ROBERT THORSBY

35 Main Street, Somewhere in NSW 1234

robert@example.com Phone: (0123) 456 789

Groff Output PDF File With Small EPS Image

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Quis autem vel eum iure reprehenderit, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

This “proof of concept” was created without any macro set (*i.e.*, other than *PSPIC* which is automatically included if the output device is “-Tps”). The line length “.ll” is 450p; and the margins together with the rules are included to indicate the boundaries of the printable area. The body text is set in Times Roman 12/14. The “Dummy EPS File” was created from the command line with *ImageMagick*, which I shall leave as an exercise for the reader. Incidentally, the native dimensions of the EPS image are 80x92. Why the height should be 92 and not 98 (*i.e.*, 7×14) I have no idea. Perhaps the gurus might enlighten us.

Between the first two paragraphs, “Lorem ipsum ... id est laborum.” and “Sed ut perspiciatis ... nulla pariatur?”, I inserted the following requests:

```
.mk Img1
.sp -0.5v
.ds Image1 /a/very/long/path/to/my/dummy_eps_file.eps
.PSPIC -L \*[Image1] 80p
.sp |\n[Img1]u+1v
.in +84p
```

I added a “.br” request followed by “.in” and some placeholder text, and processed the input file with the following single command line instruction (broken here for reasons of sanity):

```
groff -Tps -dpaper=a4 -P-pa4 "inputwithimage.gr"
| ps2pdf -sPAPERSIZE=a4 -sColorModel=LeaveColorUnchanged
   -dPDFSETTINGS=/prepress -dEmbedAllFonts=true - "inputwithimage.pdf"
```

On the first run the second paragraph remained indented well past the bottom of the image, as was expected. It showed that the breakpoint was the word “tempora” so there I took a newline and added:

```
.br
.in
```

and removed the later “.br” and “.in” requests and the additional lipsum placeholder text.

The second run had the desired effect, except that “tempora” now ended a couple of characters in from the right margin. This was fixed by adding “\p” (*i.e.*, “tempora\p”) and deleting the following line (“.br”)— the break is incorporated into the “\p” escape.

The third (final) run produced the result shown above.

Please note, the image is left-aligned and 80 points wide whereas the indent, two lines below, is 84 points. Those extra four points create a margin between the image and the text alongside it.