

# L<sup>A</sup>T<sub>E</sub>X Document Formatting Examples

Gene Michael Stover

created Thursday, 2005 May 19  
updated Sunday, 2005 November 20

*Copyright © 2005 Gene Michael Stover. All rights reserved. Permission to copy, store, & view this document unmodified & in its entirety is granted.*

## Contents

<b>1</b>	<b>What is this?</b>	<b>1</b>
<b>2</b>	<b>Summary</b>	<b>2</b>
<b>3</b>	<b>Pictures in Figures</b>	<b>2</b>
<b>4</b>	<b>Images outside of figures</b>	<b>6</b>
<b>5</b>	<b>The <code>includegraphics</code> command</b>	<b>7</b>
<b>6</b>	<b>Text Flowing Around Pictures</b>	<b>7</b>
6.1	Attempt 1, Failure . . . . .	7
6.2	Attempt 2, Partial Success . . . . .	8
<b>7</b>	<b>Figures with borders</b>	<b>8</b>
7.1	<code>usepackage float / boxed</code> technique . . . . .	9
<b>A</b>	<b>Other File Formats</b>	<b>11</b>

## 1 What is this?

Most of what I do here is discussed in one or more of these sources:

- *L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System*. By Leslie Lamport. [4]
- *The L<sup>A</sup>T<sub>E</sub>X Web Companion: Integrating T<sub>E</sub>X, HTML, and XML*. By Michel Goossens and Sebastian Rahtz. [3]
- *The L<sup>A</sup>T<sub>E</sub>X Companion*. By Michel Goossens and Frank Mittelbach. [2], &

- “Graphics & Colour with L<sup>A</sup>T<sub>E</sub>X”. By Patrick W. Daly. [1].

All I’m doing in this essay is trying out what they say, figuring it out now so I can use it when I want it later.

## 2 Summary

I found two ways to include images. They are:

- Inside a figure, using the `\epsfig` command. It works except that the HTML version loses the colour of the original encapsulated Post Script file.
- Outside of a figure. Use conditional compilation to place an encapsulated Post Script image in the printed output & a hyperlink in the HTML output. Works well except that I don’t have much control over how the image is displayed.

What I’d really like to do is make text in a paragraph flow around a small image in both the printed output & the HTML output. I don’t know how to do that yet.

## 3 Pictures in Figures

Let’s start with a colour JPEG image & put it in a figure so that it displays in both the HTML & the dead tree versions of the document.

Figure 1 shows the figure. The idea is that it looks pretty much the same in the HTML & printed versions of this essay.

Here’s how I did that:

1. I began with a colour JPEG picture<sup>1</sup> that is 640 by 480.
2. Loaded the image into The Gimp<sup>2</sup>.
3. Saved the image as a Post Script file. Options I used on the *Safe As...* window were:
  - (a) Post Script level 2
  - (b) Encapsulated Post Script
  - (c) Preserve aspect ratio
  - (d) Units of millimeters
  - (e) X & Y offsets of 1.00 & 1.00

---

<sup>1</sup><http://cybertiggyr.com/gene/pho/2004-12/04-640.jpg>

<sup>2</sup><http://www.gimp.org/>



Figure 1: My cat, Wordsworth. 2004 December.

- (f) Width & Height of 120 & 90. Those dimensions have the same ratio as 640:480. I experimented to find dimensions which were small enough for the printed page.
  - (g) New filename of 04-640.ps
4. Near the top of the L<sup>A</sup>T<sub>E</sub>X source code for this document, I inserted `\usepackage{epsfig}`.
  5. The L<sup>A</sup>T<sub>E</sub>X code for the figure is:

```

\begin{center}
\begin{figure}
\epsfig{file=04-640.ps}
\caption{My cat, Wordsworth. 2004 December.}
\label{fig-colour-jpeg}
\end{figure}
\end{center}

```

This technique seems to work for both the PDF output & the HTML output except that in the HTML version, L<sup>A</sup>T<sub>E</sub>X2HTML converted the encapsulated Post Script image to a Portable Network Graphics (PNG) file & lost the colour in the process.

Some research shows that the loss of colour happens when `pstoimg` converts the encapsulated Post Script image to a PNG. I experimented with ways to run `pstoimg` from the command line, & everything I tried lost the colors, so fixing things is not just a matter of correcting how L<sup>A</sup>T<sub>E</sub>X2html invokes `pstoimg`. Ideally, it would be a case of improving `pstoimg`.

Not bad for a first try. I'm surprised & not exactly pleased that, in the printed version, the figure is on the next page, by itself. If a future version of `pstoimg` preserves colours, this technique will be perfect for figures containing pictures.

I tried using conditional compilation to use the Post Script image in the PDF output & a hyperlink to the JPEG in the HTML output, but no luck. The PDF output is fine, but the HTML output is a great big gray, featureless box. Figure 2 demonstrates.

Here is the source code for Figure 2:

```

\begin{figure}
\begin{latexonly}
\epsfig{file=04-640.ps}
\end{latexonly}
\begin{htmlonly}
\htmladding{../pho/2004-12/04-640.jpg}
\end{htmlonly}
\caption{Using conditional compilation for images in a
figure. Should look fine in {\sc pdf}, but doesn't work
in the {\sc html}.}

```



Figure 2: Using conditional compilation for images in a figure. Should look fine in PDF, but doesn't work in the HTML.

```
\label{fig-conditional-00}  
\end{figure}
```

So that didn't work.

## 4 Images outside of figures

If I don't need the image to be in a figure, conditional compilation works.

Here is an image, outside of a figure, which uses `\epsfig` to load an encapsulated Post Script file for the PDF output & an hyperlink to the original image for the HTML version. It switches between the two commands with conditional compilation.



Here's the  $\LaTeX$  source code which displayed that image:

```
\begin{latexonly}  
  
    \begin{center}  
        \epsfig{file=04-640.ps}  
    \end{center}  
  
\end{latexonly}  
\begin{htmlonly}
```

```
\htmladdimg[ALIGN="CENTER"]{../pho/2004-12/04-640.jpg}
\\
```

```
\end{htmlonly}
```

That's pretty much what I want, though it only works outside of a figure.

## 5 The `\includegraphics` command

I used `\epsfig` for the image. For the HTML version, that leads to the problem of lost colours because `LATEX2html` must convert it to an image file. The `\includegraphics` command should allow us to use the raw image file for both the PDF & the `html`.

Here are some commands I tried from the `graphicx` package, but they all produce errors at compile-time. I'm not sure, but I think the `dvips` program may be the one which spews the error.

Here are some things I tried from `graphicx`:

- `\includegraphics[width=64mm,height=48]{../pho/2004-12/04-640.jpg}`
- `\includegraphics[bb=0 0 64mm 48mm]{../pho/2004-12/04-640.jpg}`

## 6 Text Flowing Around Pictures

### 6.1 Attempt 1, Failure

We'll start with a JPEG image. The one I used is at <http://cybertiggyr.com/gene/dfx/thumb.jpeg>. I got it from Google, & maybe I should make my own to avoid copyright problems.

I saved it as an Encapsulated Post Script file, Post Script level 2, measuring 30 millimeters high by 40 millimeters wide, but I told the Gimp to preserve the aspect ratio, so the actual image might have a slightly different size.

*Here's what I would try if I had the `picins` package on all my `LATEX` computers.*

```
\parpic [sr] [r] {\includegraphics{thumb.ps}} I used the \parpic
command of the picins package to place the image to the right of
this paragraph.
```

I sure don't like the results. In the PDF output file, the image is taller than the paragraph that contains it, & `LATEX` does not compensate by placing vertical space between that paragraph & the one which follows. What's more, the image is not included at all in the HTML output.

1. Given: Numbers  $a$  &  $b$ .
2. Let  $a^2 \Rightarrow a \times a$ .
3. Let  $b^2 \Rightarrow b \times b$ .
4. Let  $c^2 \Rightarrow a^2 + b^2$ .
5. Let  $c \Rightarrow \sqrt{c^2}$ .
6. Return  $c$ .

Figure 3: Pseudocode without a border

- |  |
|--|
| <ol style="list-style-type: none"> <li>1. Given: Numbers <math>a</math> &amp; <math>b</math></li> <li>2. Let <math>a^2 \Rightarrow a \times a</math></li> <li>3. Let <math>b^2 \Rightarrow b \times b</math>.</li> <li>4. Let <math>c^2 \Rightarrow a^2 + b^2</math>.</li> <li>5. Let <math>c \Rightarrow \sqrt{c^2}</math>.</li> <li>6. Return <math>c</math>.</li> </ol> |
|--|

Figure 4: Pseudocode with a border

## 6.2 Attempt 2, Partial Success

If I want text flow around the image in the HTML output, but I'll do without the image in the hard-copy version, I can use the `\htmladdimg` command to achieve it in the HTML version of the document. The printed version will not show the image at all, so the text will not flow around it.

In the HTML version, this paragraph flows around an image on the right. In the printed version, there is no image.

(New paragraph) It works fine.

## 7 Figures with borders

Here's how to put a border around a figure so the reader can more easily see that it's a figure & not part of the text. Such a border is particularly useful for figures whose content is mostly text (such as pseudocode).

*This technique doesn't work well, so keep reading until you come to the "usepackage float / boxed" technique.*

Figure ?? is a figure containing pseudocode & without a border.

Table ?? is an attempt at a bordered figure containing text by using a *tabular* environment inside the figure.

Here is the source code for the figure with the border.

```
\begin{figure}
```

```

\begin{center}
  \begin{tabular}{|r|} \hline
    1. & Given: Numbers {\em a} \& {\em b} \\
    2. & Let  $a^2$  \(\rightarrow a \times a \\
    3. & Let  $b^2$  \(\rightarrow b \times b. \\
    4. & Let  $c^2$  \(\rightarrow a^2 + b^2. \\
    5. & Let  $c$  \(\rightarrow \sqrt{c^2}. \\
    6. & Return {\em c}. \\
  \hline
  \end{tabular}
\end{center}
\caption{Pseudocode with a border}
\label{fig-pseudocod-borderless}
\end{figure}

```

Three bummers include:

1. Since it's a table, I must number each row myself. (A good L<sup>A</sup>T<sub>E</sub>X programmer could fix that, I'm sure.)
2. Again since it's a table, multi-line entries could be difficult. I might have to insert line breaks.
3. Again since it's a table, nested steps will be difficult.

## 7.1 usepackage float / boxed technique

Here's the best way I know so far to make a figure with a boarder.<sup>3</sup>

Figure 5 shows a figure with a border. Well, that's what it shows in the L<sup>A</sup>T<sub>E</sub>X output, such as a DVI, PostScript, or PDF file. As of 2005 November 20, L<sup>A</sup>T<sub>E</sub>X2HTML does not convert it correctly, so you see a regular figure with "boxed figure" before it & "plain figure" after it.

Let's get rid of the "boxed figure" & "plain figure" garbage in the HTML output. Figure 6 shows the result.

So we get a figure with a border in L<sup>A</sup>T<sub>E</sub>X & a regular figure (often with a shaded background) in L<sup>A</sup>T<sub>E</sub>X2HTML. It's not exactly what I wanted, but it's good enough.

Here's the source code for the previous figure:

```

\ begin{latexonly}
\floatstyle{boxed} \restylefloat{figure}
\ end{latexonly}
\begin{figure}
\begin{center}
  \begin{enumerate}
    \item Given: Numbers {\em a} \& {\em b}

```

---

<sup>3</sup>This supremely "exactly what I was looking for" technique was suggested by Mair. Many thanks, Mair.

1. Given: Numbers  $a$  &  $b$
2. Let  $a^2 \Rightarrow a \times a$
3. Let  $b^2 \Rightarrow b \times b$ .
4. Let  $c^2 \Rightarrow a^2 + b^2$ .
5. Let  $c \Rightarrow \sqrt{c^2}$ .
6. Return  $c$ .

**Figure 5:** Pseudocode with a border in  $\text{\LaTeX}$ , with preceeding & following crap in HTML

1. Given: Numbers  $a$  &  $b$
2. Let  $a^2 \Rightarrow a \times a$
3. Let  $b^2 \Rightarrow b \times b$ .
4. Let  $c^2 \Rightarrow a^2 + b^2$ .
5. Let  $c \Rightarrow \sqrt{c^2}$ .
6. Return  $c$ .

**Figure 6:** Pseudocode with a border in  $\text{\LaTeX}$  output, no border in HTML output

```

\item Let  $a^2 \rightarrow a \times a$ 
\item Let  $b^2 \rightarrow b \times b$ .
\item Let  $c^2 \rightarrow a^2 + b^2$ $.
\item Let  $c \rightarrow \sqrt{c^2}$ $.
\item Return {\em c}.
\end{enumerate}
\end{center}
\caption{Pseudocode with a border in \LaTeX\ output, no
border in {\sc html} output}
\label{fig-pseudocod-boxed2}
\end{figure}
\begin{latexonly}
\floatstyle{plain} \restylefloat{figure}
\end{latexonly}

```

Here are all the instructions about using this technique:

1. At the top of the document, put `\usepackage{float}`. Because it's only useful to  $\LaTeX$ , I guess you could wrap it in a `latexonly` environment.
2. If you want all your figures to have borders, put a `latexonly` wrapper around “`\floatstyle{boxed} \restylefloat{figure}`” after the `\usepackage float`. In fact, you could put all three commands on the same line & wrap them in `latexonly`.
3. If you just want certain figures to have borders, then in front of such a figure, put a `latexonly` wrapper around “`\floatstyle{boxed} \restylefloat{figure}`” in front of the figure & “`\floatstyle{plain} \restylefloat{figure}`” after the figure.

## A Other File Formats

- This document is available in multi-file HTML format at <http://cybertiggyr.com/gene/dfx/>.
- This document is available in Pointless Document Format (PDF) at <http://cybertiggyr.com/gene/dfx/dfx.pdf>.

## References

- [1] Patrick W. Daly. Graphics and colour with . *Max-Planck-Institut Für Aeronomie*, June 1998. <http://tex.loria.fr/graph-pack/grf/grf.htm>.
- [2] Michel Goossens and Frank Mittelbach. *The  $\LaTeX$  Companion*. Addison Wesley Longman, Inc., 1993. ISBN 0201541998.
- [3] Michel Goossens and Sebastian Rahtz. *The  $\LaTeX$  Web Companion: Integrating  $\TeX$ , HTML, and XML*. Addison Wesley Longman, Inc., 1999. ISBN 020143317.

- [4] Leslie Lamport. *LaTeX: A Document Preparation System*. Addison-Wesley Publishing Company, Inc., 1986. ISBN 0-201-15790-X.