here doesn't undefine the interface commands as they may be used in packages without rollback functionality. So we just make them do nothing which may or may not work depending on the code usage.

```
348 <latexrelease>\IncludeInRelease{0000/00/00}%
```

```
349 <latexrelease> \{shipout\}{Hook management (shipout)}%
```

```
350 <latexrelease>
```

If we roll forward then `\tex_shipout:D` may not be defined in which case `\shipout` does have its original definition and so we must not `\let` it to something else which is `\relax!`

```

351 <latexrelease>\ifcsname tex_shipout:D\endcsname
352 <latexrelease>\expandafter\let\expandafter\shipout
353 <latexrelease>                \csname tex_shipout:D\endcsname
354 <latexrelease>\fi
355 <latexrelease>
356 <latexrelease>\let \ShipoutBox\@undefined
357 <latexrelease>\let \ReadOnlyShipoutCounter \@undefined
358 <latexrelease>\let \c@totalpages \@undefined
359 <latexrelease>\let \thetotalpages \@undefined
360 <latexrelease>
361 <latexrelease>\let \DiscardShipoutBox \@undefined
362 <latexrelease>\let \DebugShipoutsOn \@undefined
363 <latexrelease>\let \DebugShipoutsOff \@undefined
364 <latexrelease>
365 <latexrelease>\DeclareRobustCommand \AtBeginDvi [1]{%
366 <latexrelease> \global \setbox \@begindvibox
367 <latexrelease> \vbox{\unvbox \@begindvibox #1}%
368 <latexrelease>}
369 <latexrelease>
370 <latexrelease>\let \AtBeginShipout \@undefined
371 <latexrelease>\let \AtBeginShipoutNext \@undefined
372 <latexrelease>
373 <latexrelease>\let \AtBeginShipoutFirst \@undefined
374 <latexrelease>
375 <latexrelease>\let \ShipoutBoxHeight \@undefined
376 <latexrelease>\let \ShipoutBoxDepth \@undefined
377 <latexrelease>\let \ShipoutBoxWidth \@undefined
378 <latexrelease>
379 <latexrelease>\let \AtBeginShipoutDiscard \@undefined
380 <latexrelease>
381 <latexrelease>\let \AtBeginShipoutAddToBox \@undefined
382 <latexrelease>\let \AtBeginShipoutAddToBoxForeground \@undefined
383 <latexrelease>\let \AtBeginShipoutUpperLeft \@undefined
384 <latexrelease>\let \AtBeginShipoutUpperLeftForeground \@undefined
385 <latexrelease>

```

We do not undo a substitution when rolling back. As the file support gets undone the underlying data is no longer used (and sufficiently obscure that it should not interfere with existing commands) and properly removing it would mean we need to make the `\undeclare@...` and its support macros available in all earlier kernel releases which is pointless (and actually worse).

```

386 %\undeclare@file@substitution{everyshi.sty}
387 <latexrelease>
388 <latexrelease>\let \AtEndDvi \@undefined

```

We do not reenabale a disabled package load when rolling back. As the file support gets undone the underlying data is no longer checked (and sufficiently obscure that it should not interfere with existing commands) and properly removing it would mean we need to make the `\reenable@package@load` command available in all earlier kernel releases which is pointless (and actually worse).

```

389 %\reenable@package@load{atenddvi}
390 <latexrelease>
391 <latexrelease>\EndIncludeInRelease
392 <*2ekernel>

```

6 Package emulation for compatibility

6.1 Package `atenddvi` emulation

`\AtEndDvi` This package has only one public command to simulating it is easy and actually sensible to provide as part of the kernel.

```

393 </2ekernel>
394 <*2ekernel | latexrelease>
395 <latexrelease>\IncludeInRelease{2020/10/01}%
396 <latexrelease>          {\AtEndDvi}{atenddvi emulation}%
397 \ExplSyntaxOn
398 \cs_new_protected:Npn \AtEndDvi {\AddToHook{shipout/lastpage}}
399 \ExplSyntaxOff

```

As the package is integrate we prevent loading (no need to roll that back):

```

400 \disable@package@load{atenddvi}
401   {\PackageWarning{atenddvi}
402     {Functionality of this package is already\MessageBreak
403       provided by LaTeX.\MessageBreak\MessageBreak
404       It is there no longer necessary to load it\MessageBreak
405       and you can safely remove it.\MessageBreak
406       Found on}}
407 </2ekernel | latexrelease>
408 <latexrelease>\EndIncludeInRelease
409 <latexrelease>\IncludeInRelease{0000/00/00}%
410 <latexrelease>          {\AtEndDvi}{atenddvi emulation}%
411 <latexrelease>\let \AtEndDvi \@undefined
412 <latexrelease>\EndIncludeInRelease
413 <*2ekernel>

```

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

```

414 </2ekernel>

```

6.2 Package `atbegshi` emulation

```

415 <*atbegshi-ltx>
416 \ProvidesPackage{atbegshi-ltx}
417   [2020/10/15 v1.0b
418     Emulation of the original atbegshi^^Jpackage with kernel methods]

```

`\AtBeginShipoutBox`

```

419 \let \AtBeginShipoutBox \ShipoutBox

```

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

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

```

420 \let \AtBeginShipoutInit \@empty

```

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

`\AtBeginShipout` Filling hooks
`\AtBeginShipoutNext` 421 `\protected \def \AtBeginShipout {\AddToHook{shipout/before}}`
422 `\protected \def \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.

423 `\protected \def \AtBeginShipoutFirst`
424 `{\@expl@@@shipout@add@firstpage@material@@Nn \AtBeginShipoutFirst}`

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

`\AtBeginShipoutDiscard` Just a different name.

425 `\let \AtBeginShipoutDiscard \DiscardShipoutBox`

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

`\AtBeginShipoutAddToBox` We don't expose them.

`\AtBeginShipoutAddToBoxForeground` 426 `\let \AtBeginShipoutAddToBox`
`\AtBeginShipoutUpperLeft` 427 `\@expl@@@shipout@add@background@box@@n`
`\AtBeginShipoutUpperLeftForeground` 428 `\let \AtBeginShipoutAddToBoxForeground`
429 `\@expl@@@shipout@add@foreground@box@@n`
430 `\let \AtBeginShipoutUpperLeft`
431 `\@expl@@@shipout@add@background@picture@@n`
432 `\let \AtBeginShipoutUpperLeftForeground`
433 `\@expl@@@shipout@add@foreground@picture@@n`

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

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

434 `\ExplSyntaxOn`
435 `\cs_new:Npn \ShipoutBoxHeight { \dim_use:N \l_shipout_box_ht_dim }`
436 `\cs_new:Npn \ShipoutBoxDepth { \dim_use:N \l_shipout_box_dp_dim }`
437 `\cs_new:Npn \ShipoutBoxWidth { \dim_use:N \l_shipout_box_wd_dim }`
438 `\ExplSyntaxOff`

(End definition for `\ShipoutBoxHeight`, `\ShipoutBoxWidth`, and `\ShipoutBoxDepth`. These functions are documented on page ??.)

439 `</atbegshi-ltx>`

If the package is requested we substitute the one above:

440 `(*2kernel)`
441 `\declare@file@substitution{atbegshi.sty}{atbegshi-ltx.sty}`
442 `</2kernel>`

6.3 Package everyshi emulation

```
443 <*everyshi-ltx>
444 \ProvidesPackage{everyshi-ltx}
445 [2020/10/15 v1.0b
446 Emulation of the original everyshi^^Jpackage with kernel methods]
\EveryShipout This package has only two public commands so simulating it is easy:
\AtNextShipout
447 \protected \def \EveryShipout {\AddToHook{shipout/before}}
448 \protected \def \AtNextShipout {\AddToHookNext{shipout/before}}
(End definition for \EveryShipout and \AtNextShipout. These functions are documented on page 7.)
449 % This is one difference between \pkg{everyshi} and the kernel
450 % implementation, the latter does not directly use box 255.
451 %
452 % For usage by ordinary users this makes no difference but of a
453 % package use complicated code together with \pkg{everyshi} and
454 % directly manipulates box 255 then this package needs updating.
455 % In most cases the updates are simple because the kernel offers
456 % hooks that makes such complicated code unnecessary.
457 %
458 % We therefore add a little file into the adjusted package
459 % \begin{macrocode}
460 %%
461 %% In normal circumstances the above emulation is sufficient and in
462 %% all known packages (we know of) that use everyshi it either works or
463 %% the packages have been adjusted.
464 %%
465 %% Code that directly manipulates box 255, however, might fail.
466 %% If that is the case look at the shipout hooks offered now as
467 %% they are normally sufficient to avoid such manipulations (or
468 %% replace box 255 with \ShipoutBox in the code.
469 %%
470 </everyshi-ltx>
If the package is requested we substitute the one above:
471 <*2kernel>
472 \declare@file@substitution{everyshi.sty}{everyshi-ltx.sty}
473 </2kernel>
Rather important :-)
474 <@@=>
```


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**
- `\AddEverypageHook` 6
 - `\AddThispageHook` 6
 - `\AddToHook` 5, 5, 5, 6, 6, 118, 398, 421, 447
 - `\AddToHookNext` 3, 5, 6, 6, 422, 448
 - `\Alph` 4
 - `\arabic` 4, 278
 - `\AtBeginDvi` 3, 5, 11, 11, 19, 331, 365
 - `\AtBeginShipout` 5, 370, 421
 - `\AtBeginShipoutAddToBox` 5, 381, 426
 - `\AtBeginShipoutAddToBoxForeground` ..
..... 5, 382, 426
 - `\AtBeginShipoutBox` 5, 419
 - `\AtBeginShipoutDiscard` 5, 379, 425
 - `\AtBeginShipoutFirst` 5, 373, 423
 - `\AtBeginShipoutInit` 5, 420
 - `\AtBeginShipoutNext` 5, 371, 421
 - `\AtBeginShipoutUpperLeft` .. 5, 5, 383, 426
 - `\AtBeginShipoutUpperLeftForeground` .
..... 5, 5, 384, 426
 - `\AtEndDvi` 3, 6, 388, 393
 - `\AtNextShipout` 6, 447
- B**
- `\baselineskip` 150, 201
 - `\begin` 313, 459
 - `\bfseries` 314
 - bool commands:
 - `\bool_gset_false:N` 15, 43, 51
 - `\bool_gset_true:N` 10, 89, 272
 - `\bool_if:NTF` 21, 49, 293
 - `\bool_lazy_and:nnTF` 73, 81, 295
 - `\bool_new:N` 6, 116, 131
 - box commands:
 - `\box_dp:N` 122
 - `\box_ht:N` 8, 121, 230
 - `\box_if_empty:NTF` 40, 58
 - `\box_if_horizontal:NTF` 162, 214
 - `\box_if_vertical:NTF` 136, 187
 - `\box_move_up:nn` 176, 230
 - `\box_new:N` 23, 132
 - `\box_set_dp:Nn`
..... 149, 158, 175, 200, 211, 229, 260
 - `\box_set_eq_drop:NN` 54
 - `\box_set_ht:Nn`
..... 148, 157, 174, 199, 210, 228, 259
 - `\box_set_wd:Nn` 147, 173, 198, 227
 - `\box_use:N` . 92, 153, 178, 206, 231, 261
 - `\box_wd:N` 123, 224, 232
 - `\l_shipout_box` 1,
2, 8, 8, 12, 12, 23, 29, 40, 46, 54,
58, 69, 92, 135, 136, 143, 154, 157,
158, 162, 169, 179, 187, 194, 204,
210, 211, 214, 221, 223, 224, 230, 232
- C**
- `\clearpage` 17, 17
 - cs commands:
 - `\cs_gset:Npn` 110
 - `\cs_gset_eq:NN` 31, 109, 249, 306
 - `\cs_gset_protected:Npx` 20
 - `\cs_if_exist_use:NTF` 241, 242, 246, 247
 - `\cs_new:Npn` 33,
39, 106, 117, 120, 134, 185, 238,
251, 265, 268, 278, 311, 435, 436, 437
 - `\cs_new_eq:NN`
..... 7, 98, 275, 277, 330, 332, 333
 - `\cs_new_protected:Npn`
..... 8, 13, 18, 19, 271, 398
 - `\cs_set_eq:NN` 24,
44, 56, 91, 95, 335, 337, 339, 341, 343
 - `\cs_set_protected:Npn` 25, 331
 - `\csname` 353
- D**
- `\DebugShipoutsOff` 4, 332, 363
 - `\DebugShipoutsOn` 4, 332, 362
 - `\DeclareRobustCommand` 365
 - `\def` 326, 421, 422, 423, 447, 448
 - dim commands:
 - `\dim_new:N` 127, 128, 129, 130
 - `\dim_set:Nn` 121, 122, 123, 124
 - `\dim_use:N` 435, 436, 437
 - `\c_max_dim` 141, 167, 192, 219
 - `\c_zero_dim` 147,
148, 149, 155, 173, 174, 175, 198,
199, 200, 227, 228, 229, 257, 259, 260
 - `\DiscardShipoutBox` .. 2, 3, 5, 330, 361, 425
- E**
- `\end` 315
 - `\endcsname` 351, 353
 - `\enddocument` 17, 17, 18, 280
 - `\EndIncludeInRelease` .. 347, 391, 408, 412
 - `\EveryShipout` 6, 447
 - exp commands:
 - `\exp_not:N` 56, 91

<code>\expandafter</code>	352	377, 379, 381, 382, 383, 384, 388,
<code>\ExplSyntaxOff</code>	345, 399, 438	411, 419, 420, 425, 426, 428, 430, 432
<code>\ExplSyntaxOn</code>	5, 397, 434	
F		
<code>\fi</code>	292, 329, 354	
G		
<code>\gdef</code>	291	
<code>\global</code>	366	
group commands:		
<code>\group_begin:</code>	53	
<code>\group_end:</code>	55	
H		
<code>\hbadness</code>	166, 168, 218, 220	
<code>\hbox</code>	1, 2, 5	
hbox commands:		
<code>\hbox:n</code>	301	
<code>\hbox_set:Nn</code>	145, 171, 196, 225	
<code>\hbox_set_to_wd:Nnn</code>	169, 221, 257	
<code>\hbox_unpack:N</code>	179, 223	
<code>\hfuzz</code>	165, 167, 217, 219	
hook commands:		
<code>\hook_if_empty:nTF</code>	70, 107	
<code>\hook_if_empty_p:n</code>	74, 82, 296	
<code>\hook_new:n</code>	99, 100, 101, 102, 103	
<code>\hook_use:n</code>	47, 72, 78	
<code>\hss</code>	15, 258	
I		
<code>\ifcsname</code>	351	
<code>\ifnum</code>	4, 328	
<code>\ignorespaces</code>	258	
<code>\IncludeInRelease</code>	3, 348, 395, 409	
int commands:		
<code>\int_compare:nNnTF</code> ..	34, 80, 281, 287	
<code>\int_eval:n</code>	283	
<code>\int_gincr:N</code>	48, 64	
<code>\int_new:N</code>	274, 276	
<code>\int_use:N</code>	4, 4, 66, 85, 291	
<code>\int_value:w</code>	27	
<code>\c_max_int</code>	142, 168, 193, 220	
<code>\c_zero_int</code>	52	
iow commands:		
<code>\iow_now:Nn</code>	290	
K		
<code>\kern</code> ..	13, 155, 205, 207, 224, 232, 253, 255	
L		
<code>\LaTeX</code>	316, 322	
<code>\let</code>	20, 104,	
	105, 352, 356, 357, 358, 359, 361,	
	362, 363, 370, 371, 373, 375, 376,	
<code>\lineskip</code>	151, 202	
<code>\lineskiplimit</code>	152, 203	
M		
<code>\maxdimen</code>	18, 279, 281, 328	
<code>\MessageBreak</code> ..	60, 112, 402, 403, 404, 405	
N		
<code>\nofiles</code>	17	
<code>\null</code>	304	
<code>\number</code>	279	
P		
<code>\PackageInfo</code>	50	
<code>\PackageWarning</code>	41, 59, 401	
<code>\par</code>	320	
<code>\pdfhorigin</code>	242	
<code>\pdfvariable</code>	241, 246	
<code>\pdfvorigin</code>	247	
<code>\pkg</code>	449, 453	
<code>\PreviousTotalPages</code>	4, 18, 18, 326	
prg commands:		
<code>\prg_do_nothing:</code>	109, 249	
<code>\protect</code> ..	8, 9, 10, 10, 10, 44, 56, 91, 95, 98	
<code>\protected</code>	421, 422, 423, 447, 448	
<code>\ProvidesPackage</code>	416, 444	
<code>\put</code>	2, 5, 6, 6, 15	
R		
<code>\ReadOnlyShipoutCounter</code>	4	
<code>\ReadOnlyShipoutCounter</code> ..	4, 4, 274, 357	
<code>\relax</code>	20, 139,	
	140, 165, 166, 190, 191, 217, 218, 306	
<code>\RequirePackage</code>	4	
<code>\Roman</code>	4	
S		
scan commands:		
<code>\scan_stop:</code>	253, 255, 256	
<code>\setbox</code>	366	
<code>\shipout</code>	1,	
	1, 3, 4, 7, 7, 8, 8, 10, 20, 31, 349, 352	
shipout commands:		
<code>\l_shipout_box_dp_dim</code>		
	1, 122, 125, 127, 158, 211, 436	
<code>\l_shipout_box_ht_dim</code>		
	1, 121, 124, 127, 157, 177, 210, 435	
<code>\l_shipout_box_ht_plus_dp_dim</code> ..		
	1, 124, 127, 143, 194, 205, 207	
<code>\l_shipout_box_wd_dim</code>		
	1, 123, 127, 169, 221, 437	
<code>\shipout_debug_off:</code>	4, 7, 333	
<code>\shipout_debug_on:</code>	4, 7, 332	

`\shipout_discard`: [271](#), [330](#)
`\shipout_discard_box`: [3](#)
`\g_shipout_readonly_int` [4](#),
[8](#), [64](#), [66](#), [80](#), [85](#), [274](#), [283](#), [287](#), [291](#)
`\g_shipout_totalpage_int` [4](#)
`\g_shipout_totalpages_int` [4](#), [8](#), [48](#), [276](#)
shipout internal commands:
`__shipout_add_background_box:n` .
. [108](#), [134](#), [266](#), [338](#)
`__shipout_add_background_`
`picture:n` [76](#), [265](#), [342](#)
`__shipout_add_firstpage_`
`material:Nn` [110](#), [117](#), [331](#), [336](#)
`__shipout_add_foreground_box:n` .
. [86](#), [185](#), [269](#), [340](#)
`__shipout_add_foreground_`
`picture:n` [71](#), [268](#), [344](#)
`__shipout_debug:n` [7](#), [7](#), [65](#), [84](#)
`\g__shipout_debug_bool` . . . [6](#), [10](#), [15](#), [21](#)
`__shipout_debug_gset`: [7](#)
`\g__shipout_discard_bool`
. [43](#), [49](#), [51](#), [131](#), [272](#)
`__shipout_excuse_extra_page`:
. [303](#), [311](#)
`__shipout_execute`: [7](#), [25](#), [31](#)
`__shipout_execute_cont`: [37](#), [39](#)
`__shipout_execute_firstpage_`
`hook`: [10](#), [79](#), [106](#)
`__shipout_execute_test_level`:
. [28](#), [33](#)
`__shipout_get_box_size:N`
. [12](#), [46](#), [69](#), [120](#), [135](#)
`\l__shipout_group_level_tl` [26](#), [32](#), [35](#)
`\c__shipout_horigin_tl` [238](#), [253](#)
`__shipout_init_page_origins`:
. [238](#), [252](#)
`\g__shipout_lastpage_handled_`
`bool` [89](#), [116](#), [293](#)
`__shipout_picture_overlay:n`
. [251](#), [266](#), [269](#)
`\l__shipout_saved_badness_tl`
. [132](#), [138](#), [146](#), [159](#), [164](#),
[172](#), [181](#), [189](#), [197](#), [209](#), [216](#), [226](#), [234](#)
`__shipout_saved_protect`: . . . [44](#), [95](#), [98](#)
`\l__shipout_tmp_box` [132](#), [145](#), [147](#),
[148](#), [149](#), [153](#), [171](#), [173](#), [174](#), [175](#),
[178](#), [196](#), [198](#), [199](#), [200](#), [206](#), [225](#),
[227](#), [228](#), [229](#), [231](#), [257](#), [259](#), [260](#), [261](#)
`\c__shipout_vorigin_tl` [238](#), [255](#)
shipout/background [2](#), [99](#)
shipout/before [2](#), [99](#)
shipout/firstpage [2](#), [99](#)
shipout/foreground [2](#), [99](#)
shipout/lastpage [2](#), [99](#)

`\ShipoutBox` [1](#), [2](#), [3](#), [6](#), [23](#), [356](#), [419](#), [468](#)
`\ShipoutBoxDepth` [376](#), [434](#)
`\ShipoutBoxHeight` [22](#), [375](#), [434](#)
`\ShipoutBoxWidth` [377](#), [434](#)
skip commands:
`\skip_zero:N` [150](#), [151](#), [152](#), [201](#), [202](#), [203](#)
`\space` [67](#)
`\special` [2](#)
`\string` [113](#), [291](#)

T

TeX and L^AT_EX 2_ε commands:
`\@abspage@last` [16](#), [17](#), [17](#),
[18](#), [67](#), [80](#), [279](#), [281](#), [283](#), [291](#), [328](#), [329](#)
`\@auxout` [290](#)
`\@begindvi` [6](#)
`\@begindvibox` [2](#), [366](#), [367](#)
`\@cclv` [6](#)
`\@empty` [104](#), [105](#), [420](#)
`\@expl@@@shipout@add@background@box@Cn`
. [335](#), [427](#)
`\@expl@@@shipout@add@background@picture@Cn`
. [335](#), [431](#)
`\@expl@@@shipout@add@firstpage@material@Cn`
. [335](#), [424](#)
`\@expl@@@shipout@add@foreground@box@Cn`
. [335](#), [429](#)
`\@expl@@@shipout@add@foreground@picture@Cn`
. [335](#), [433](#)
`\@extra@page@added` [306](#)
`\@kernel@after@enddocument` [280](#)
`\@kernel@after@enddocument@afterlastpage`
. [286](#)
`\@kernel@after@shipout@lastpage` .
. [17](#), [83](#), [87](#), [104](#), [297](#), [302](#)
`\@kernel@before@begindocument` . . . [326](#)
`\@kernel@before@shipout@background`
. [75](#), [77](#), [104](#)
`\@latex@warning` [111](#)
`\@undefined` [356](#), [357](#), [358](#), [359](#), [361](#),
[362](#), [363](#), [370](#), [371](#), [373](#), [375](#), [376](#),
[377](#), [379](#), [381](#), [382](#), [383](#), [384](#), [388](#), [411](#)
`\c@totalpages` [276](#), [358](#)
`\declare@file@substitution` [441](#), [472](#)
`\disable@package@load` [400](#)
`\g@addto@macro` [280](#), [286](#), [327](#)
`\if@files` [289](#)
`\reenable@package@load` [20](#), [389](#)
`\set@typeset@protect` [45](#)
`\unclare@...` [20](#)
`\undeclare@file@substitution` [386](#)

tex commands:
`\tex_afterassignment:D` [28](#)
`\tex_aftergroup:D` [8](#), [36](#)

<code>\tex_currentgrouplevel:D</code>	27, 35	use commands:	
<code>\tex_deadcycles:D</code>	52	<code>\use_none:n</code>	7
<code>\tex_setbox:D</code>	29	<code>\UseHook</code>	86, 108, 301
<code>\tex_shipout:D</code>	92, 299	<code>\usepackage</code>	4
<code>\tex_vss:D</code>	262		
<code>\textheight</code>	299		
<code>\the</code>	139, 140, 165, 166, 190, 191, 217, 218	V	
<code>\thepage</code>	16	<code>\vbadness</code>	12, 140, 142, 191, 193
<code>\thetotalpages</code>	4, 16, 278, 359	<code>\vbox</code>	1, 12, 299, 367
tl commands:		vbox commands:	
<code>\tl_const:Nn</code>	239, 244	<code>\vbox_set_to_ht:Nnn</code>	143, 194
<code>\tl_if_empty_p:N</code>	75, 83, 297	<code>\vbox_to_zero:n</code>	254
<code>\tl_new:N</code>	32, 133	<code>\vbox_unpack:N</code>	154, 204
<code>\tl_set:Nn</code>	26, 138, 164, 189, 216	<code>\vfil</code>	312, 324
totalpages	4	<code>\vfuzz</code>	12, 139, 141, 190, 192
<code>\typeout</code>	3, 66, 84		
		W	
U		<code>\write</code>	2, 10
<code>\unitlength</code>	2, 15, 256		
<code>\unvbox</code>	367	X	
		<code>\xdef</code>	279, 283, 329