

AcroTeX.Net

**GraphicxSP
Forms**

D. P. Story

In the preamble, we make the following definitions: We embed for EPS files, two of which are set a transparency groups:

```
\embedEPS[hiresbb,transparencyGroup]{AdobeDon}{AdobeDon}
\embedEPS[hiresbb]{Airplane}{000_0151}
\embedEPS{3Dcheck}{3Dcheck}
\embedEPS{3Dx}{3Dx}
```

We also use the `createImage` environment to create more images, some new, others made from the embedded files.

Create variations on `AdobeDon` by giving the image various degrees of opacity:

```
\begin{createImage}[0 0 \widthOf{AdobeDon}\space\heightOf{AdobeDon}]{dAdobeDon}
  [ /ca .3 /SetTransparency pdfmark
  [ {AdobeDon} /SP pdfmark
\end{createImage}
\begin{createImage}[0 0 \widthOf{AdobeDon}\space\heightOf{AdobeDon}]{rAdobeDon}
  [ /ca .5 /SetTransparency pdfmark
  [ {AdobeDon} /SP pdfmark
\end{createImage}
```

Take the `Airplane` image a reflect it with respect to the x and y axes.

```
\begin{createImage}[0 0 \widthOf{Airplane}\space-\heightOf{Airplane}]{dAirplane}
  1 -1 scale [ {Airplane} /SP pdfmark
\end{createImage}
\begin{createImage}[0 0 -\widthOf{Airplane}\space\heightOf{Airplane}]{rAirplane}
  -1 1 scale [ {Airplane} /SP pdfmark
\end{createImage}
```

Now create two new graphic images using raw PostScript. In all cases we use the `createImage` environment, first optional argument is the bounding box, the second is the symbolic name for the graphic. The symbolic name can be used in the `\includegraphics` command or the `\insertEPS` command.

```
\begin{createImage}[0 0 100 100]{x0}
.7529 setgray 0 0 100 100 rectfill 1 setgray 2 2 moveto 2 98 lineto 98 98 lineto
96 96 lineto 4 96 lineto 4 4 lineto fill 0.34 setgray 98 98 moveto 98 2 lineto
2 2 lineto 4 4 lineto 96 4 lineto 96 96 lineto fill
0 setgray 22.5 22.5 moveto 1 0 0 setrgbcolor /Helvetica 72 selectfont (0) show
\end{createImage}
\begin{createImage}[0 0 100 100]{xX}
.7529 setgray 0 0 100 100 rectfill 1 setgray 2 2 moveto 2 98 lineto 98 98 lineto
96 96 lineto 4 96 lineto 4 4 lineto fill 0.34 setgray 98 98 moveto 98 2 lineto
2 2 lineto 4 4 lineto 96 4 lineto 96 96 lineto fill
0 setgray 27 22.5 moveto 0 0 1 setrgbcolor /Helvetica 72 selectfont (X) show
\end{createImage}
\begin{createImage}[0 0 \widthOf{Airplane}\space-\heightOf{Airplane}]{dAirplane}
1 -1 scale [ {Airplane} /SP pdfmark
\end{createImage}
\begin{createImage}[0 0 -\widthOf{Airplane}\space\heightOf{Airplane}]{rAirplane}
-1 1 scale [ {Airplane} /SP pdfmark
\end{createImage}
```

Subsequent pages demonstrate how to use these images in the context of form fields.

This page demonstrates that graphics embedded by the BP and EP pdfmarks are part of the content of the page and can therefore be placed in different layers.



Click here:

Note: Need `app.runtimeHighlight=false` for the checkboxes to work correctly.

The appearance attributes of the check box above, yes it is a checkbox, is as follows:

```
\AP{/N << /Yes {Airplane} /Off {dAirplane} >>  
  /D << /Yes {Airplane} /Off {dAirplane} >>  
  /R << /Yes {dAirplane} /Off {rAirplane} >>
```

Here are more traditional uses for a checkbox, but this time using custom checks and crosses.

The verbatim listing of the second check box is, for example,

```
\resizebox{11bp}{!}{\checkbox[%
  \A{\JS{%
    var f = this.getField("reportCk3D");\r
    f.value = !!event.target.isBoxChecked(0);\r
    this.dirty=false;
  }}
  \V{Off}\DV{Off}\AS{Off}\H{N}\autoCenter{n}
  \AP{/N << /Yes {3Dcheck} /Off {3Dx} >> }
]{ckBox3D}{\widthOf{3Dcheck}bp}{\heightOf{3Dcheck}bp}{Yes}}
\textField[\Q1\Ff\FfReadOnly\autoCenter{n}]{reportCk3D}{.5in}{11bp}
```

We use `\resizebox` from the `graphicx` package to adjust the sizes of the appearances.

One last example, this one uses transparency. We now we create a push button with the face of AdobeDon as the normal appearance (somewhat paradoxically), AdobeDon at 50% opacity for the rollover appearance, and AdobeDon at 30% for the down appearance. Will it work, we'll see:

Push button: