

The vulcan Package: a Repair Patch for L^AT_EX

Peter Flynn

University College Cork

Computer Centre

pflynn@imbolc.ucc.ie

URL: <http://imbolc.ucc.ie/~pflynn/>

Abstract

Over many years, T_EX and L^AT_EX systems have proved successful in providing high quality, affordable typesetting from the desktop. This success has tended to disguise some of the less felicitous design decisions made in good faith in earlier days. These are now making it harder to keep L^AT_EX-based systems in line with user expectations, as the defaults remain rooted in a document model and a rendering which reflect the period of L^AT_EX's genesis.

Of the thousands of L^AT_EX packages, many have already tackled the allied problems of providing better formatting and configuration management and of adding new formatting features. This paper presents an attempt to tackle three of the more deep-seated problems: providing an improved document model, an updated rendering, and some fixes for what many users perceive (often wrongly) as bugs.

The document model is compared with current practice elsewhere in the text field, and the fixes answer some of the most common user requests from the various FAQs and questions asked on 'comp.text.tex'. The changes are presented as a standard L^AT_EX 2_ε package which will be submitted to the CTAN when testing is complete.

Background

When Lamport introduced the L^AT_EX structured documentation system in 1983, the T_EX world entered a new era. There was now a standard set of commands (macros) for processing some common document types, which could be learned very simply, 'even by academics and business types', as an esteemed colleague once phrased it. People who wanted acceptable quality typesetting without having to learn the low-level formatting hitherto used by dedicated typesetters could now obtain it, and L^AT_EX rapidly became a *de facto* standard for laboratory and other research documents, both in academia and business.

However, as Lamport himself acknowledged during his informal presentation at the Santa Barbara TUG meeting in 1994, the default typographic styling of the `article`, `report`, `letter`, and `book` document classes leaves something to be desired. The best that can be said about them is perhaps that they formed a standard at a time when no other was available. Regular readers of the 'comp.text.tex' Usenet newsgroup will be familiar with the fre-

quently asked questions about how to change the default styles.

Equally obvious even to the casual user is the lack of some of the most basic features expected by publishers, such as provision for author affiliations and subtitles, and document controls like submission and approval dates.

The concept or model of 'a document' which L^AT_EX represents is a curious hybrid. Early markup systems tended to regard the document as sequences of text blocks (paragraphs, lists, sections, chapters) separated by headings which applied to all text that followed, until the next heading. The end of a section was thus implicit in the occurrence of a new heading. More recent models, notably SGML, tend to regard the document as composed of textual units which are hierarchically 'containerized'; that is, they have explicitly marked start- and end-points¹ and subsections lie physically within the bounds of their parent section. L^AT_EX by default uses parts of both

¹ The quite separate issue of markup minimization for convenience is not relevant: the reference is to fully normalized markup.

models, reflected for example in its uncontained sectioning and in its use of bounded environments.

Development For ten years the concept of all but the most trivial changes to the default styles was beyond most \LaTeX (and even \TeX) users, who had a job to do and were not prepared to commit precious time to making changes in largely undocumented baseline code. The customization features of \LaTeX 2.09 were not extensive, and yet it is a tribute to the robustness of Lamport’s work and the dedication of hundreds of volunteers that many macro packages were written to extend its functionality.

1993–94 saw the arrival of \LaTeX 2 ϵ , intended as an interim solution to some of the problems while work on \LaTeX 3 continued. The second edition of Lamport’s book appeared (1994), updated for \LaTeX 2 ϵ , and also the compendious and wonderful *\LaTeX Companion*, which provided for the first time a substantial corpus of in-depth detail on the internals of the system, as well as copious information on packages and macros.

Structure Nothing was done, however, about the level of markup detail or the default appearance of the document styles. This is understandable when \LaTeX is viewed from the point of view of a standard, but it is regrettable, as it is the visual appearance of the default styles which is the very aspect of the system which draws the most criticism, and is perhaps second only to broken markup as the reason which has contributed most to the poor image and take-up rate of \LaTeX outside the scientific and academic fields.

A second factor in the use of structured documentation systems also came to the fore in the decade before \LaTeX 2 ϵ . The SGML standard² grew in use (and slowly, too, perhaps for reasons not unlike those which faced \LaTeX). The potential for a system which could separate content almost completely from appearance was attractive to many \LaTeX users, and those who worked on \LaTeX 2 ϵ were very much aware of the existence and growth of SGML. Users, too, were not unaffected by the growing popularity of one rather small and restricted application of SGML called HTML (HyperText Markup Language).

The Vulcanize project In concluding that there are areas of \LaTeX ’s implementation which need attention, there is no suggestion that \LaTeX itself — ‘`latex.ltx`’, for example — is in any way ‘bust’, so there is no implication that the core concepts of

\LaTeX itself need to be ‘fixed’. It is contended, however, that \LaTeX ’s default styles and provision of markup are, at the very least, sub-optimal.

Changing this basic provision of markup is not possible as it is frozen, and too many documents rely on the continued existence of the \LaTeX defaults. One solution to this is the development of transparent ‘drop-in’ styles which provide solutions to the problems without affecting anything else. This allows existing files to be processed without changes to the markup, and allows new files to be created with markup which is 100% backward-compatible.

The Vulcanize project,³ and the `vulcan` package which it is developing, grew out of the author’s work in supporting \LaTeX in a research environment and also using it in a commercial production environment. The simple practicalities of this made it clear that while perhaps 85% of style writing could be handled by existing packages and options, there was a repeated necessity for some common features which were not well addressed by the standard packages. The present macros are therefore aimed at fixing these aspects by supplementing the standard \LaTeX defaults with three additional features:

1. extra commands to support many of the requirements for ‘logical’ markup (Lamport, 1988), not a few of which are derived from the experiences of SGML and XML
2. new formats for the default document classes which become operational without any change to existing markup, drawing on practices and developments in typographic design over the last two decades or more
3. document design support for authors and others wanting to make changes, based more closely on the way the typesetting and publishing industry works

The initial stage of work planned for the project is to cover the first item above, and part of the second. A later stage will tackle the more difficult problems of ‘customizable customization’ implied by the third.

It has been noted that while the additional provisions of `vulcan` are in themselves non-standard, there is an opportunity at this stage to try and harmonize some of the work being done by publishers and others, especially for the front matter

² Standard Generalized Markup Language: ISO 8879:1985 (Goldfarb, 1985).

³ As was noted in the author’s recent column in *TUGboat*, (Flynn, 1998) the reference is to the vulcanization of rubber being used to seal leaks.

or metadata, and that the L^AT_EX3 project may be a suitable forum for discussion.

Other work There have been many packages which enhance the default T_EX and L^AT_EX styles, and some of them also provide alternative methods of specifying changes to the styles. Among the best known are probably *blu*, *lollipop*, and *koma*. The last in particular is of interest, as it implemented, for the first time in a T_EX system so far as the author is aware, the page layout parameters favored by Jan Tschichold in later life (Tschichold, 1935). These have been borne in mind, although not directly used, in the layout changes implemented in *vulcan*.

It is no part of this project to detract from these packages, all of which provide much-needed additional features. On the contrary, it is an explicit aim to supplement them by addressing a related but distinct set of problems.

Commands in *vulcan*

This section deals with the primary stage of the project, which is to develop additional commands to support a greater use of logical markup. The greatest part of this deals with the metadata found in document headers and title blocks. Although the current proposals do not relate directly to those of the Dublin Core⁴ or ISO 11179,⁵ the author is monitoring developments in this area, and suggestions from catalogers as well as publishers have been sought in relation to the need for markup to hold ISBNs, ISSN, CIP blocks, etc.

Titling L^AT_EX's `\maketitle` command creates a title block (or title page, if the `titlepage` option to the current class is in effect), using the values supplied in the immediately preceding `\title`, `\author`, and `\date` commands. If the beginner omits one or more settings, there are defaults or error messages provided by L^AT_EX.

These commands in standard L^AT_EX accommodate only the most basic of document identification, and require sometimes extensive additions to handle the requirements of publishers, institutions, or corporate standards. Code such as that in Figure 1 is all too common, and is largely due to the fact that the only way in the standard classes to effect a subtitle is to bind it typographically to the title,

⁴ A proposed methodology for adding metadata to HTML files to aid retrieval on the Web.

⁵ A standard proposing Registries to disambiguate the names used for data elements. This would allow context-sensitive searches to take place without the user needing to know what name an information provider has used for a particular class of information.

and the only way to effect a multi-author representation is to bind it in a similar manner to the `\author` command by formatting. While it can be argued, correctly, that the default styles can be used to achieve footnoted affiliations, for example, the problem is that they are not called that: they have to be called `\thanks`, which is misleading, especially for the beginner. While it is not possible to provide 'syntactic sugar'⁶ for every variation of formatting, the current contention is that the default styles are seriously deficient in their provision of markup and that this is a significant stumbling block to the advancement of L^AT_EX as a serious publishing solution.

One of the greatest problems of acceptance for L^AT_EX is that after all the persuasive sales pitch has been made about the importance of proper markup and identification, and how portable it makes your L^AT_EX, the new user is expected to perpetrate code like that in Figure 1 because the default document classes do not provide for anything else.

Figure 1: Traditional methods of article attribution in L^AT_EX

```
\title{Read Once, Write Many\
{\Large Disambiguating multi-author
attributions in Restoration comedies}\}
\author{Mae West$^a$\\
Mae East$^b$\\
Wile E Coyote$^a$\\
\\
{\small $^a$ University of Short
Island, NY}\}
{\small $^b$ Chicken College, RI}}
```

Read Once, Write Many
Disambiguating multi-author attributions in
Restoration comedies

Mae West^a
Mae East^b
Wile E Coyote^a

^a University of Short Island, NY
^b Chicken College, RI

The problem has long been recognized, and many publishers provide style files for their own journals (*TUGboat* is no exception), but these are clearly very specific solutions, as they address the immediate disambiguation requirement for formatting. It is clear that as publishers slowly move towards more productive use of SGML, the format dependencies are also slowly changing, but in the common case of the author writing a document when

⁶ An alternative way of expressing something; strictly unnecessary, but designed to keep people sweet.

a suitable publisher has not yet been identified, many current formats are simply not useful.

For the three basic commands implemented by the standard classes, the `vulcan` package provides a replacement default title (‘Untitled Document’) and author (‘Anonymous’), and modifies L^AT_EX’s date default to use the day–full-month–year format. In addition, it provides for an optional subtitle, and author affiliation details (job title, department, organization, email address, website, and author bio) for multiple authors in such a way as to make the data more congruent with that required by many publishers. This avoids the need for users to add manual formatting to achieve a reasonably acceptable generic article format. As more style files for known publishers’ journals or series are developed, it is hoped that a common core of similar markup will emerge: one of the known barriers to acceptance of L^AT_EX by publishers is the need to make a significant investment in programming in order to realize the required markup.

Author details, if they appear in the title block at all, usually occur in one of two places: a) immediately after each author; or b) grouped together after all the authors, and referenced with a raised footnote letter. To achieve this in `vulcan` with minimum effort on the part of the author, the positioning of the author details within the markup is used to determine where to place them.

- author details which are included *within* the scope of the `\author` command argument (i.e. before the closing curly brace of the author’s name), will be printed ‘footnoted’ in a block after all author names are complete
- author details which *follow* the closing curly brace of the `\author` command (i.e. they lie after each `\author` command), will be printed as they stand, separately after each author

An example of the first layout is clearly seen in Figure 2. The information which can be provided about each author is defined below (some syntactic sugar has been added to make it more context-sensitive):

Figure 2: Sample title page in `vulcan`.

```
\title{Encoding Boole’s Algebra}
\subtitle{(Markup and Semantics) or \
(Markup and Presentation)}
\author{Peter Flynn
\affil{University College Cork}
\office{Computer Centre}
\email{pflynn@ucc.ie}}
\author{Angela Gilham
\affil{University of Oxford}
\office{Computer Laboratory}
\email{angela.gilham@comlab.ox.ac.uk}}
\conf{ALLC/ACH 2002}
\journal{Markup Languages}\vol5\num4
\submitted{31 December 2002}
\maketitle
```

Markup Languages, 5:4

Encoding Boole’s Algebra

(Markup and Semantics) or
(Markup and Presentation)

PETER FLYNN^a ANGELA GILHAM^b

^aUniversity College Cork Computer Centre
(pflynn@ucc.ie)

^bUniversity of Oxford Computer Laboratory
(angela.gilham@comlab.ox.ac.uk)

Submitted: 31 December 2002

<code>\jobtitle</code>	Synonyms are <code>\position</code> and <code>\rank</code> .
<code>\affiliation</code>	A synonym is <code>\institution</code> , and the abbreviations <code>\affil</code> and <code>\inst</code> also work.
<code>\office</code>	A synonym is <code>\department</code> , and the abbreviation <code>\dept</code> also works.
<code>\address</code>	Postal address.
<code>\email</code> and <code>\webpage</code>	Email address and Web site, if any.
<code>\blurb</code>	This is the author’s short-form biography.

The arrangements for handling multiple authors sharing one affiliation is undecided at the time of writing. Different publishers employ very different formats for this, some attaching the affiliation to the first author’s name, some attaching it to the last, and some using footnote-style raised figures in multiple occurrences. There needs to be more discussion about how to make it easiest for the writer or editor to assign affiliations to author names.

Markup is also provided for the identification of publication-oriented metadata. This often occurs as a running header, at least on the title page, so it is implemented in `vulcan` in that position as left-hand and right-hand heads:

<code>\copyrightholder,</code> <code>\copyrightdate</code>	A copyright notice (holder and year); left head.
<code>\journal,</code> <code>\vol,</code> <code>\num</code>	The name of the journal or other publication, if relevant (with volume number and issue number); left head.
<code>\conf,</code> <code>\sponsor</code>	The name of the conference or event (and the funder or sponsoring organization), if relevant; right head.

Omission of date information works as for standard \LaTeX (today’s date). Extra commands are provided to identify progress through the editorial process: `\submitted` and `\accepted` are more commonly used by authors, and `\received` and `\reviewed` are more commonly used by editors.

An `\ack` command is supplied to identify funding sponsors or acknowledge other assistance.

In all cases the order of the commands is not important, except that author-specific data must accompany its owner directly, in one of the two places explained above.

Sectioning and structural markup The majority of \LaTeX ’s default sectioning structure and appearance has been retained, but some aspects which are the cause of numerous FAQs have been changed.

The `sanshead` option sets all section heads, table and figure captions, headings on front- and back-matter sections, and description list topics in the current default sans-serif font. All display section heads are set raggedright at all times.

The concept of a chapter has been removed from code which is applied in the case of the `report` class, as reports very rarely have chapters.⁷ On those occasions when a very large report in chapters is needed, the `chaprep` option restores the original use of chapters. This not only conforms with observed practice, but solves an important annoyance for the new user.

A new float type, `example`, has been added, using the `float` package of Anselm Lingnau.

⁷ It is unclear how chapters ever got into the report document class in the first place.

Example 1. Example of an example

The semantics of example markup require some care, as it has been noted that in North America, this term appears to be taken to mean exclusively ‘example of verbatim programming code’, whereas in Europe it can be any kind of example of anything, depending on the context of the document, from the brushstrokes in a painting by Cézanne to the right way to boil a floury potato.

Formatting

This work forms part of that identified as the second item in the section “[The Vulcanize project](#)”.

Title block The layout of the title block or page has been radically altered, as described in the previous section: see Figure 2. Instead of the classic centred design, `vulcan` uses a flushleft default layout, with sans-serif document title and subtitle, caps and small caps for author names, and smaller type for author details, dates, and acknowledgements.

However, the header layout is just an instance of the `\maketitle` macro, so it can easily be altered to reflect a new design, without the need to borrow heavily from the class file. The remaining work on parameterization (see the section “[Conclusions](#)”) will address this.

Indentation \LaTeX ’s automatic `\noindent` after section heads can be extended to new paragraphs following lists, quotations, and so on by using the `trapindent` option. This removes an unpleasant visual imbalance where a new paragraph indent would follow the end of the hanging indentation of a fore-going list. The `flushquote` removes the intrusive initial indentation in the `\quotation` environment.

Extended cross-references Extended cross-references have been implemented for parts, chapters, sections, tables, figures, examples, (and lists, where possible), so that it is no longer necessary to name the class of object you refer to. If a reference is to a figure, then the word ‘Figure’ will be inserted automatically. This is similar to the references seen in `TeXinfo`,⁸ which are fully explicit, giving section name and number as well as the page reference, and is already provided for in some class files, as prefixes for counters. The objective is to free authors from the need to carry such details in the head, and to allow text containing labels to be moved from one environment to another without the need to conduct a manual search and check for referential integrity.

⁸ A plain \TeX documentation package common on Unix systems

The `\ref` and `\label` commands have been modified to handle the relevant additional details, so `vulcan` files will continue to work with standard LaTeX in this regard, except that the auto-naming will not occur. Work is ongoing to eliminate some minor conflicts with other referencing packages.

In-line lists One class of lists which is inexplicably missing from LaTeX is the in-line list.⁹ These are extremely common in all classes of document, where they a) provide informality, b) consume less space, and c) enable a list to form a part of a complete sentence. They have been implemented in `vulcan` in the same way as other lists (*ie* as an environment), thus `\begin{inline}`, `\end{inline}`, and `\item` can be used *inside* a paragraph.

There is an optional argument to affect the numbering style, using the same values as for section numbering (`alph`, `Alph`, `roman`, `Roman`, and `arabic`), with a default of italicized lowercase letters (other styles are upright). This argument can also be given in the `\usepackage{inlist}` command if invoked separately, to preset the default.

Hanging indentation Hanging indentation has been introduced on footnotes by default, using the macro suggested in *The LaTeX Companion* (p. 73).

Conclusions

Some parts of the `vulcan` package have been in use in the author’s work area for many years. Parts of the current version of the `vulcan` package have therefore already been used to typeset papers and books, and work is continuing on improving robustness and functionality, and on adding compatibility with the darker recesses of the `book` and `report` classes.

The objective of codifying these macros was to provide a reliable escape route for LaTeX users who needed a better default presentation format than is provided by the basic installation, and at the same time to explore the possibilities for creating a new suite of document class style implementations to provide new LaTeX users with some visual justification for their placement of faith. It can be an unpleasant experience for someone whose only experience of text processing has been with Word or WordPerfect (or worse) to discover that so much manual labor is needed with LaTeX when its proponents have been expounding its superiority.¹⁰

Some experiments have been made with full parameterization for all the layout details in the new header, by way of preparation for the next

phase of the project. This would allow very simple customization, but the very large number of LaTeX ‘lengths’ (TeX’s `\dimenn` registers) needed for all controls would place an unreasonable burden on the use of subsequent packages. The project is open to suggestions about how to achieve a lighter weight solution.

⁹ As this goes to press I notice the new `paralist` package which also implements these lists.

¹⁰ The words ‘mote’ and ‘beam’ seem appropriate here.

Peter Flynn

References

Flynn, Peter. “Typographer’s Inn”. *TUGboat* **19**(1), 353–355, 1998. [1051](#)

Goldfarb, Charles. “The SGML Handbook”. 1985. [1051](#)

Goossens, Michel, F. Mittelbach, and A. Samarin. *The L^AT_EX Companion*. Addison-Wesley, Reading, MA, 1994.

Lamport, Leslie. “Document production: Visual or Logical?”. *TUGboat* **9**(1), 8–10, 1988. [1051](#)

Tschichold, Jan. *Designing Books*. Benno Schwabe, Basel, 1935. Quoted in Lewis, John, *Typography: basic principles*, Studio Books, London, 1963, p.39. [1052](#)