Beamer theme `bgblocks`: absolute positioning in Beamer presentations *

Cédric Ware**

2018/02/03

Beamer theme `bgblocks` allows to position blocks of text or pictures at a specific absolute position on the slide, with a finer control on foreground/background than similar packages such as `textpos` or `eso-pic`.

1 Basic usage

1.1 Block definition and positioning

`\defbgblock`

Blocks are defined by using the macro `\defbgblock{`⟨*options*⟩`}{`⟨*content*⟩`}` in the preamble. Figure 1 shows an example of frame with a few blocks defined as explained below. The full example code is given section 1.5 on page 4.

Two more-or-less required options are `x=`⟨*dimen*⟩ and `y=`⟨*dimen*⟩, to specify where the block is positioned relative to the lower-left corner of the frame. For instance, `\defbgblock{x=0pt,y=0pt}{X}` will place an “X” centered in the frame corner.

The block’s position may be specified to be one of the corners of the block content, instead of its center. Typically, in the above case, option `above right` would place the text above and to the right of `(0pt,0pt)`, meaning that the lower-left corner of the text occupies this position. Available related options, in similar fashion to Ti*k*Z, are: `above`, `above left`, `above right`, `below`, `below left`, `below right`, `left` and `right`.

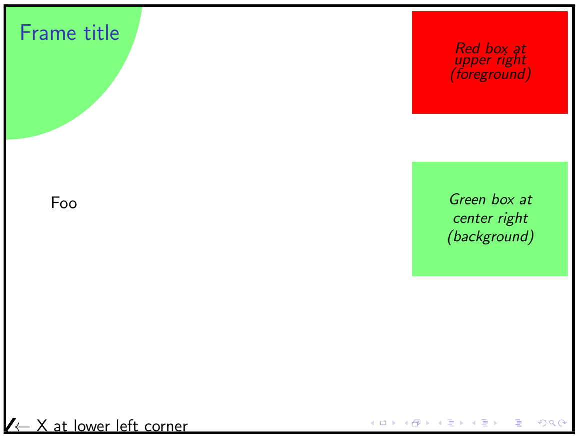

Frame title
Red box at upper right (foreground)
Foo
Green box at center right (background)
X ← X at lower left corner

Figure 1: Example of frame with a few blocks defined using the code shown in section 1.5.

---

[*] Version: 3.1.3

[**] cedric.ware@telecom-paristech.fr

Blocks can also easily be placed relative to other corners of the frame by calculating positions using dimensions `\bgtextblockheight` and `\bgtextblockwidth`[1]. Using a package such as `calc` for a convenient notation of operations, `\defbgblock{x=\bgtextblockwidth-1mm,y=\bgtextblockheight-1mm}{`⟨*...*⟩ will place its content 1mm below and 1mm to the left of the upper right corner of the frame.

Blocks can optionally be named, which allows fonts and colors to be specified using Beamer templates: `\defbgblock[`⟨*name*⟩`]{`⟨*...*⟩`}{`⟨*...*⟩`}` can be accessed through `\setbeamercolor{\bgblocktemplate{`⟨*name*⟩`}}`. Figure 1 shows an example of this, which lets a red `beamercolorbox` to be used as a block.

1.2 Background and foreground

By default, blocks are positioned “above the background”, meaning that they will be overlapped by the frame text but not by the background itself. However, the code shown in figure 1 has blocks defined with options “`background`” and “`foreground`”. Against an empty background, the difference isn’t immediately apparent. Figure 2 shows what happens using a blue frame-title background and a grid frame background.

`\setupbgblocks`

Note the `\setupbgblocks` after `\setbeamertemplate`; this recalculates the positions and “fore/background-ness” of all blocks. It must be invoked after any change in Beamer templates `background` and `footline`.

1.3 Disabling blocks for the title page and plain frames

`\disablebgblocks`

Sometimes the title page must not have the same decorations as regular frames. Blocks can be disabled locally to a TEX group using `\disablebgblocks`. This is shown in figure 3.

`\enablebgblocks`
`\enableallbgblocks`

It is also possible to invoke `\enablebgblocks{`⟨*block name list*⟩`}` to selectively enable only certain blocks. `\enableallbgblocks` is equivalent to `\enablebgblocks` invoked with the list of all defined blocks.

Blocks are always disabled in plain frames.

1.4 Compatibility with other packages and themes, and previous versions

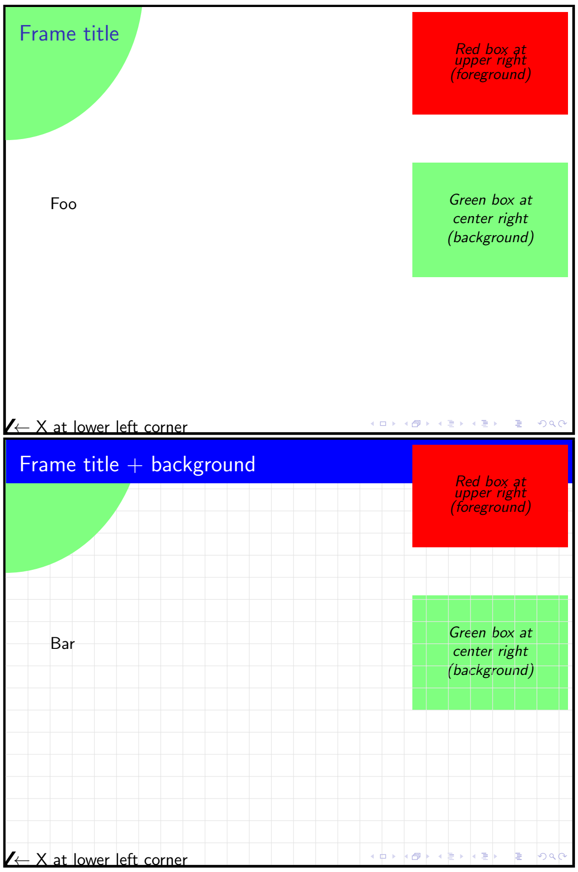
`bgblocks` version 3 and above is NOT compatible with previous versions. Version 3 is a complete reimplementation using `picture` instead of `pgfpicture`, mostly for performance reasons and some simplifications. PGF (version 2 and higher) is still required for `pgfkeys`.

---

[1] By default, these lengths are equal to `\paperheight` and `\paperwidth` minus space occupied by head/footlines and sidebars; package options may specify to ignore headlines etc—see section 2.4.1.

```
\begin{frame}{Frame title}
Foo
\end{frame}
```

```
\setbeamercolor{frametitle}%
               {bg=blue, fg=white}
\setbeamertemplate{background}[grid]
\setupbgblocks
\begin{frame}{Frame title + background}
Bar
\end{frame}
```



Frame title

Red box at upper right (foreground)

Foo

Green box at center right (background)

← X at lower left corner

Frame title + background

Red box at upper right (foreground)

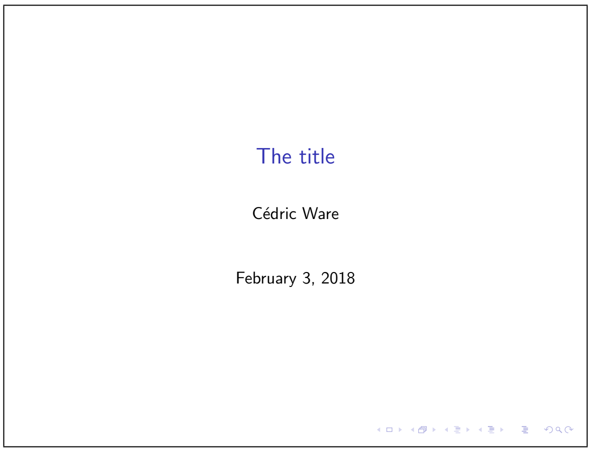
Bar

Green box at center right (background)

← X at lower left corner

Figure 2: Same blocks against non-empty frame and frame-title backgrounds. The red box, declared with the “`foreground`” option, is drawn on top of everything else, while the green box, as “`background`”, is covered by the background grid. The other blocks overlap the grid but not e.g. the frame title.

```
{
  \disablebgblocks
  \maketitle
}
```



The title

Cédric Ware

February 3, 2018

Figure 3: Disabling blocks for the title frame.

Beamer themes that modify the background and footline should work, provided that `\setupbgblocks` is executed after the last modification of said templates (see 2.1).

The only other potential interaction with other themes and packages is the handling of dimensions `\headheight` and `\footheight`. `bgblocks` assumes that these dimensions correspond to headline and footline heights.

1.5 Complete example code

```
\documentclass{beamer}
\useoutertheme{bgblocks}

%% Define simple blocks: an X in the lower-left corner, and a label nearby;
%% and a large light-green disc centered on the top-left corner (so that
%% only a quarter of the disc is visible).
\defbgblock{x=0pt, y=0pt}{\Huge X}
\defbgblock{x=1ex, y=0pt, above right}{$\leftarrow$ X at lower left corner}
\defbgblock{x=0pt, y=\bgtextblockheight}{%
  \resizebox{8cm}{8cm}{%
    \textcolor{green!50!white}{$\bullet$}%
  }%
}

%% Slightly more complex block: a red box whose top and right borders are 1mm
%% inside the top-right corner (requires package calc).  Use a block name so
%% as to demonstrate setting font and color.  Block is in the foreground.
\usepackage{calc}
\defbgblock[named block upper right]{%
  x=\bgtextblockwidth-1mm,
  y=\bgtextblockheight-1mm,
  foreground,
  below left}{%
  \begin{beamercolorbox}
    [sep=2em, wd=10em, center]{\bgblocktemplate{named block upper right}}
    Red box at upper right (foreground)
  \end{beamercolorbox}%
}
\setbeamercolor{\bgblocktemplate{named block upper right}}{bg=red}
\setbeamerfont{\bgblocktemplate{named block upper right}}
              {size=\small,family=\itshape}

%% Likewise light-green box at center right, in the background.
\defbgblock[named block center right]{%
  x=\bgtextblockwidth-1mm,
  y=0.5\bgtextblockheight,
  background,
  left}{%
  \begin{beamercolorbox}
    [sep=2em, wd=10em, center]{\bgblocktemplate{named block center right}}
    Green box at center right (background)
  \end{beamercolorbox}%
}
\setbeamercolor{\bgblocktemplate{named block center right}}{bg=green!50!white}
\setbeamerfont{\bgblocktemplate{named block center right}}
              {size=\small,family=\itshape}

%% Meta-information, then the document itself.
\title{The title}
\author{C\'edric Ware}

\begin{document}
{
  % Disable blocks temporarily for the title page.
  \disablebgblocks
  \maketitle
}

%% Normal frame.
\begin{frame}{Frame title}
Foo
\end{frame}

%% Change frame: add a grid background and a colored-band frame title.
%% One must call \setupbgblocks to recalculate block positions.
\setbeamertemplate{background}[grid]
\setbeamercolor{frametitle}%
               {bg=blue, fg=white}
\setupbgblocks
\begin{frame}{Frame title + background}
Bar
\end{frame}

\end{document}
```

2 Advanced usage

2.1 Working principle

`bgblocks` operates by inserting a `picture` environment in three places among the regular Beamer templates:

- just before `background` (background blocks);
- just after `background` (above-background blocks);
- just after `footline` (foreground blocks).

`\setupbgblocks`

This incurs a limitation: every time one of these templates is reset, the hooks must be re-inserted, using macro `\setupbgblocks`. (This macro also affects selective enabling of blocks, see section 2.3.)

2.2 Block definition

`\defbgblock` The blocks are defined by the `\defbgblock` macro, which has two alternative syntaxes:

- `\defbgblock[`⟨*optional name*⟩`]{`⟨*keys*⟩`}{`⟨*content*⟩`}`
- `\defbgblock[`⟨*optional name*⟩`]{`⟨*keys,*`content={`⟨*content*⟩`}`⟩`}`

The `{`⟨*keys*⟩`}` are handled by `pgfkeys`. The first syntax may be clearer; the second is more logical, and allows to manipulate the content through `pgfkeys` mechanisms. For instance, using `content/.expanded={`⟨*content*⟩`}` expands the content before defining the block.

Unnamed blocks are implicitly given a number as their name, which is the current number of defined blocks.

Blocks should be defined in the preamble. It is possible to define them later, but they will have to be enabled manually (see section 2.3).

2.2.1 Predefined block keys

- `x=`⟨*dimen*⟩, `y=`⟨*dimen*⟩ specify the position of the block, as lengths relative to the lower corner of the text block. These ⟨*dimen*⟩s may be expressions if the `calc` package is loaded.
- `raw picture element` indicates that the block content is to be inserted directly into a `picture` environment, and incorporates its own `\put` or `\multiput` directives. This key is on by default, but specifying `x` or `y` automatically disables it.
- `center` (default), `above`, `below`, `left`, `right`, `above left`, `above right`, `below left`, `below right`: specify where the block should be typeset relative to the given position. For example, in Ti*k*Z-like fashion, `above right` places the block above and to the right of the (x, y) coordinates given; in other words, the given (x, y) is the position of the lower-left corner of the block. (Unlike Ti*k*Z, however, it is not possible to shift the position using "`above=`⟨*dimen*⟩". Use `x`, `y` and package `calc`.)
- `textpos=`⟨*pos*⟩ is an alternative to the `above right` etc. options: it specifies the optional argument of `picture`-environment macro `\makebox` for this block. For instance, `textpos=lb` makes the block content to be typeset as `\makebox[lb]{`⟨*content*⟩`}`, with the same result as `above right`.
- `foreground`, `background`, `above background` (default): specify the fore/background placement of this block.

- Custom keys can be defined in any way allowed by `pgfkeys`; keys and their values will be available within the block content through `\theblockkeys` and `\theblock` macros (wrappers around `\pgfqkeys` and `\pgfvalueof`).

2.2.2 Block contents and templates

`\bgblocktemplate`

For each block, a Beamer template named `\bgblocktemplate{`⟨*block name*⟩`}` is defined. Two styles are generated: "`empty`" doesn't typeset the block content; "default" typesets it either in a `\makebox`, or directly if key **`raw picture element`** was used. This makes it easy to locally disable a block by using `\setbeamertemplate{\bgblocktemplate{`⟨*name*⟩`}}[empty]`, or even changing its appearance by creating other styles. Keep in mind, though, that the template is expanded in a `picture` environment; the default style handles the required `\put` and `\makebox` commands.

Within the block content, `\theblockname` expands to the current block name, `\theblockkeys` and `\theblock` provide access to all defined keys, and some flags are positioned. See the source of `\selectblock` and `\@defbgblock` starting page 13 for details.

2.3 Selective enabling of blocks

The three `picture` environments each typesets a sub-list of blocks generated according to block fore/background options. The lists are initialized at the end of the preamble, which is why blocks defined in the document body won't appear automatically.

`\enablebgblocks`

This scheme allows selective enabling of blocks: macro `\enablebgblocks{`⟨*block1,block2,...*⟩`}` activates the blocks named in its argument by dispatching them into the three sub-lists. The sub-lists are emptied before this dispatching, so that only the given blocks will be typeset thereafter.

`\enableallbgblocks`
`\disablebgblocks`
`\bgblocklist`

Shortcuts are available to enable all or none of the defined blocks: `\enableallbgblocks` and `\disablebgblocks`. A comma-separated list of block names is maintained in macro `\bgblocklist`. This list is not affected other than by `\defbgblock`, but can be redefined so that `\enableallbgblocks` enables only some of the blocks; the "hidden" blocks will still be defined and can be enabled manually by `\enablebgblocks`.

The effect of these macros are local to a TeX-group, which allows to change the blocks temporarily, and restore them at the end of the group; see figure 3 for an example. Starred versions `\enablebgblocks*{`⟨...⟩`}`, `\enableallbgblocks*` and `\disablebgblocks*` change the sub-lists globally.

Macro `\setupbgblocks` implicitly performs `\enableallbgblocks`. In fact, it disables then re-enables all blocks as a precaution due to implementation details. Starred version `\setupbgblocks*` leaves them disabled.

2.4 Package options

Theme `bgblocks` has two main categories of package options, relative on the one hand to the default fore/background placement of the blocks; and, on the other hand, whether or not to take into account the headline, footline and sidebars.

`\bgblockglobalkeys`

Options can be set in the usual way, loading the theme using `\useoutertheme[`⟨*options*⟩`]{bgbl` or, more reliably, by invoking `\bgblockglobalkeys{`⟨*options*⟩`}` before defining the blocks.[2]

2.4.1 Head/footline and sidebar options

By default, blocks are positioned relative to the lower-left corner of the "text block", meaning the area of the slide excluding the headline, footline and sidebars, but including the frame title.[3]

It is possible to selectively ignore these frame elements using options:

- `ignore headline` (or `ignore headline=true`) incorporates the headline's height into the text block; `ignore headline=false` restores the default behavior.
- Likewise `ignore footline`, `ignore left sidebar` and `ignore right sidebar`; for the footline and left sidebar, this also shifts the coordinates' reference to the slide's edges.
- `ignoreheadfoot` is an alias for `ignore headline,ignore footline`; likewise `ignoreleftright` for both sidebars.

2.4.2 Fore/background default placement

Blocks whose definition doesn't include a fore/background specification are placed in the above-background position by default. This can be changed using the `defaultz` key, whose possible values are `foreground`, `background` and `above background`.

- `\bgblockglobalkeys{defaultz=foreground}` (or just `\bgblockglobalkeys{foreground}`) places blocks in the foreground by default.

---

[2] Options are implemented using the powerful `pgfkeys` package. However, LaTeX's default mechanism interferes (deleting spaces, notably), which may prevent some key uses. `\bgblockglobalkeys` doesn't have this problem. Package `pgfopts` is a better solution, but I didn't want to make it a requirement for using `bgblocks`.

[3] Used with some Beamer themes, e.g. `sidebar`, it may appear that `bgblocks`' text block also excludes the frame title. This is because the frame title is actually pushed up over the headline.

- `\bgblockglobalkeys{defaultz=background}` (or just `\bgblockglobalkeys{background}`) places blocks in the background by default.
- `\bgblockglobalkeys{defaultz=above background}` (or `\bgblockglobalkeys{above background}`, for completeness), restores the default behavior of placing blocks above the background by default.

These options have no effect on blocks that have already been defined.

3 Code documentation

3.1 Housekeeping and options

`bgblocks` makes extensive use of `pgfkeys`: both for global options, which use the hierarchy `/bg block/@global`, and each block parameters, stored in `/bg block/`⟨*name*⟩`/`⟨*...*⟩. Shortcuts `\bgblockkeys` and `\bgblockglobalkeys` access this hierarchy.

In addition, `\bgblockkeys*` ensures keys are stored *globally*, that is, persist outside of a TEX-group.

```

%% Use pgfkeys for options and block parameters.
\RequirePackage{pgfkeys}

%% Utility macros to access keys in the proper hierarchy.
\newcommand\bgblockkeysvalueof[2]{\pgfkeysvalueof{/bg block/#1/#2}}
\newcommand\bgblockifdefined[1]{\pgfkeysifdefined{/bg block/#1/@name}}

%% \bgblockkeys: macro to store/execute keys in the proper hierarchy.
%% \bgblockkeys*: keys persist outside of a TeX-group.
\newcommand*\bgblockkeys{\@ifstar\@bgblockkeys@star\@bgblockkeys@nostar}
\newcommand\@bgblockkeys@nostar[2]{%
  \pgfqkeys{/bg block/#1}{#2}%
}
\newcommand\@bgblockkeys@star[2]{%
  \begingroup
    \ifnum\the\globaldefs>0\relax \else \globaldefs=1\fi
    \@bgblockkeys@nostar{#1}{#2}%
  \endgroup
}

%% Same for global options.
\newcommand\bgblockglobalkeys[1]{\bgblockkeys{@global}{#1}}

```

Define a number of flags and default values, then declare global keys that will handle options.

```
%% Flags to ignore/allow head/footlines and left/right sidebars.
\newif\ifbgblock@ignoreheadline
\newif\ifbgblock@ignorefootline
\newif\ifbgblock@ignoreleftsidebar
\newif\ifbgblock@ignorerightsidebar

%% Prepare global keys before processing options.
\bgblockglobalkeys{%
  %
  % Default block fore/background position.
  %
  defaultz/.is choice,
  defaultz/foreground/.code={\def\@bg@defaultz{foreground}},
  defaultz/background/.code={\def\@bg@defaultz{background}},
  defaultz/above background/.code={\def\@bg@defaultz{above background}},
  %
  % Handling of head/footlines, sidebars.
  %
  ignore headline/.is if=bgblock@ignoreheadline,
  ignore footline/.is if=bgblock@ignorefootline,
  ignore left sidebar/.is if=bgblock@ignoreleftsidebar,
  ignore right sidebar/.is if=bgblock@ignorerightsidebar,
  %
  % Aliases, especially no-space versions, because \usepackage strips them.
  %
  foreground/.style={defaultz=foreground},
  background/.style={defaultz=background},
  above background/.style={defaultz=above background},
  abovebackground/.style={above background},
  ignoreheadline/.is if=bgblock@ignoreheadline,
  ignorefootline/.is if=bgblock@ignorefootline,
  ignoreleftsidebar/.is if=bgblock@ignoreleftsidebar,
  ignorerightsidebar/.is if=bgblock@ignorerightsidebar,
  ignore headfoot/.style={ignore headline,ignore footline},
  ignoreheadfoot/.style={ignore headfoot},
  ignore leftright/.style={ignore left sidebar,ignore right sidebar},
  ignoreleftright/.style={ignore leftright},
  ignore sidebar left/.style={ignore left sidebar},
  ignore sidebar right/.style={ignore right sidebar},
  %
  % Default values.
  %
  defaultz=above background,
  ignore headline=false,
  ignore footline=false,
  ignore left sidebar=false,
  ignore right sidebar=false,
}

```

Process options. Don't use LaTeX's normal mechanism, otherwise `pgfkeys` gets into expansion problems. Instead, expand the global option list and pass it to `\bgblockglobalkeys`.

```
%% Process options all at once, suppressing LaTeX's normal mechanism.
\edef\@bg@pkgname{\@currname}
\edef\@bg@alloptions{\@ptionlist{\@currname.sty}}
\expandafter\bgblockglobalkeys\expandafter{\@bg@alloptions}
\DeclareOption*{}
\ProcessOptions

```

Miscellaneous utility functions used later.

```
%% Generate an error when obsolete syntax is detected.
\newcommand*\@breakcompat[1]{%
  \PackageError{\@bg@pkgname}{%
    Sorry, you have used #1,
    no longer suppported with version 3 and above.  Please refer
    to the documentation of package \@bg@pkgname
  }%
}

%% Append an element to a comma-separated list.
%% Two versions: global assignment, or local to TeX-group.
\newcommand*\@bg@appendtolist[2]{%
  \ifx\@empty#1\g@addto@macro#1{#2}\else\g@addto@macro#1{,#2}\fi
}
\newcommand*\@bg@appendtolist@local[2]{%
  \ifx\@empty#1\edef#1{#2}%
  \else
    \edef\@bg@tmplist{#1,#2}%
    \edef#1{\@bg@tmplist}%
  \fi
}

%% Redefine a macro so that its current definition becomes global.
\newcommand*\@bg@makeglobal[1]{%
  \begingroup
    \expandafter\def\expandafter\@bg@temp\expandafter{#1}%
    \expandafter\gdef\expandafter#1\expandafter{\@bg@temp}%
  \endgroup
}


%% Wrapper for \makebox.
\newcommand\@bg@makebox@zerodim[2]{\makebox(0,0)[#1]{#2}}

%%
%% Housekeeping is done.  The rest is available only in presentation mode.
%%
```

```
\mode<presentation>

```

3.2 Block definition: interface

Macro `\defbgblock` is mainly a syntax-adjusting wrapper around the main definition command `\@defbgblock`, explained below. The wrapper:

- maintains the current block number;
- forbids `\defbgblock*` (characteristic of old-version syntax) and a few other obsolete macros;
- makes sure the block name is completely expanded;
- translates from syntax `\defbgblock{`⟨*options*⟩`}{`⟨*content*⟩`}` to `\defbgblock{`⟨*options*, *`content={`*⟨*content*⟩*`}`*⟩`}`.[4]

```
%% Counter: current block number.
\newcounter{bg@blocknumber}

%% Interface to define blocks: mostly a series of wrappers to main macro.
%% Wrappers give default block name, make sure block name is expanded,
%% handle syntax \defbgblock{...}{...} vs \defbgblock{...,content={...}}.
\newcommand\defbgblock{%
  \stepcounter{bg@blocknumber}%
  \@ifstar
     {\@breakcompat{an obsolete form of \string\defbgblock}}%
     \@defbgblock@expandname
}
\newcommand\@defbgblock@expandname[2][\the\c@bg@blocknumber]{%
  \edef\@bg@tmp@blockname{#1}%
  \expandafter\@defbgblock@testbrace\expandafter{\@bg@tmp@blockname}{#2}%
}
\def\@defbgblock@testbrace#1#2{%
  \@ifnextchar\bgroup{\@defbgblock@withbrace{#1}{#2}}{\@defbgblock{#1}{#2}}%
}
\def\@defbgblock@withbrace#1#2#3{\@defbgblock{#1}{#2,content={#3}}}

%% While we're at it, define legacy macro as another test of obsolete syntax.
\newcommand\defbgblockpgf{\@breakcompat{obsolete macro \string\defbgblockpgf}}
\newcommand\appendbgblock{\@breakcompat{obsolete macro \string\appendbgblock}}
\newcommand\prependbgblock{\@breakcompat{obsolete macro \string\prependbgblock}}

```

---

[4] The latter syntax, using `\defbgblock` with ⟨*content*⟩ given along with the options, can be used directly. It can be especially useful if ⟨*content*⟩ must be expanded, which can then be performed directly by `pgfkeys` using, for example: `\defbgblock {`⟨..., `content/.expanded={`⟨*content*⟩`}`⟩`}`.

Also useful, define some utilities and flags, which are automatically set before typesetting each block, or by invoking `\selectbgblock{`⟨*block name*⟩`}`.

```
%% Some utilities and flags needed to define blocks.
\newcommand*\bgblocktemplate[1]{bg block #1}% Beamer template for given block.
\newif\ifblockpict       % True if block is raw picture element, no \put needed.
\newif\ifblockbackground % True if block is in the background.
\newif\ifblockforeground % True if block is in the foreground.

\newcommand*\selectbgblock[1]{% Set flags for given block.
  %
  % Check that block is defined.
  %
  \bgblockifdefined{#1}{}{%
    \PackageError{\@bg@pkgname}{Bg block ‘#1’ not defined.}%
  }%
  %
  % Set macros for current block name, pgfkeys shortcuts.
  %
  \def\theblockname{#1}%
  \def\theblockkeys{\bgblockkeys{#1}}%
  \def\theblock{\bgblockkeysvalueof{#1}}%
  %
  % Set \ifblock<name> flags using the block-specific \ifs created
  % by \defbgblock.
  %
  \csname if@bg@#1@pict\endcsname
    \blockpicttrue
  \else
    \blockpictfalse
  \fi
  \csname if@bg@#1@background\endcsname
    \blockbackgroundtrue
  \else
    \blockbackgroundfalse
  \fi
  \csname if@bg@#1@foreground\endcsname
    \blockforegroundtrue
  \else
    \blockforegroundfalse
  \fi
}

```

3.3 Block definition: main command

The main definition command `\@defbgblock{`⟨*name*⟩`}{`⟨*options+content*⟩`}` defines a block. It begins by adding the block name to the list of all blocks.

Then it creates block-specific `\if` flags that denote some of the block's options (text or raw picture element, foreground or background). Relevant values are stored by calling `pgfkeys`. Then Beamer templates with the block content are defined.

```
%%
%% Main definition command \@defbgblock.
%% Uses global comma-separated list of block names: \bgblocklist
%% and temporary macros for current block position.
%%
\newcommand*\bgblocklist{}
\newlength\@bg@tmp@posx
\newlength\@bg@tmp@posy
\newcommand\@bg@tmp@textpos{}

\newcommand\@defbgblock[2]{%
  %
  % Add block name to list of all blocks, create block-specific flags.
  %
  \@bg@appendtolist{\bgblocklist}{#1}%
  \expandafter\newif\csname if@bg@#1@pict\endcsname
  \csname @bg@#1@picttrue\endcsname
  \expandafter\newif\csname if@bg@#1@background\endcsname
  \expandafter\newif\csname if@bg@#1@foreground\endcsname
  %
  % Process block options.
  %
  \bgblockkeys*{#1}{%
    @name/.initial={#1},
    raw picture element/.default=true,
    raw picture element/.code={\csname @bg@#1@pict##1\endcsname},
    content/.value required,
    content/.initial=,
    x/.value required,
    y/.value required,
    x/.style={raw picture element=false,posx=##1},
    y/.style={raw picture element=false,posy=##1},
    posx/.initial=0pt,
    posy/.initial=0pt,
    textpos/.initial=,
    center/.style={textpos={}},
    above/.style={textpos=b},
    above left/.style={textpos=rb},
    above right/.style={textpos=lb},
    below/.style={textpos=t},
    below left/.style={textpos=rt},
    below right/.style={textpos=lt},
    left/.style={textpos=r},
    right/.style={textpos=l},
    center/.value forbidden,
    above/.value forbidden,
    above left/.value forbidden,
    above right/.value forbidden,
    below/.value forbidden,
    below left/.value forbidden,
    below right/.value forbidden,
    left/.value forbidden,
    right/.value forbidden,
    background/.code={%
      \csname @bg@#1@backgroundtrue\endcsname
      \csname @bg@#1@foregroundfalse\endcsname
    },
    foreground/.code={%
      \csname @bg@#1@backgroundfalse\endcsname
      \csname @bg@#1@foregroundtrue\endcsname
    },
    above background/.code={%
      \csname @bg@#1@backgroundfalse\endcsname
      \csname @bg@#1@foregroundfalse\endcsname
    },
    background/.value forbidden,
    foreground/.value forbidden,
    above background/.value forbidden,
    \@bg@defaultz,
    #2
  }%
```

The new Beamer templates are defined in two styles: "`empty`" with no content, useful to disable a block locally; and "`default`" that either typesets the content at the right position in a `\makebox` (via `\@bg@makebox@zerodim`, expanding the argument), or inserts it directly into the picture if option `raw picture element` is set.

```
%
% Create Beamer templates; two styles: empty, or with block content.
%
\defbeamertemplate{\bgblocktemplate{#1}}{empty}{}%
\defbeamertemplate*{\bgblocktemplate{#1}}{default}{%
  \ifblockpict
    \theblock{content}%
  \else
    \setlength{\@bg@tmp@posx}{\theblock{posx}}%
    \setlength{\@bg@tmp@posy}{\theblock{posy}}%
    \theblockkeys{textpos/.get=\@bg@tmp@textpos}%
    \put(\strip@pt\@bg@tmp@posx,\strip@pt\@bg@tmp@posy){%
      \expandafter\@bg@makebox@zerodim\expandafter{\@bg@tmp@textpos}{%
        \usebeamercolor*{\bgblocktemplate{#1}}%
        \usebeamerfont*{\bgblocktemplate{#1}}%
        \theblock{content}%
      }%
      }%
    \fi
  }%
```

A color and a font are set for this template, inherited from color/font `bg blocks` (plural), defined globally.

```
  \setbeamercolor*{\bgblocktemplate{#1}}{parent=bg blocks}%
  \setbeamerfont*{\bgblocktemplate{#1}}{parent=bg blocks}%
}
%% End of \@defbgblock.

\setbeamercolor*{bg blocks}{parent=normal text}
\setbeamerfont*{bg blocks}{size=\normalsize,parent=normal text}

```

3.4 Block typesetting

Macro `\@bg@typesetblocks{`⟨*offset x*⟩`}{`⟨*offset y*⟩`}{`⟨*list*⟩`}` typesets a list of blocks in a `picture` environment with the given (x, y) offset. Except in plain frames, where this macro does nothing. All coordinates are given in TeX points: `\unitlength` is temporarily set to `1pt`, so that they can be derived from actual lengths using `\strip@pt`.

```
%% Typeset a list of blocks in a picture with given offsets (x,y).
\newcommand*\@bg@typesetblocks[3]{% Arguments: {x}{y}{comma-separated-list}.
  \ifbeamer@plainframe
    % Disable bg blocks in plain frames.
  \else
    \begingroup
      \setlength{\unitlength}{1pt}%
      \begin{picture}(0,0)(#1,#2)
        \@for\@block:=#3\do{%
          \selectbgblock{\@block}%
          \usebeamertemplate{\bgblocktemplate{\@block}}%
        }%
      \end{picture}%
    \endgroup
  \fi
}

```

These blocks are to be typeset in Beamer's `background` and `footline` templates. We define "hooks", one for each of foreground, background and above-background blocks, to be added to those templates by `\setupbgblocks`, defined below.

Each of these hooks invokes `\@bg@typesetblocks` on the list of blocks in the corresponding category at a precalculated position—all maintained by `\setupbgblocks` as well. `\if` flags are used so that blocks will be typeset only once even if hooks were to be added multiple times.[5]

Note that `\bgblockfootlinehook` calls `\hskip` before typesetting the block picture, so that it is always invoked at the right border; we need this to calculate the positions. The actual horizontal stretch is `0.00001fill`, so that it "wins" against the `\hfil` that's already pushing the footline, but stays negligible if there is already a `\hfill` in the footline that already does the work.

This trick doesn't work with `\bgblockabovebackgroundhook` because Beamer typesets the `background` template in an `\hbox`. We thus have to calculate the background width.

```
%%
%% For each of foreground, background and above-background, define a
%% block list, assorted flags and positions, and a macro to typeset the
%% corresponding picture.  Flags and positions will be precalculated
%% and the hooks added to Beamer's templates by \setupbgblocks below.
%%
\newcommand*\bgblocklistforeground{}% Names of foreground blocks.
\newif\if@bg@do@foreground@         % Flag to typeset blocks only once.
\newlength\@bg@pict@foreground@x    % X-offset for the picture environment.
\newlength\@bg@pict@foreground@y    % Y-offset likewise.
\newcommand*\bgblockfootlinehook{%
  \if@bg@do@foreground@
    \hskip 0pt plus 0.00001fill% Skip to right border without breaking a \hfill.
    \@bg@typesetblocks{\strip@pt\@bg@pict@foreground@x}%
                       {\strip@pt\@bg@pict@foreground@y}%
                       {\bgblocklistforeground}%
    \@bg@do@foreground@false
  \fi
}

%% Same for background blocks.
\newcommand*\bgblocklistbackground{}
\newif\if@bg@do@background@
\newlength\@bg@pict@background@x
\newlength\@bg@pict@background@y
\newcommand*\bgblockbackgroundhook{%
  \if@bg@do@background@
    \@bg@typesetblocks{\strip@pt\@bg@pict@background@x}%
                       {\strip@pt\@bg@pict@background@y}%
                       {\bgblocklistbackground}%
    \@bg@do@background@false
  \fi
}
```

---

[5] Hooks may end up multiple times if `\setupbgblocks` or `\enablebgblocks` are invoked several times in the same TeX-group, which can't be avoided in practice if the `footline` or especially the `background` template is changed.

```

%% Same for above-background blocks.
\newcommand*\bgblocklistabovebackground{}
\newif\if@bg@do@abovebackground@
\newlength\@bg@pict@abovebackground@x
\newlength\@bg@pict@abovebackground@y
\newcommand*\bgblockabovebackgroundhook{%
  \if@bg@do@abovebackground@
    \@bg@typesetblocks{\strip@pt\@bg@pict@abovebackground@x}%
                      {\strip@pt\@bg@pict@abovebackground@y}%
                      {\bgblocklistabovebackground}%
    \@bg@do@abovebackground@false
  \fi
}

```

3.5 Block selection and setup

The block lists and picture positions must be calculated at the beginning of the document, and each time the `background` template is changed or a foreground/background block setting is modified.

`\enablebgblocks`

First, macro `\enablebgblocks{`⟨*block1,block2,...*⟩`}` dispatches the given blocks into the appropriate fore/background lists. The assignment is local to the TeX-group unless the starred version `\enablebgblocks*{`⟨*...*⟩`}` is used. This allows to selectively enable certain blocks, and have the block lists revert to normal at the end of the TeX-group. Shortcuts are provided to enable or disable all blocks: `\enableallbgblocks` and `\disablebgblocks` are equivalent to `\enablebgblocks` called with `\bgblocklist` or with an empty list; starred versions likewise.

`\enableallbgblocks`

`\disablebgblocks`

```
%%
%% \enablebgblocks: dispatch the given block names into the appropriate
%% fore/background lists.  Starred version makes the assignment global.
%%
\newcommand*\enablebgblocks{\@ifstar\@enablebgblocks@global\@enablebgblocks}
\newcommand*\@enablebgblocks[1]{%
  % Empty lists, then dispatch given blocks using auxiliary macro
  % \@bg@addblocktolist, defined below.
  \def\bgblocklistbackground{}%
  \def\bgblocklistforeground{}%
  \def\bgblocklistabovebackground{}%
  \@for\@block:=#1\do{%
    \expandafter\@bg@addblocktolist\expandafter{\@block}%
  }%
}
\newcommand*\@enablebgblocks@global[1]{%
  % Starred version: same as \@enablebgblocks but make list definitions global.
  \@enablebgblocks{#1}%
  \makeglobal\bgblocklistbackground
  \makeglobal\bgblocklistforeground
  \makeglobal\bgblocklistabovebackground
}

%% Shortcuts to enable or disable all blocks at once.
\newcommand*\enableallbgblocks{%
  \@ifstar{\enablebgblocks*{\bgblocklist}}{\enablebgblocks{\bgblocklist}}%
}
\newcommand*\disablebgblocks{%
  \@ifstar{\enablebgblocks*{}}{\enablebgblocks{}}%
}

```

An auxiliary macro handles adding one given block name to the appropriate list, depending on its foreground/background flags. This macro is separate to make it easier to expand the block name.

```
%% Auxiliary macro: add one block to relevant list (local to TeX-group).
\newcommand*\@bg@addblocktolist[1]{%
  \selectbgblock{#1}%
  \ifblockforeground
    \@bg@appendtolist@local{\bgblocklistforeground}{#1}%
  \else\ifblockbackground
    \@bg@appendtolist@local{\bgblocklistbackground}{#1}%
  \else
    \@bg@appendtolist@local{\bgblocklistabovebackground}{#1}%
  \fi\fi
}%

```

`\setupbgblocks`

Then, the actual setup is performed in macro `\setupbgblocks`, which inserts the required hooks into the templates and invokes the position-calculating macro defined below. As a precaution, it disables all blocks before computing positions, then re-enables all blocks; for convenience, a starred version is provided that leaves all blocks disabled.

`\setupbgblocks` must be invoked every time the Beamer template `background` or `footline` is reset. Its effect is local to the TeX-group.

```
%%
%% Main block-setup macro.  Calls macro to compute positions, inserts hooks.
%% Disables then re-enables all blocks; starred version leaves them disabled.
%%
\newcommand*\setupbgblocks{%
  \@ifstar{\@setupbgblocks}{\@setupbgblocks\enableallbgblocks}%
}
\newcommand*\@setupbgblocks{%
  %
  % Empty all lists.  Then compute positions, which is better performed
  % without any block typeset in the background template.  (Shouldn't
  % make a difference, but you never know...)
  %
  \disablebgblocks
  \@bg@computepositions
```

Inserting hooks into the templates is somewhat tricky if successive calls to `\setupbgblocks` still must result in the blocks being typeset only once.

Hook `\bgblockfootlinehook`, which typesets foreground blocks, is inserted after the footline, so that these blocks are typeset last and overlap everything. However, flag `\if@bg@do@foreground@` is enabled not before the hook, but before the *footline* itself, so that the `\@bg@do@foreground@false` at the end of `\bgblockfootlinehook` suppresses any subsequent hooks.

Likewise, `\if@bg@do@abovebackground@` is enabled before the background; and `\if@bg@do@background@` after the background, for blocks in the next frame.

All flags are initialized to `true` after hooks are inserted.

```
  %
  % Insert hooks.  Also enable relevant flags; the order is important
  % so that successive calls to \setupbgblocks still result in the
  % blocks being typeset only once.
  %
  \addtobeamertemplate{footline}%
                      {\@bg@do@foreground@true}%
                      {\bgblockfootlinehook}%
  \addtobeamertemplate{background}%
                      {\bgblockbackgroundhook\@bg@do@abovebackground@true}%
                      {\bgblockabovebackgroundhook\@bg@do@background@true}%
  \@bg@do@foreground@true
  \@bg@do@background@true
  \@bg@do@abovebackground@true
}

```

3.6 Block positioning

In order to have absolute positions, the coordinates of the `picture`s that typeset the blocks must be offset according to where they are inserted on the page. This depends on the template contents: for example, if the background is empty, the background-block picture will be typeset at the top-left corner; whereas if the background covers the whole frame, the picture will end up at the *lower-left* corner (and the above-background picture at the lower-right corner).

`\bgtextblockheight`
`\bgtextblockwidth`

First, define `\bgtextblockheight` and `\bgtextblockwidth`, the dimensions of the "textblock": the whole slide by default, minus the head/footline and sidebars depending on the package options. Also define a temporary box to typeset the `background` template in, and specific lengths to handle the `4pt` Beamer manually adds above the footline.

```
%% Define height/width of the text block.
\newlength\bgtextblockheight
\newlength\bgtextblockwidth

\newbox\@bg@tempbox                  % Temporary box for background template.
\newlength\@bg@actualfootheight      % Temporary dimen for \footheight...
\newlength\@bg@beamerfootoffset      % ... minus Beamer footline offset.
\setlength\@bg@beamerfootoffset{4pt}
```

Then macro `\@bg@computepositions` calculates all lengths and offsets. It needs to typeset the `background` template to get its actual size, without the blocks—but any blocks that are enabled at this point will also be typeset. Since they are contained in `picture`s of zero size, it shouldn't make a difference; however, prudence suggests that this macro be called with blocks disabled.

```
%%
%% Calculate picture coordinate offsets.
%% This macro should be called with all blocks disabled.
%%
\newcommand*\@bg@computepositions{%
  %
  % Begin with calculating the actual footline height.
  %
  \setlength{\@bg@actualfootheight}{\footheight}%
  \addtolength{\@bg@actualfootheight}{-\@bg@beamerfootoffset}%
  %
  % Start the textblock with \paperheight and \paperwidth.  \headheight etc.
  % will be subtracted later depending on options.
  %
  \setlength{\bgtextblockheight}{\paperheight}%
  \setlength{\bgtextblockwidth}{\paperwidth}%
  %
  % Foreground blocks come after footline, always at lower-right corner.
  %
  \setlength{\@bg@pict@foreground@x}{\paperwidth}%
  \setlength{\@bg@pict@foreground@y}{0pt}%
  %
  % Background blocks are before background, at lower-left corner of
  % the *background*; this will be at left border so X=0, and Y is
  % \paperheight minus background height.  Typeset background to
  % find out its dimensions.
  %
  \setbox\@bg@tempbox=\hbox{\usebeamertemplate***{background}}%
  \setlength{\@bg@pict@background@x}{0pt}%
  \setlength{\@bg@pict@background@y}{\paperheight}%
  \addtolength{\@bg@pict@background@y}{-\ht\@bg@tempbox}%
  %
  % Likewise, above-background picture is at lower-right corner of the
  % background, so X=background width.  Y is the same as for background
  % blocks, no need to calculate it.
  %
  \setlength{\@bg@pict@abovebackground@x}{\wd\@bg@tempbox}%
  %
  % If headline not ignored, subtract its height from textblock height.
  %
  \ifbgblock@ignoreheadline
  \else
    \addtolength{\bgtextblockheight}{-\headheight}%
  \fi
  %
  % If footline not ignored, shift Y coordinates by its height and
  % subtract its height from textblock height.
  %
  \ifbgblock@ignorefootline
  \else
    \addtolength{\bgtextblockheight}{-\@bg@actualfootheight}%
    \addtolength{\@bg@pict@foreground@y}{-\@bg@actualfootheight}%
    \addtolength{\@bg@pict@background@y}{-\@bg@actualfootheight}%
  \fi
  %
  % Same for sidebars, with X coordinates shifted by width of left sidebar.
  %
  \ifbgblock@ignorerightsidebar
  \else
    \addtolength{\bgtextblockwidth}{-\beamer@rightsidebar}%
  \fi
  \ifbgblock@ignoreleftsidebar
  \else
    \addtolength{\bgtextblockwidth}{-\beamer@leftsidebar}%
    \addtolength{\@bg@pict@foreground@x}{-\beamer@leftsidebar}%
    \addtolength{\@bg@pict@background@x}{-\beamer@leftsidebar}%
    \addtolength{\@bg@pict@abovebackground@x}{-\beamer@leftsidebar}%
  \fi
  %
  % Above-background Y is the same as for background Y, copy it.
  %
  \setlength{\@bg@pict@abovebackground@y}{\@bg@pict@background@y}%
}

```

Finally, the blocks are enabled by inserting `\setupbgblocks` after Beamer finishes setting up its templates at the beginning of the document. Then restore `\mode<all>`.

```
\g@addto@macro\beamer@lastminutepatches{\setupbgblocks}

```

```
\mode<all>
```