

**AcroTeX.Net**

**Importing and Placing Images**

**using AeB Pro**

**D. P. Story**

This file demos the use of four new commands:

```
\declareImageAndPlacement, \declareMultiImages
\insertPreDocAssembly, \placeImage
```

These are described in detail in `aebpro_man.pdf` (the AeB Pro manual). Here, we content ourselves by example, and adding in any comments the spirit moves me to make.

In this document, we import four image files `man1.pdf`, `girl.png`, `AcroFord.jpg`, and `scot.gif`, just to illustrate that you can import a variety of graphic formats. All files are converted to PDF when they are imported anyway, so actually the most efficient thing to do is to convert the files to PDF before hand, which we shall not do here.

In the preamble, we have

```
\declareImageAndPlacement{name=manAvatar,           (1)
    path=graphics/man1.pdf,placement={Avatar1,Avatar2}}
\declareMultiImages                                 (2)
{%
    {path=girl.png,placement={Avatar3,Avatar4}}      (3)
    {path=AcroFord.jpg,placement=AcroFord}           (4)
    {path=scot.gif,placement=Scot}                   (5)
}
\begin{docassembly}
\insertPreDocAssembly                               (6)
\end{docassembly}
```

The `\declareImageAndPlacement` and `\declareMultiImages` (lines (1)–(5)) must occur outside the `docassembly` environment. `\declareImageAndPlacement` is used to import one image, and specify its placement; `\declareMultiImages` does the same thing, but with multiple images.

In line (1), we import an image using the `\declareImageAndPlacement` command.

- The `name` key is used to specify a symbolic name to the image that is used later (not seen by the user), the `name` parameter is only specified in line (1). This parameter is optional, if not specified, one is automatically generated.
- The `path` key is the path to the graphic file. This path can be relative or absolute. If absolute, it must be expressed in Acrobat's device independent path syntax, for example, `/c/acrotex/myimages/myimage.png`. You can see in these examples the graphic formats are PDF, PNG, JPG, and GIF. Other graphic formats are supported as well.
- The `final` key is the `placement` key, its value is a comma-delimited list of filenames given to the push buttons that receive these images as an appearance. The push buttons are created using the `\placeImage` command.
- The `page` key, not shown, is used with PDF graphics files. A PDF file may contain many pages with a image on each page. Use the `page` key to specify the page that contains the image you want to use. The default is `page=0`.

In line (2), we have the `\declareMultiImages` command, this is nothing but a loop that calls `\declareImageAndPlacement` for each of its arguments. The arguments of the command `\declareMultiImages` are the as those of `\declareImageAndPlacement` enclosed in matching

braces, as shown in lines (3)–(5).

The first image, defined in (1), could have been included in the `\declareMultiImages` command.

In line (6), we have the `\insertPreDocAssembly` command inside the `docassembly` environment. This command expands to a series of JavaScript lines created by the commands `\declareImageAndPlacement` and `\declareMultiImages`.

Now, within the body of document, we insert `\placeImage` commands. The following two are PDF images:

```
\placeImage{Avatar1}{.5in}{.5in}\quad\placeImage{Avatar2}{.25in}{.25in}
```

The background is transparent for this avatar, so the colored background of the page is seen around the figure.

```
\placeImage{Avatar3}{.5in}{.5in}\quad\placeImage{Avatar4}{.25in}{.25in}
```

These two are similar, the same image is placed in buttons with have different names. The background of this PNG graphic is transparent.

```
\resizebox{2in}{!}{\placeImage{AcroFord}{669bp}{274bp}}\quad
\placeImage{AcroFord}{1.5in}{274bp*\ratio{1.5in}{669bp}}
```

This is a JPG image of the old English Ford our family owned back in 1953 at Schulthorpe RAF Station, England. Even then, you can see that acrotex was destined for fame. Notice the field name of these two images are the same, so we get the same image appearing.

A comment on rescaling. I've rescale my old English Ford two ways: (1) using the `\rescalebox` command; and (2) using the `calc` package, here, we want 1.5in width, and the height is scale accordingly, with scale factor of  $\ratio{1.5in}{669bp}$ . The original image was  $669bp \times 274bp$ .

The final example is a GIF file, it has a transparent background as well. The code is

```
\resizebox{!}{2in}{\placeImage{Scot}{200bp}{268bp}}
```

The image is

This is how I feel when I am pulled out of retirement! That's all for now. This is Jet Jackson signing off, "Justice, through Strength and Courage! Out!"

**PS:** Don't forget to save the document after the images are imported. oo