

AcroTeX.Net

The aeb_tilebg Package Creating Tiled Backgrounds

D. P. Story

Table of Contents

1	Introduction	3
2	Alternate package name: aeb-tilebg	3
3	The Preamble and Package Options	3
4	The Commands	4
5	Reducing the file size	6
5.1	Tiling using the graphicxsp package	6
5.2	Tiling using graphicxsp and layers	7
5.3	Comparison of file sizes	7

1. Introduction

This package, `aeb_tilebg`, is a simple application of established packages `graphicx`, `multido` and `web`.

Download your favorite tiled background swatch from the Internet or create one yourself, convert it to an `.eps` or a `.pdf` format (depending if you use `distiller` or `pdftex`), place that image in the same folder as your source document. Anywhere in your document, use the command `\setTileBgGraphic` to bring in your tiled background using the `graphicx` package; for example,

```
\setTileBgGraphic[scale=.5]{\graphics_file}
```

The `multido` package places these images in the background, starting from the upper-left corner (the lower left corner if the `useLtiling` option is taken) going across and down (respectively, going across and up), one row at a time. The `web` package is used to manage the graphical template created.

During the development of this package, I Googled “backgrounds”, “tiled backgrounds” and “math tiled backgrounds” and found, amongst many, the following links

```
http://www.patswebgraphics.com/tiled.html
http://www.rhiossampler.com/Backgrounds/TBGIndex.htm
http://www.backgroundcity.com/
http://www.wou.edu/~burton1/mathart/mathart.html
```

I downloaded a few of the swatches and used them during the testing phase. The mathematical tiles of last the URL listed above are particularly interesting.

2. Alternate package name: `aeb-tilebg`

This package is listed on CTAN as `aeb-tilebg`, there is now a ‘dummy’ package by that name that loads `aeb_tilebg` with the specified options.

3. The Preamble and Package Options

In the preamble, place the `aeb_tilebg` package like so

```
\usepackage[options]{aeb_tilebg}
```

This must be placed after the introduction of the `web` package. A typical preamble might be

```
\documentclass{article}
\usepackage[driver,tight,designiv,usetemplates]{web}
\usepackage[options]{aeb_tilebg}
```

where `<driver>` can be `dvipsone`, `dvips` or `pdftex` (which include the `lualatex` application) and `xetex`. Only the `usetemplates` option and the driver type are required options here, the others are optional.¹

The options of the `aeb_tilebg` package are as follows:

¹The `web` package automatically detects `pdftex` and `xetex` so they need not be specified.

1. `uselleftiling`: When the package tiles a background, it begins in the upper left corner, goes across the row from left to right, then moves downward to the next row. This is the default behavior. The algorithm uses the `multido` command and floating point arithmetic; there may be some round off that leaves the tiles not quite contiguous, though I haven't observed this myself. If the upper-left starting point does not give satisfactory results, try using `uselleftiling`, this option causes `aeb_tilebg` to begin tiling in the lower-left corner; tiling moves left-to-right and upward. This uses a faster, more accurate algorithm of `multido` and should be superior and give the contiguous tiling you want.
2. `draft`: Tiling slows down \LaTeX compilation and obscures the text content as viewed in the previewer; therefore, use the `draft` to suppress tiling during the content development phase of the document. Remove the option to build the final document.
3. `ignoreforpaper`: The default behavior of `aeb_tilebg` is to automatically remove the tiling when the `forpaper` option of the `web` package is selected. To ignore this default "forpaper" behavior, select the `ignoreforpaper` option; then, even if the `forpaper` option is used, tiling of the document will occur.

4. The Commands

There are only a few commands of this package, which we list and discuss in this section.

To specify the graphics file that is to be tiled, use the `\setTileBgGraphic`, the central command of this package:

```
\setTileBgGraphic[⟨key_values⟩]{⟨filename⟩}
```

Parameter Description: The two parameters are passed to the `\includegraphics` command of the `graphicx` package, consequently, their descriptions are the same: the `⟨filename⟩` is the base name of the graphics file and `⟨key_values⟩` are any options of the `\includegraphics` command desired. The package automatically inserts the `hiresbb` option of `\includegraphics` to get a better fit of the tiles; specifying this option, therefore, is not needed.

Command Location: This command can be placed in the preamble to tile beginning at the first page, or anywhere in the body of the document to tile the current page, and all pages thereafter. The tile graphic can be changed freely, just by specifying a new graphics file in the `\setTileBgGraphic` command.

Below is an example of usage:

```
\setTileBgGraphic[scale=.5]{IndianBlanket}
```

The `scale` option re-scales the graphics by a factor given as its value. This option is often used to set the size of the tile, appropriate to the screen size. Other useful options are `width` and `height`.

You can disable and enable tiling using the following two commands:

```
\disableTiling \enableTiling
```

When you execute `\enableTiling`, the graphic tile currently defined will be used. You can change the tiling as well


You can disable tiling by using the `\disableTiling` command then...

```
\disableTiling
...
\enableTiling
\setTileBgGraphic[scale=.2]{tmfpurple_50}
```

For `multido`, the number of iterations must be specified in advance. The `aeb_tilebg` package uses `\multidostop` to break out of a `multido` loop early. By default it is set at 10, but can be changed through the `\maxiterations` command,²

```
\maxiterations{<number>}
```

here, the parameter `<number>` is the maximum number of iterations to be performed by the (nested) `multido`'s that tile the background. The default is `\maxiterations{10}`.

 The file `tilebg_tst.tex` is the demo file for the above series of commands.

The tiling algorithm automatically determines how many rows and columns you need to completely cover the background. This, of course, leads to partial tiles appearing in the background. In most cases this is not a problem; however, if you are particularly picky, you want only complete tiles to appear, no partials please. To accomplish this wonder feat of magic, use the `\autosetScreenSizeWithMargins` command.

```
\autosetScreenSizeWithMargins{<n_rows>}{<n_cols>}{<lm>}{<rm>}{<tm>}{<bm>}
```

Parameter Description: The command has six parameters, as listed below and described briefly.

1. `<n_rows>`: number of rows
2. `<n_cols>`: number of columns
3. `<lm>`: length of left margin
4. `<rm>`: length of right margin
5. `<tm>`: length of top margin
6. `<bm>`: length of bottom margin


²It may become necessary to increase this number if you decide to have a lot of tiles per page (increasing the size of your file, I might add) or if you use the `ignoreforpaper` command, where you are now trying to tile a larger area than typically is done for the screen.

Important: This command, which sets the screen dimensions of the document, can be used only in the preamble. When using this command, do not specify a design argument for the web package,³ and don't use either the `\margins` or `\screensize` commands of web as these two commands are used by `\autosetScreensizeWithMargins`.

An example of usage is

```
\setTileBgGraphic[scale=.25]{bike_10}
\autosetScreensizeWithMargins{7}{5}{.25in}{.25in}{24pt}{.25in}
```

The last four parameters are typical margin settings for the web package.

 `tilebg_tst_auto.tex` is the demo file for the `\autosetScreensizeWithMargins` command.

5. Reducing the file size

Tiling the background increases the file size of the document. I believe `pdftex` and `xtex` can reuse a graphics file to reduce the file size, and does a decent job.

5.1. Tiling using the `graphicxsp` package


One approach to reducing file size is to embed your tile graphic once and use and reuse it without significantly increasing file size. How is this done? Using the `graphicxsp` package.

We outline the method through example. In the preamble, we have,

```
\usepackage[tight,designiv,usetemplates]{web}
\usepackage{graphicxsp}
\usepackage{aeb_tilebg}

\embedEPS[transparencyGroup]{bike}{tile/bike_10}
\setTileBgGraphic[name=bike,
  transparency={/ca .3},scale=.25]{tile/bike_10}
%\autosetScreensizeWithMargins{7}{5}{.25in}{.25in}{24pt}{.25in}
```

You can optionally set the dimensions of the page to exactly 7 rows and 5 columns. With `graphicxsp`, you embed the graphic using `\embedEPS` and give it a name, `bike` in this example. In the optional argument of `\setTileBgGraphic` we pass that name using the `<name>` key and pass some transparency settings as well. See the documentation of `graphicxsp` for more information.

 `tilebg_test_sp.tex` is the test file for demonstrating this method. Acrobat Distiller is required as the PDF creator.

³This is referring to the web options of `designi-designvii`.

5.2. Tiling using `graphicxsp` and layers

One criticism of a background or tiled background is that it is really not desired when the document is printed. However, if you put the background in layers and set the layers never to print, the problem is solved. To use layers, the `uselayers` option of the `aeb_pro` package must be used (with Distiller as the PDF creator). To get the tiles to appear in their a layer (all in the same layer), use the `aeb_tilebg \placeTilesinLayers` command.


```
\placeTilesinLayers[⟨name⟩]{⟨KV-pairs⟩}
```

The `⟨name⟩` argument is the name of the layer, the default is `tileBG`. The `⟨KV-pairs⟩` argument consists of key-value pairs that are recognized by the `\xBld` command of `AeB Pro`, refer to the documentation of `\xBld` in the `AeB Pro` manual.

We outline the method through example. In the preamble, we have,

```
\usepackage[
  graphicxsp,uselayers,
  web={tight,designiv,usetemplates}
]{aeb_pro}
\usepackage{aeb_tilebg}
\embedEPS[transparencyGroup]{bike}{tile/bike_10}
\placeTilesinLayers{initState=on,print=never}
\setTileBgGraphic[name=bike,
  transparency={/ca .3},scale=.25]{tile/bike_10}
```

This is pretty much the same as the previous example, except `aeb_pro` is used with the `uselayers` option.


-  `tilebg_tst_layers.tex` is the test file for demonstrating this method. Acrobat Distiller is required as the PDF creator, the `Standard_transparency.joboptions`, which ships with `aeb_pro`, is required to distill this file so it has a transparent background.

5.3. Comparison of file sizes

We build and noted the file size of one of the test files under various conditions.

Driver	<code>graphicx</code>	<code>graphicxsp</code>	Reduced Size PDF
<code>dvips</code>	797KB	42KB	23KB
<code>pdftex</code>	62KB	-	-
<code>xetex</code>	27KB	-	-

Using the standard graphics package `graphicx`, `dvips` does the worst with `xetex` doing best. For `dvips`, with the test file uses `graphicxsp`, the file size is reduced to 43KB, and reduced further when the Reduced Size PDF operation is performed by Acrobat, accessed through the menu `File > Save As Other > Reduced Size PDF`.⁴

Now, I simply must get back to my retirement. 

⁴The user-interface is often redesigned, this is the location for Acrobat 11.