

AcroTeX.Net

**Opacity-Pro**  
**Support for transparency**

**D. P. Story**

## Table of Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Requirements and Options</b>	<b>3</b>
2.1	LaTeX Package Requirements . . . . .	3
2.2	Package options . . . . .	3
2.3	PDF Creator Requirements . . . . .	3
2.4	Transparency Requirements . . . . .	3
<b>3</b>	<b>Setting transparency</b>	<b>4</b>

## 1. Introduction

The `opacity-pro` package defines one command and one environment that are used to set the *opacity* and *blend* of an object. The reader interested in transparency should read Chapter 7 of the *PDF Reference, Sixth Edition* for Version 1.7. The techniques used in this package were derived from the *pdfmark Reference*.

As the suffix “pro” might suggest, this package is for those who use distiller version 6.0 or greater to produce PDFs. The `opacity-pro` uses distiller with the Adobe PDF Settings set to process the transparency operator. For your convenience, the Adobe PDF Settings file `Standard_transparency.joboptions` is included in the distribution. Place this file wherever distiller looks for the `.joboptions` files.<sup>1</sup>

The package doesn’t really require other packages, but normally, it is used with the `color` or the `graphicx` packages. It does require that a `.dvi` to `.ps` converter be used that recognizes the special `\special{ps: ...}`. This includes, of course, `dvips`.

## 2. Requirements and Options

This package belongs to the high-class family of **AeB Pro**, hence, the major requirement of this package is that the PDF be created using **Acrobat Distiller**, see Section 2.3 for details.

### 2.1. $\LaTeX$ Package Requirements

A minimal example can be successfully compiled using the `opacity-pro` package alone.

### 2.2. Package options

`dvips` There are two options `dvips` and `dvipsone` corresponding to the two converter utilities  
`dvipsone` `dvips.exe` and `dvipsone.exe`. The `dvips` option is the default so it need not be specified.

### 2.3. PDF Creator Requirements

The big restriction on this package is the requirement to use Acrobat Distiller (version 5.0, or version 6.0 for transparency). The package was developed using Acrobat Distiller 8.1. The package supports the creation of Postscript using `dvips` and `dvipsone`. These “drivers” are defined through the required package `aeb_pro`.

### 2.4. Transparency Requirements

Distiller required To get the transparency effect, Acrobat Distiller version 6.0 or later is required. The default setting of the distiller does not support the **SetTransparency pdfmark**; it is necessary to edit the `.joboptions` file.

The procedure for editing `.joboptions` to support transparency is as follows:

#### 1. Start Acrobat Distiller

<sup>1</sup>Go to Settings > Edit Adobe PDF Settings ... in the Distiller application window, then click the SaveAs button. A Save Adobe PDF Settings As dialog box opens, and you can then see where Distiller likes to save its `.joboptions` file. Copy the provided `.joboptions` to the folder and restart Distiller, the `Standard_transparency` should now be visible in the drop down Default Settings list.

2. From the Default Settings list, select the setting you want to edit, usually, this will be the Standard job options.
3. Select `Settings > Edit Adobe PDF Settings (Ctrl+E)` from the distiller menu.
4. Click the `SaveAs` button at the bottom of the `Adobe PDF Settings` dialog box. Save your `.jobsettings` file under a new name, say `Standard_transparency` and make a note of where the distiller saves this file.
5. With your favorite text editor, navigate to the folder where you saved your new `.joboptions` file, and open it in your editor.
6. Look for the line that says `/AllowTransparency false`, and change this to read `/AllowTransparency true`. Save the changes and close the file.
7. Use this `.joboptions` file, `Standard_transparency` for example, whenever you distill with transparency pdfmarks. If your  $\LaTeX$  file uses transparency, and you are using a `.joboptions` file with `/AllowTransparency false`, distillation will fail and the distiller log should say

```
%%[Error: The PostScript contains Transparency pdfmark, job aborted.]%%
%%[ /AllowTransparency is false in job option settings.]%%
%%[ Error: undefined; OffendingCommand: pdfmark;
    ErrorInfo: Transparency Group ]%%
```

This suggests that you should use your `.joboptions` file that supports transparency!

**Note:** The `opacity-pro` package comes with a Distiller job options file named `Standard_transparency.joboptions`. Instead of going through the above instructions you can simply drop this file in a place where Distiller expects to find `.joboptions` files.<sup>2</sup>

### 3. Setting transparency

In this section we present the `\settransparency` command and the `settransparency` environment.

```
\settransparency*[\langle BM \rangle]{\langle ca \rangle}{\langle CA \rangle}[\langle PDFKVs \rangle]{\langle content \rangle}
```

The syntax for the environment is,

```
\begin{settransparency}[\langle BM \rangle]{\langle ca \rangle}{\langle CA \rangle}[\langle PDFKVs \rangle]
  \langle content \rangle
\end{settransparency}
```

<sup>2</sup>Go to `Settings > Edit Adobe PDF Settings ...` in the Distiller application window, then click the `SaveAs` button. A `Save Adobe PDF Settings As` dialog box opens, and you can then see where Distiller likes to save its `.joboptions` file. Copy the provided `.joboptions` to the folder and restart Distiller, the `Standard_transparency` should now be visible in the drop down Default Settings list.

There is a star-version as well.

```
\begin{settransparency*} [BM] {ca}{CA}[PDFKVs]
  content
\end{settransparency*}
```

The parameters are,

- \* (optional) If present, the PDF entries **ca** and **CA** are set as `/ca <ca>` and `/CA <CA>`; otherwise, the algorithm for setting the **ca** and **CA** entries into the PDF is used. The algorithm is described in the paragraph **Limitations of dvips** below.
- BM* Current blend mode. Names recognized are Normal, Multiply, Screen, Overlay, Darken, Lighten, ColorDodge, ColorBurn, HardLight, SoftLight, Difference, Exclusion, Hue, Color, Saturation and Luminosity. See the accompanying file `blendmodes.tex` for a description of each.
- ca* Current alpha constant, specifying the constant shape or constant opacity value to be used for *non-stroking operations*, which includes filling paths. A number between 0 and 1, inclusive. Default is 1.0.
- CA* Current stroking alpha constant, specifying the constant shape or constant opacity value to be used for *stroking operations*. A number between 0 and 1, inclusive. Default is 1.0.
- PDFKVs* (optional) These pairs can be inserted using the optional fourth parameter (see page 38 of the *pdfmark Reference, Version 8.0*, for a listing of these other key-value pairs).

In the environment case, the last parameter is optional, so  $\TeX$  will be looking for a left brace ‘[’, if the *content* of the environment begins with a command, that command will get expanded while  $\TeX$  looks for a left brace; if this is a problem, simply put an empty optional argument ‘[]’ to make  $\TeX$  happy.

The target object, *contents*,<sup>3</sup> is either the fifth argument, or the content of the environment.

Refer to the examples provided by this distribution to see the command and environment in action.

**Limitations of dvips.** A background, such as one created by `\colorbox` or `\fcolorbox` may be stroked (*ca*) as `dvipsone` does, or filled (*CA*) as `dvips` does. Text is usually filled (*ca*). Because `dvips` fills a background and text, it creates a problem when trying to have text overlay a colorful background, where the text and background have different opacity values (both set by *ca*).

**An algorithm.** For `dvips`, we implement the following algorithm: The value of *ca* is taken from the value of *CA*, if the value of *CA* is specified. In this way, for `dvips` we always have *ca* = *CA*. We do it this way because of the `digicap-pro` package; the user sets the `textop` key (corresponding to *CA*) and the `bgop` key (*ca*).

<sup>3</sup>These can be text, pictures, color boxes, and so on.

**Bypassing the algorithm with `*`.** When the star-option is taken, the algorithm is bypassed; as a result we set `/ca`  $\langle ca \rangle$  and `/CA`  $\langle CA \rangle$ . This is useful in instances where paths are stroked. See the sample file `op-pst.tex` for an example of the use of the star-option.

**Recommendations.** The setting of  $\langle CA \rangle$  should be small enough to get discernable transparency, but not so much that the text is difficult to read. Recommendation: set  $\langle ca \rangle$  to `.7` or thereabouts.

### Contrasting background and text using different opacities

**dvips** The first item below illustrates that `dvips` does not obey the second argument. The second item show a workaround.<sup>4</sup>

- **Some text**

```
\settransparency{.7}{.2}{\colorbox{yellow}{\textbf{Some text}}}
```

In the above yellow box, both the background yellow and the foreground text have opacity `.7` because `dvips` fills both background and text.

- **Some text**

```
\settransparency{.2}{.2}{\colorbox{yellow}
  {\settransparency{.7}{.7}{\textbf{Some text}}}}
```

That is, we enclose the yellow background in a transparency value of `.2` (the first argument is read only) and within the `\colorbox` argument, we use a transparency value of `.7` (first argument is read only).

For `dvips` users, the second argument can be empty, because, again, only the first argument has any effect on filled objects.

**dvipsone** **Some text**

```
\settransparency{.7}{.2}{\colorbox{yellow}{\textbf{Some text}}}
```

For the `dvipsone` driver,<sup>5</sup> the second parameter has an effect.

That's all for now, I simply must get back to my retirement. ~~DS~~

<sup>4</sup>The effects illustrated for this case are simulated as the author uses the `dvipsone` driver.

<sup>5</sup>very few users of `dvipsone` remain on earth, I am one.