

Tip of the Day—EPS Files

David Arnold

March 31, 1999

Tip 10 — What’s a Bounding Box?

Yesterday we placed an encapsulated postscript image in our document and centered it with the commands `\begin{center} ... \end{center}`. It is clear that this environment *centers* whatever is included between `\begin{center}` and `\end{center}`, but you may not realize that the `center` environment also adds a little white space both before and after the centered environment.



However, when you compile and preview this file, you might note that there seems to be more white space *after* the encapsulated postscript image than the amount of white space appearing before the image. One would hope that the `center` environment would put *equal* amounts of white space both before and after the centered environment. Is this anomaly due to \LaTeX 's page layout mechanism (which could be the case) or is our graphic at fault?

GSView Analysis of the Image

When you unzipped `tip10.zip`, you accrued four files: `tip10.tex`, `town.eps`, `town2.eps`, and `town3.eps`. Open GSView. In GSView, select File→Open and open the file `town.eps`. This will open the image of the town in GSView. In GSView, select Options→Show bounding box. A check mark will indicate that this menu option is selected and you will note a dashed line surrounding your graphic in GSView. This dashed box is the so-called “bounding box” of your encapsulated postscript image. Selecting Options→Show bounding box again will hide the bounding box, but we want the bounding box to be visible for the remainder of this activity.

Postscript is a page layout language, working a page at a time. The origin of the page, denoted $(0, 0)$, is located at the extreme lower left-hand corner of the page. All measurements are made from this origin and the unit of measurement is `bp`, postscript's "big points." There are 72 bps per inch. In GSVIEW, as you move the mouse cursor over the page, the position of the mouse cursor, in bps,¹ is indicated on the status bar, just below the menu bar.

Move your cursor to the lower left-hand corner of the dashed bounding box and note that its coordinates (in bps) are approximately $(197, 292)$. Similarly, move the cursor to the upper right-hand corner of the dashed bounding box and note that its coordinates (again in bps) are approximately $(392, 488)$.

Return to WinEdt. In WinEdt, select File→Open and open the file `town.eps`. Note that the first few lines of the file are as follows.

```
%!PS-Adobe-2.0 EPSF-1.2
%%Creator: MATLAB, The Mathworks, Inc.
%%Title: F:\StaffDevelopment\tips\goldhill.eps
%%CreationDate: 03/30/99 22:00:34
%%DocumentNeededFonts: Helvetica
%%DocumentProcessColors: Cyan Magenta Yellow Black
%%Pages: 1
%%BoundingBox: 197 292 392 488
%%EndComments
```

Take special note of the line:

```
%%BoundingBox: 197 292 392 488
```

The nature of the bounding box should now be clear.

Manually Changing the Bounding Box

Return to GSVIEW and note that the bounding box does not snugly surround the graphic image. Rather, there is white space between the bounding box and the image, and, to our dismay, it is not distributed evenly about the image. There is much more white space on the bottom than there is on the top and this may explain why there is more white space below the image shown on page 1 of this document.

Let's adjust the bounding box so that it fits snugly to the image of the town, leaving **no** white space between the image and the bounding box. In GSVIEW, move your mouse so that it is positioned over the lower left-hand corner of the image of the town and note that its coordinates are approximately (in bps) $(221, 311)$. Similarly, note that the coordinates of the upper right-hand corner of the image of the town are approximately $(388, 478)$.

¹Actually GSVIEW reports these measurements as points, as in `pt`, but this is understood to be postscript points, or `bps`. L^AT_EX's points are scaled so that 72.27 L^AT_EX pts equals one inch.

Return to WinEdt. Select the `town.eps` tab to make the `town.eps` file current, if it isn't already. Select File→Save As and save the file as `town2.eps` in the same folder as `tip10.tex` and `town.eps`. Next, change the line

```
%%BoundingBox: 197 292 392 488
```

to

```
%%BoundingBox: 221 311 388 478
```

and save the file. When the file `town2.eps` is placed in the file between `\begin{center}` and `\end{center}`, you get the following result.



Note that eliminating the white space by adjusting the bounding box has now evened up the white space that appears both before and after the image in this output file.

Here is the code used to include `town2.eps`.²

```
\begin{center}  
\includegraphics[clip]{town2}  
\end{center}
```

Note that the `clip` option is used with the `\includegraphics` command. This is required if you want to clip the image to the new bounding box. Moreover, YAP may not understand this clipping, so you have to change the DVI file to postscript with DVIPS, then view the resulting postscript file in GSView to appreciate the improvement.

²Perhaps the \LaTeX gurus can explain to me why the `verbatim` environment seems to create more white space after than before the environment in this case (darnold@northcoast.com).

Experiment

I would like to encourage you to play with the bounding box parameters. Don't worry, you won't hurt anything with this free play. For example, change the bounding box in `town2.eps` to

```
%%BoundingBox: 296 327 374 426
```

and save the file as `town3.eps`. The code

```
\begin{center}  
\includegraphics[clip]{town3}  
\end{center}
```

produces the following image.



Again, use DVIPS to create the postscript file, then view the result in GSView.
I hope you now better understand the concept of a bounding box.