

AcroTeX.Net

# The aeb\_tilebg Package Creating Tiled Backgrounds

D. P. Story

## Table of Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>The Preamble and Package Options</b>	<b>3</b>
<b>3</b>	<b>The Commands</b>	<b>4</b>
<b>4</b>	<b>Reducing the file size</b>	<b>6</b>

## 1. Introduction

This newest 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 `uselleftiling` 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/~burtonl/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. 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,nodirectory]{web}
\usepackage[<options>]{aeb_tilebg}
```

where `<driver>` can be `dvipson`, `dvips` or `pdftex`.<sup>1</sup> Only the `usetemplates` option and the driver type are required options here, the others are optional.

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

<sup>1</sup>`dvipdfm` may be used in this package, but was not tested with the `aeb_tilebg` package.

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  $\text{\LaTeX}$  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.

### 3. 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:** These two parameters are passed to the `\includegraphics` command of the `graphicx` package, so 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,<sup>2</sup>

```
\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

**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 `Web`,<sup>3</sup> 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

<sup>2</sup>It 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.

<sup>3</sup>This is referring to the `Web` options of `designi-designvii`.

```
\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.

## 4. Reducing the file size

Tiling the background increases the file size of the document. I believe pdftex can reuse a graphics file to reduce the file size, and does a decent job. Acrobat too can reuse a graphic file. For users of Acrobat Pro, this section discusses a technique that can be used.

The key to this technique is to use the `addWatermarkFromFile()` JavaScript method for Acrobat. For convenience, it is suggested that the `aeb_pro` package be used to make this technique straightforward. The preamble might look like the following:

```
\documentclass{article}
\usepackage[
  driver=<dvips|dvipsone>,
  web={tight,usetemplates,nodirectory}
]{aeb_pro}
\usepackage{aeb_tilebg}
```

The steps are as follows:


1. Open the file `tilebg_blank.tex`. This is a simple file that produces one (or more) page(s). Set the design parameter of Web, as desired, or, remove the design option, set your own screen size, or let `\autosetScreenSizeWithMargins` set the size by using, as described above (using a graphic tile of your choice).
2. Compile the file, distill, and save the `.pdf` with a name of your choice.
3. Now, in your document, the one that uses `aeb_pro`, insert the `docassembly` environment in the preamble:

```
\begin{docassembly}
\addWatermarkFromFile({
  bOnTop:false,
  cDIPath:"/C/acrotex/aeb_tilebg/bike_bg.pdf"
});
\executeSave();
\end{docassembly}
```

4. If you are not using `\autosetScreenSizeWithMargins`, in the preamble, set up the file without any tiling by executing `\disableTiling`. If you are using the command `\autosetScreenSizeWithMargins`, the screen size must be set up to match the screen size of your background template; in this case, your preamble should read something like this

```
\setTileBgGraphic[scale=.25]{tile/bike_10}  
\autosetScreenSizeWithMargins{7}{5}{.25in}{.25in}{24pt}{.25in}  
\disableTiling
```

5. Compile and distill, the Acrobat JavaScript retrieves your background template embeds it in the PDF document, and uses and reuses the graphic on each of the specified pages.

 See the files `tilebg_blank.tex` and `tilebg_tst_pro.tex` for an example.

Now, I simply must get back to my retirement. 