Travels in T_EX Land: Tweaking LAT_EX to do something a little different

David Walden

1 Introduction

Sooner or later, many users new to LATEX begin to want to tweak LATEX to work differently than it does "out of the box." These tweaks may be

- to provide capabilities that do not already exist in LATEX
- to modify existing LATEX capabilities to work a little differently
- to give LATEX a considerably different look—perhaps one that feels more comfortable to the user or more appropriate for the user's task
- for a combination of these or other reasons

Such changes are possible by changing existing LATEX parameters (e.g., counters and lengths), by creating user-defined commands and environments, and by modifying existing LATEX capabilities.

The following sections of this note sketch several ways of making such changes, so *relative beginners* can get a picture of some of the spectrum of possibilities:

- finding an appropriate package or class (Section 2)
- creating a new command or environment to do something new (Section 3)
- creating a new command or environment to do something differently than LATEX already does it (Section 4)
- changing an existing command or environment (Section 5)

Note that I, too, am far from being an expert $\[Mathbb{E}]_E X$ user. This note sketches some of what I would have liked to have found sketched in one place (rather than having to hunt in books or on the Web) when I was first trying to tweak $\[Mathbb{E}]_E X$.¹

¹Another good place to start is chapter 10 of Helmut Kopka and Patrick W. Daly's book *A Guide to ET*_EX, fourth edition, Addison-Wesley, 2004.

2 Finding an appropriate package or class

One approach to getting LATEX to perform differently than it does "out of the box" is to find an existing class, package, etc.,² that supports what you want. For instance, many people want tighter spacing for list structures than basic LATEX provides, and you can find a number of classes and packages at

http://www.tex.ac.uk/faq

that change various aspects of lists.

However, in this note I will mostly ignore the option of finding another class or package to do what you want and will concentrate on modifying the class or package you are already using. First, it can be a bother to be shifting among classes depending on what behavior you want from LATEX for a particular document because then you have to be aware of subtle differences in how things work from document to document. Second, there are *lots* of existing LATEX capabilities (classes, packages, etc.), and finding the one you want could be a big project. However, if you want to search for a class or package to address your particular need, here are some places to look:

- the T_EX FAQ (http://www.tex.ac.uk/cgi-bin/texfaq2html?introduction=yes)
- one of the CTAN search engines at http://www.tug.org/ctan.html
- the comp.text.tex Usenet group
- *The LATEX Companion* (Frank Mittelbach et al., 2nd ed., Addison-Wesley, 2004)

3 Doing something new

Sometimes you want to extend $\[Mathbb{E}]$ to do something completely new. Of course, creating new commands and environments is covered in many books on $\[Mathbb{M}]$. See, for instance, sections 10.3 and 10.4 of Kopka and Daly's book — cited in footnote 1. Nonetheless, I will give three simple examples. For instance, I always have the \CK command defined that produces a boldface **CK** to mark places where something needs checking in a manuscript I am drafting. My trivial definition for this command is

```
\newcommand{\CK}{\textbf{CK}}
```

Thus, if I give this command after a fact I want to remember to check, e.g., a date as in

was born in $1948\CK{}$

the printed result will be

was born in 1948CK

²Generally, speaking, classes, which come in .cls files, specify the overall structure and look of a document; generally speaking, packages, which come in .sty files, augment or extend basic LATEX or classes.

```
\newcommand{\snfig}[3]{ %insert a scaled numbered figure
   \begin{figure}[htbp]
    %center it and scale the included .eps figure
    \hfill\scalebox{#3}{\includegraphics{fig#2.eps}}\hfill
    %must have done \usepackage[dvips]{graphics} to use \includegraphics
    \caption{\label{fig:#2}#1 \texttt{\small[fig#2]}}
    \end{figure}
  }
```

Figure 1. A new command, /snfig

It is called with three arguments, e.g.,

\snfig{caption}{unique part of fig's file name}{scale factor}

In addition to inserting the .eps figure, the command creates a label by which the figure can be referenced and puts the figure file id at the end of the caption. The latter is deleted from the definition before printing the finished manuscript, but it is very handy to have while continuing to edit the figures and the manuscript. For instance, the following command

\snfig{Schematic of contingency planning}{060}{.4}

results in the output in Figure 2, assuming I previously created the figure in the file fig060.eps.

From time to time I have also added new functionality to LATEX using a new environment. See, for instance, Figure 3, an environment I use to write distinctive notes to myself in the manuscript I am drafting.

Authors' note(s): Here is an example of the result of using the anote environment.

Much of the time, however, you want to change the way an existing aspect of LATEX works. I describe some ways to do this in the following sections.

³"Writing a big book — A first experience with LATEX," David Walden, *TUGboat*, volume 24, number 2, 2003, pp. 211–215.

⁴For more about how figures and tables work, see chapter 9 of Kopka and Daly's book cited in footnote 1.



Figure 2. Schematic of contingency planning [fig060]

```
\newenvironment{anote}% environment for author notes
{\bfseries\sffamily% stuff that goes before my text
\bigskip\noindent{}Authors' note(s):}% more before stuff
{\bigskip{}}% stuff that goes after my text
```

Figure 3. A new environment, anote

4 Doing things differently

As mentioned in Section 2, often people want tighter spacing within a list than the itemize, enumerate, and description environments normally provide. When I first started using IATEX to write my book, I tried to change the parameters of the built-in list environments to tighten spacing, but I couldn't make incrementing and decrementing the various list parameters work correctly.⁵ Therefore, I wrote my own environments with absolute values for the various parameters as shown, for example, in Figure 4.⁶

5 Changing an existing aspect of LATEX

5.1 Changing parameters for an existing part of LATEX

Lots of aspects of LATEX can be modified by changing length and counter parameters.⁷ For instance, you can change the amount new paragraphs normally indent with a command such as

\setlength{\parindent}{.5in}

⁵More about this in Section 5.4.

⁶See figure 4.1 on page 76 of Kopka and Daly's book cited in footnote 1 for definitions of the various length parameters relating to lists. I also followed Kopka and Daly's examples (section 4.4.4 of their book) for how to create a list environment. My brute force approach of just changing the length parameters until I like how things looked lacks generality in more ways than one. Why changing parameters doesn't work to change the function of the list environments is explained at the URL given at the end of the first paragraph of Section 2.

⁷See sections 10.1 and 10.2 of Kopka and Daly's book for a list of such lengths and counters.

```
%define the blist environment for bulleted lists with less
% space between items than in the itemize environment
\newenvironment{blist}{\begin{list} %stuff for before my list items
{\bfseries\upshape $\bullet$}
{\setlength{\labelwidth}{.lin} \setlength{\leftmargin}{.35in}
\setlength{\labelsep}{.lin} \setlength{\rightmargin}{0cm}
\setlength{\parsep}{2pt} \setlength{\topsep}{4pt}
\setlength{\itemsep}{-2pt}
}
} %end of before-stuff
{\end{list}} %stuff for after my list items
```

Figure 4. A new environment, blist

5.2 Redefining an existing part of LATEX

Rather than defining the new blist environment as I did in Figure 4, I could have redefined the *itemize* environment itself. To do this, I would use

\renewenvironment{itemize}

rather than

\newenvironment{blist}

in Figure 4 and also change the introductory comment appropriately.

Giving the modified capability a new name has the advantage that you can call either the normal or modified version in your document. Modifying the normal version and keeping its name has the advantage that you can change the action of this capability in your document without going back to change the name where it is used throughout your document.⁸

5.3 Modifying an existing part of LATEX

For the book I was writing when I began to learn LaTeX, I wanted figure and table captions to use a period rather than a colon after the figure or table number, and I wanted both the table or figure number and the caption itself to be in bold face type. I did this as shown in Figure 5.

I learned how to make these changes by querying the comp.text.tex list where I was told to find the appropriate part of the latex.ltx file (which includes lots of T_EX) in my T_EX distribution, copy that part to the preamble of my document, and make the desired change to the LATEX basic definition now in my preamble, which would redefine this aspect of basic LATEX. I didn't really understand the T_EX but I was

⁸The method of modifying the itemize environment described here throws away much of the generality of itemize as built into LATEX.

```
%change style of figure and table captions
\makeatletter %needed since caption stuff uses an at-sign as letter
\long\def\@makecaption#1#2{%
 \vskip\abovecaptionskip
 \sbox\@tempboxa{\textbf{#1}. \textbf{#2}} %line 5
 \ifdim \wd\@tempboxa >\hsize
 {\textbf{#1}. \textbf{#2}\par} %line 7
 \else
 \global \@minipagefalse
 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
 \fi
 \vskip\belowcaptionskip}
\makeatother %the at-sign must be turned back into a non-letter
```

Figure 5. A modification to an aspect of basic LATEX — captions

able to accomplish my purpose by adding the textbf commands around arguments #1 and #2 in lines 5 and 7 of the figure and also adding the period in both lines.

Sometimes, especially if the change required is to a package other than basic LATEX, I make a private copy of the package giving it a file name that makes it clear that this is not the official version and specify that with my usepackage command. This is discouraged in the LATEX literature for good reasons, I am sure. However, oftentimes I find it more convenient to change a private copy of a package rather than making a change in my preamble as shown in Figure 5, and I don't see much problem with it as long as I rename the private copy of the package.

5.4 Adding to an existing part of LATEX

Suppose for some reason I wanted to set off list environments with horizontal rules. Of course, I could define a new list environment that included these horizontal rules. For instance, to do this I could change Figure 4 to have

```
\newenvironment{hr-blist}{\hrule\begin{list}
```

in place of

```
\newenvironment{blist}\begin{list}
```

I would also add \hrule immediately after \end{list} in the last line of Figure 4 and appropriately change the introductory comment.

However, if I wanted all instances of use of the *itemize* environment to include such horizontal rules without me having to change the name of each use, I could redefine the existing version that mostly uses the normal version but with my modification. I also learned how to do this from comp.text.tex, as shown in Figure 6,

```
%redefine itemize environment so the list items are bracketed by hrules
\let\realitemize\itemize
\let\endrealitemize\enditemize
\renewenvironment{itemize}{%
    \hrule %the beginning stuff
    \realitemize}
    {\hrule %the end stuff
    \endrealitemize%
    }
```

Figure 6. Modifying the itemize environment to include horizontal rules

which required going to the file for basic LAT_EX (latex.ltx) to see how the itemize environment is defined.

There is a significant difference between how I handled the change in Figure 5 and how I handled it in Figure 6. In the former case, you have to copy all the ugly code for handling captions into your program. In the latter case, you only have to know the necessary "hooks" into the ugly code. Unfortunately, in either case, you have to look at the ugly code and understand it enough to make changes or hook into it — unless some nice person just gives you the verbatim change you want, as happened to me in both instances.

Acknowledgments

Karl Berry helped improve the presentation and accuracy of this note, as did the journal's reviewers. I especially appreciate the authors of various books in which I looked and of the people on the comp.text.tex list who answered my questions — they taught me everything I know related to this article.

Biographical note

David Walden is retired after a career as an engineer, engineering manager, and general manager involved with research and development of computer and other high tech systems. Readers with suggestions relating to this paper can contact him via his web site, www.walden-family.com.