The **abspos** package
# Absolute placement with coffins

Magnus Lie Hetland

September 22, 2022

The abspos package lets you place contents at an absolute position, anchored at some specified part of the content, similar to how tikz nodes work, though without using the two-pass strategy of tikz. It also avoids messing with the order of beamer overlays, which is what happens when one uses the textpos package with the overlay option.[1] The solution used is quite straightforward, combining *coffins* (using l3coffins) with the placement mechanisms of atbegshi.

The main command of the package is \absput, which is somewhat similar to the \put command of the LaTeX picture environment. By default, it places its contents in the middle of the page, so \absput{Hello} will put the word "Hello" front and center. Variations are then possible using an optional argument consisting of keys and key–value pairs, as shown in the following.
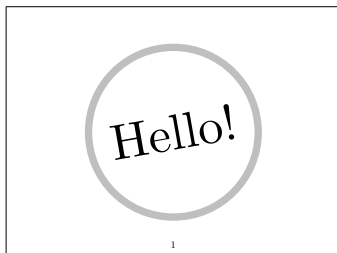

```
% In preamble:
% \usepackage{tikz}
\absput{\tikz{
    \draw[line width=8pt, lightgray] circle[radius=3.25];
}}
\absput[angle=10, scale=5]{Hello!}
\absput[pg-b, y=.5cm]{\thepage}
```

Supplying a width creates a vertical box. If the surrounding text is set, for example, using \raggedright (as is common in beamer), the contents of the \absput command will be, too. So if you want your text justified, you can use something like the \justifying command from ragged2e.
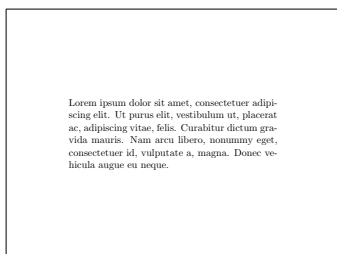

```
% In preamble:
% \usepackage{ragged2e}
% \usepackage{lipsum}
\absput[width=8cm]{
    \justifying
    \lipsum[1][1-5] % Example text
}
```

---

[1]Another possible solution is to use the pst-abspos package, which relies on pstricks.

The options can also be set globally (or within a group) using `\absposset`. For example, if you with to scale everything you place by a certain amount, you could do that as in the following.

```
\absposset{scale=5}
\absput{Scaled!}
```

```
Scaled!
```

By default, the contents are placed *in front* of the current page, but this can be modified by supplying your own "page coffin" (an l3coffins or xcoffins coffin) and typesetting that with, e.g., atbegshi. If, for example, you wish to place something in the background of every slide of a beamer presentation, you could do that by adding the following to the preamble.[2]

```
\setbeamercolor{background canvas}{bg=} % So the coffin isn't obscured

\NewCoffin \MyPageCoffin
\AtBeginShipout{
    \AtBeginShipoutUpperLeft{
        \TypesetCoffin \MyPageCoffin [t,l] (0pt,0pt)
    }
}
```

Now simply add a `\absput` command at the beginning of the document, before your first `frame` (after `\begin{document}`), using `pg=\MyPageCoffin` to specify which coffin to use for rendering and positioning contents. This gives you a coordinate system starting at the top left corner of the page. If you wish to anchor your positioned elements elsewhere, you'll need to resize your page coffin (see p.).

Reference

`\absposset` `\absposset {⟨options⟩}`

Sets the options, as described under `\absput`, to the value they should have within the current group (or globally, if not inside a group).

`\absput` `\absput [⟨options⟩] {⟨contents⟩}`

Places ⟨*contents*⟩ at some position (possibly scaled or rotated) as dictated by the optional ⟨*options*⟩, supplied as keys and key–value pairs. The *defaults* are used as values if only the keys are given, and the *initial values* are used if the options are not specified. The available options are as follows.

- `angle=`⟨*angle*⟩: The angle to which the coffin is rotated before being positioned.

  No default. Initial value `0`.

---

[2]Note that because it will end up behind even the slide itself, you need to remove the background color, by setting `bg` to nothing, as in the example code. If you want it behind the content, but still have a background color, you would need to add that yourself, with a colored box or filled tikz rectangle, possibly also using the `\absput` command.

- h=⟨*h-anchor*⟩: The horizontal anchor to use for positioning. This corresponds to a l3coffins (or xcoffins) *pole*. The available anchors are l for left, hc for center and r for right. These are also available as separate valueless keys, so if you supply \absput with the key l, that is equivalent to using h=l, etc.

  There are also two-letter shortcuts defined for setting the horizontal and vertical anchors simultaneously, with the vertical one first. For example, tl is equivalent to v=t,h=l. In these double shortcuts, the prefixes h and v have been stripped from the center anchors, so that bottom center is bc, for example, and not bhc.

  No default. Initial value hc, which centers the coffin horizontally at ⟨*x-coord*⟩.

- pg=⟨*page-coffin*⟩: Supplies the coffin to use as the page, or "canvas," for placing the contents.

  No default. The initial value is an internal canvas that is rendered in front of the current page (on shipout), and then cleared.

- pg-h=⟨*page-h-anchor*⟩: Similar to h, except that it applies to the canvas (set by pg). Shortcuts (like pg-l for pg-h=l) are defined for the default anchors (l, hc, r).

  Double shortcuts are defined for pg-h and pg-v in a similar manner to those defined for h and v, so that pg-tl is equivalent to pg-v=t,pg-h=l.

  No default. Initial value hc.

- pg-v=⟨*page-v-anchor*⟩: Similar to v, except that it applies to the canvas (set by pg). Shortcuts (like pg-t for pg-v=t) are defined for the default anchors (t, vc, b, T, B, H).

  Combined shortcuts such as pg-tl and pg-br, etc., are also defined. See the descriptions of h and pg-h for more.

  No default. Initial value vc.

- scale=⟨*scale*⟩: The factor by which the coffin is scaled.

  No default. If scale is not used, no scaling is performed.

- v=⟨*v-anchor*⟩: The vertical anchor to use for positioning (see h). The available anchors are t for top, vc for center, b for bottom and H for baseline. These are also available as separate valueless keys, so if you supply \absput with the key t, that is equivalent to using v=t, etc.

  Note that if the width key is set, a vertial coffin is used, which makes the additional anchors (or poles) T (baseline of the first line, or other material at the top) and B (baseline for the last line, or other material at the bottom).

  Combined shortcuts such as tl and br, etc., are also defined. See the description of h for more.

  No default. Initial value vc, which centers the coffin vertically at ⟨*y-coord*⟩.

- width=⟨*width*⟩: Set the width constraining the contents, before scaling. If this is set, the contents are set in vertical mode; otherwise, they are set in horizontal mode.

  No default and no initial value.

- x=⟨*x-coord*⟩: The x coordinate, measured from the current horizontal page anchor, set by pg-h.

  No default. Initial value 0pt.

- y=⟨*y-coord*⟩: The y coordinate, measured from the current vertical page anchor, set by pg-v.

  No default. Initial value 0pt.

\absputcoffin   \absputcoffin [⟨*options*⟩] ⟨*coffin*⟩

Places ⟨*coffin*⟩ at some position (possibly scaled or rotated), just like \absput. The difference is that a coffin is provided directly, rather than constructed by typesetting some supplied contents. This means that the width key is not used, though it is permitted for consistency (and ease of reusing option lists).

Implementation

```
1 ⟨*package⟩
2 ⟨@@=apos⟩
```

First, we need to import the necessary packages. We're using expl3 to set up the package, so we get l3coffins for free (and cannot import it directly, anyway). If abspos is used with beamer, then atbegshi is already included. Elsewhere, though, it will not be, so we require it here.

```
3 \RequirePackage{atbegshi}
```

We make sure we have the variants we need of some existing commands:

```
4 \cs_generate_variant:Nn \tl_if_novalue:nTF       { VTF }
5 \cs_generate_variant:Nn \tl_if_novalue:nF        { VF }
6 \cs_generate_variant:Nn \coffin_gattach:NnnNnnnn { NooNoonn }
```

\g_apos_shipout_coffin   This variable is normally used to store the contents to be placed on the current page. It accumulates the contents from every call to \absput until the page is about to be typeset (shipped out), at which point \g_apos_shipout_coffin is typeset in front.

```
7 \coffin_new:N \g_apos_shipout_coffin
```

\apos_reset_shipout_coffin: Used to initialize, and later reinitialize, the shipout coffin, to simply be an empty coffin with the width and height of the current page.

```
8 \cs_new:Npn \apos_reset_shipout_coffin: {
9     \hcoffin_gset:Nn \g_apos_shipout_coffin {
10        \phantom{\rule{\paperwidth}{\paperheight}}}
11    }
12 }
```

The following hook ensures that that \g_apos_shipout_coffin is typeset in front of the page as it is being shipped out (anchored top left), and then cleared, ready to accumulate contents for the next page. The atbegshi command used wraps its contents in a picture environment, which we don't really need, but it seems to be more trouble than it's worth to get rid of it.

```
13 \apos_reset_shipout_coffin:
14 \AtBeginShipout{
```

```
15      \AtBeginShipoutUpperLeftForeground{
16          \coffin_typeset:Nnnnn \g_apos_shipout_coffin {t} {l} {0pt} {0pt}
17          \apos_reset_shipout_coffin:
18      }
19  }
```

\g_apos_canvas_coffin   This variable is used as part of the `pg` key, and is initially set equal to `\g_apos_shipout_-`
`coffin`, so that whatever is rendered to the canvas (i.e., `\g_apos_canvas_coffin`) is
placed on top of the page contents on shipout, and then cleared. If a different behavior
is desired, another coffin can can be supplied instead. Rather than constructing a coffin,
we just set up a token list, which initially will be aliased to the actual coffin `\g_apos_-`
`shipout_coffin` as part of the key handling.

```
20  \tl_new:N \g_apos_canvas_coffin
```

\apos_reset_overlay: This is a utility to reset `beamer` overlay specifications at the end of our coffin contents.
The idea is that if we use, say, `\pause` inside `\absput`, then everything after the `\pause`
command will be paused, including material *outside* `\absput`. To restrict such commands
to the contents provided to `\absput`, we use `\onslide<1->` after typesetting the contents
in our coffin. This is only useful (or possible) when using `beamer`, so otherwise, we just
alias `\ap_reset_overlay:` to `\relax`, i.e., don't do anything.

```
21  \tl_new:N \apos_reset_overlay:
22  \@ifclassloaded{beamer}
23      { \tl_gset:Nn    \apos_reset_overlay: { \onslide<1-> } }
24      { \tl_gset_eq:NN \apos_reset_overlay:   \relax          }
```

We now define the various keys used to configure `\absput`, as described in the reference.

```
25  \keys_define:nn { abspos } {
26
27      angle .tl_set:N          = \l_apos_angle_tl,
28      angle .initial:V         = \c_novalue_tl,
29      angle .value_required:n  = true,
30
31      h     .tl_set:N          = \l_apos_hanchor_tl,
32      h     .initial:n         = hc,
33      h     .value_required:n  = true,
34
35      pg    .code:n            = { \tl_set_eq:NN \g_apos_canvas_coffin #1 },
36      pg    .initial:n         = \g_apos_shipout_coffin,
37      pg    .value_required:n  = true,
38
39      pg-h  .tl_set:N          = \l_apos_on_hanchor_tl,
40      pg-h  .initial:n         = hc,
41      pg-h  .value_required:n  = true,
42
43      pg-v  .tl_set:N          = \l_apos_on_vanchor_tl,
44      pg-v  .initial:n         = vc,
45      pg-v  .value_required:n  = true,
46
47      scale .tl_set:N          = \l_apos_scale_tl,
48      scale .initial:V         = \c_novalue_tl,
49      scale .value_required:n  = true,
50
51      v     .tl_set:N          = \l_apos_vanchor_tl,
```

```
52      v      .initial:n         = vc,
53      v      .value_required:n  = true,
54
55      width .tl_set:N           = \l_apos_width_tl,
56      width .initial:V          = \c_novalue_tl,
57      width .value_required:n   = true,
58
59      x      .tl_set:N          = \l_apos_x_tl,
60      x      .initial:n         = { 0pt },
61      x      .value_required:n  = true,
62
63      y      .tl_set:N          = \l_apos_y_tl,
64      y      .initial:n         = { 0pt },
65      y      .value_required:n  = true,
66
67  }
68
69  \clist_const:Nn \__apos_hanchors_clist {l,hc,r}
70  \clist_const:Nn \__apos_vanchors_clist {t,vc,b,H,T,B}
71
72  \cs_new:Npn \__apos_define_shortcuts:nnn #1 #2 #3 {
73      \keys_define:nn { abspos } {
74
75      #1    .meta:n            = { #2 },
76      #1    .value_forbidden:n = true,
77
78      pg-#1 .meta:n            = { #3 },
79      pg-#1 .value_forbidden:n = true,
80
81      }
82  }
83
84  \cs_generate_variant:Nn \__apos_define_shortcuts:nnn { Vnn }
85
86  \clist_map_inline:Nn \__apos_hanchors_clist {
87
88      \__apos_define_shortcuts:nnn { #1 } { h = #1 } { pg-h = #1 }
89
90  }
91
92  \clist_map_inline:Nn \__apos_vanchors_clist {
93
94      \__apos_define_shortcuts:nnn { #1 } { v = #1 } { pg-v = #1 }
95
96  }
97
98  \clist_map_inline:Nn \__apos_vanchors_clist {
99      \clist_map_inline:Nn \__apos_hanchors_clist {
100
101         \tl_clear:N \g_tmpa_tl
102
103         \tl_if_eq:nnTF { #1 } { vc }
104             { \tl_put_right:Nn \g_tmpa_tl { c } }
105             { \tl_put_right:Nn \g_tmpa_tl { #1 } }
```

```
106
107        \tl_if_eq:nnTF { ##1 } { hc }
108            { \tl_put_right:Nn \g_tmpa_tl { c } }
109            { \tl_put_right:Nn \g_tmpa_tl { ##1 } }
110
111        \__apos_define_shortcuts:Vnn
112            \g_tmpa_tl { v = #1, h = ##1 } { pg-v = #1, pg-h = ##1 }
113
114    }
115 }
```

In order to let the user set options outside the actual use of `\absput` and `\absputcoffin`, we introduce a wrapper:

```
116 \NewDocumentCommand \absposset { +m } {
117     \keys_set:nn { abspos } { #1 }
118 }
```

`\absput` This is the main command of the package. It takes a key–value list as its first optional argument, followed by the contents that are to be placed.

```
119 \NewDocumentCommand \absput { +O{ } +m } {
120     \group_begin:
```

First we handle the keys, updating variables holding coordinates, etc.

```
121     \keys_set:nn { abspos } { #1 }
```

Then the contents are typeset, with some additions, into a temporary coffin. The additions are that whitespace is stripped around the contents, and that we reset any overlay counters (in case they are modified by `\onslide` or `\pause`, or the like). If we're not using beamer, this last part is a no-op.

If width has been set, the contents is set in vertical mode (using the given width); otherwise, it is set in horizontal mode.

```
122     \tl_set:Nn \l_tmpa_tl {
123         \ignorespaces #2 \unskip
124         \apos_reset_overlay:
125     }
126     \tl_if_novalue:VTF \l_apos_width_tl {
127         \hcoffin_set:Nn \l_tmpa_coffin \l_tmpa_tl
128     } {
129         \vcoffin_set:Nnn \l_tmpa_coffin \l_apos_width_tl \l_tmpa_tl
130     }
```

Finally, actually place the typeset coffin on the canvas coffin.

```
131     \__apos_absput_coffin:N \l_tmpa_coffin
132     \group_end:
133 }
```

`\absputcoffin` A thin wrapper around the internal `\__apos_absput_coffin:N`. The only extra work done here is setting the keys.

```
134 \NewDocumentCommand \absputcoffin { +O{ } m } {
135     \group_begin:
136     \keys_set:nn { abspos } { #1 }
137     \__apos_absput_coffin:N #2
138     \group_end:
139 }
```

`\__apos_absput_coffin:N` An internal macro for placing a given coffin, using the options currently in effect (i.e., as configured by `\keys_set:nn` in some outer group). Used by `\absput` and `\absputcoffin`.

```
140 \cs_new:Npn \__apos_absput_coffin:N #1 {
```

If a scale has been provided, we scale the temporary coffin (with the same factor in both directions).

```
141     \tl_if_novalue:VF \l_apos_scale_tl {
142         \coffin_scale:Nnn #1 \l_apos_scale_tl \l_apos_scale_tl
143     }
```

If an angle has been provided, we rotate the temporary coffin.

```
144     \tl_if_novalue:VF \l_apos_angle_tl {
145         \coffin_rotate:Nn #1 \l_apos_angle_tl
146     }
```

We now join the temporary coffin to our canvas coffin, so it will be rendered at the correct position at shipout.

```
147     \coffin_gattach:NooNoonn
148         \g_apos_canvas_coffin
149             {\l_apos_on_hanchor_tl} {\l_apos_on_vanchor_tl}
150         #1
151             {\l_apos_hanchor_tl} {\l_apos_vanchor_tl}
152             {\l_apos_x_tl} {\l_apos_y_tl}
```

Ideally, what we just did should have been enough. However, until October 2021, the `\coffin_gattach:` command was not actually global,[3] so to have our modification take effect outside the current group also for older versions of l3coffins, we perform a final (normally redundant) global self-assignment before ending the `\absput` command.

```
153     \coffin_gset_eq:NN \g_apos_canvas_coffin \g_apos_canvas_coffin
154 }
```

```
155 ⟨/package⟩
```

---

[3]https://tex.stackexchange.com/questions/618198