BLUe's Format — the off-off alternative

Kees van der Laan
Hunzeweg 57
9893 PB Garnwerd
The Netherlands
Email: `cgl@rc.service.rug.nl`

Abstract

BLUe's format is an independent set of TeX macros to assist self-publishing authors with creating, formatting, exchanging and maintaining compuscripts. It comes with a user's guide and TeXnical documentation. The format builds upon `manmac`, upon functionalities provided in the *TUGboat* styles, and upon experience gained by the AMS in TeX formatting.

Introduction

In this short paper I will account for the history and global aspects of BLUe's format[1] without going into too much detail. BLUe's format was already on its way as I studied LaTeX, AMS-TeX, *TUGboat* styles, Spivak's, and of course Knuth's works, only I did not realize it at the time. I envisioned a series of manuals: Publishing with LaTeX, Publishing with TeX (PWT for short), Publishing with SGML, and Publishing with you-name-it. The idea was that all these guides would essentially treat the same typesetting goals with only the tool exchanged.

At BachoTeX94, where I presented "Manmac BLUes" and concluded by saying "If only there had been a user's guide for `manmac`, the (LA)TeX world would look different," Bogusław Jackowski challenged me to write such a guide. I replied that I had 2.5D METAFONT on my mind. Once home, however, I thought it over and at the EuroTeX94 in Gdansk that fall I reported about the birth of a BLUe's format which not only built on `manmac` but also accounted for developments since the debut of `manmac`.

At EuroTeX95 I spoke about indexing on-the-fly within one pass, and about the database approach to store and reuse formats, tools (add-ons), references, addresses, and pictures. A tool to assist conversion of a script towards other environments also emerged, called BLUe-2-LaTeX — nicknamed BLUe's convertor assistant.

Now, at TUG96 I can look back, while continuing with 2.5D METAFONT, and say that BLUe's format is a personalized alternative to LaTeX, and off-off for the time being. The macros are shareware. The documentation is free for personal use.

Why?

I needed macros to provide computer-assisted handling of information, be it consumption or production and dessimination, and — something which I understand — would serve a lifetime.

Looking back, BLUe's format system is, on the one hand, an answer to such questions as:

- What has Knuth done in electronic publishing?
- Why?
- How?
- How does he use it?
- What can I do with it?

On the other hand, it integrates developments since.[2]

The user is me,[3] but it might be worthwhile for all those authors who:

- practise self-publishing
- choose for English and ASCII
- adhere to TeX formatting
- favor minimal markup
- like a lifetime tool, with goodies such as stability, consistency, simplicity, portability, generality, flexibility, extensibility
- favor an open, documented system
- prefer an extensible (formatting) language
- support the public domain software credo

[1] BLUe stands for Mr. BLUe — my innocent user and relative of Ben Lee User of *The TeXbook* fame (Knuth 1986).

[2] Compare Douglas Adams' "The answer is 42 if only I knew the question."

[3] Well, . . . and a few others. A Dutch student reported "The way you describe Knuth's markup makes you feel that this is the natural way to mark up scripts."

However, the style-designer, and hacker, might find it interesting to see how to handle a bibliography in one pass, how to index in one pass, how to code a minimal markup macro on top of a two-part macro with nearly the same functionalities, or how to provide for options without parsing arguments.

What is BLUe's format?

BLUe's format is a self-contained set of macros on top of plain TeX and `manmac`, designed to assist authors with creating, formatting, exchanging and maintaining compuscripts. It comes with a user's guide[4] and TeXnical documentation. These last two have appeared as articles in MAPs, the so-called "Paradigm series" (see references).

The format builds upon `manmac` (Knuth's macros to typeset *The TeXbook* and *The* METAFONT*book*), the functionalities provided by the *TUGboat* styles, and experience gained by the AMS in TeX formatting.

It contains a default note format, as well as a format for reports, letters, and transparencies. The letter format allows mail-merge with addresses provided in a database. It also allows the typesetting of address labels. Graphics can be done via TeX alone by Knuth's boxes of `manmac` or his picture environment subset `gkpmac`,[5] and by some turtle graphics macros of my own. A ToC (Table of Contents) utility assists an author when developing a text. Indexing can be done in one pass, completely within TeX.

Add-ons, formats, references, and pictures are stored in databases, such that you don't have to "pay" for what you don't use, nor do you have to be aware of the physical location. The latter is very convenient when working on different systems. A convertor assistant — blue-2-LaTeX — facilitates submitting compuscripts for regular publication.

Usage aspects

Each format has its specific tags. The default note format differs from the letter format, because the task is different. The transparencies format shares the tags with the default but the markup is biased by line-by-line processing, because it centres.

A template for the default note format reads as follows:

```
\input blue.tex \loadindexmacros
                \loadtocmacros
\bluetitle ...
\bluesubtitle ...
\blueissue ...
\bluekeywords ...
\blueabstract ...
\beginscript
\bluehead ...

<copy proper>

\pasteupindex\pasteuptoc
\endscript
```

A `\blue<tag>` takes a blank line as (implicit) terminator, and there are headings for three levels.

The token variables `\author`, `\address`, and `\netaddress` are provided with defaults, so they are not needed in the source file.[6] Bibliographic entries are inserted into the database `lit.dat`; the look-and-feel of typeset references has been borrowed from the AMS. The markup of index reminders is consistent with the approach taken by Knuth for *The TeXbook* and *The* METAFONT*book*.

Special paragraphs. All special paragraphs are coded independently; they can be used with any flavor of TeX. Details are provided in the "Publishing with TeX" user's guide. Some highlights are noted here.

The special paragraphs from `manmac` include `\begindisplay` and `\enddisplay`, `\beginsyntax` and `\endsyntax`, and so on. Items take care of automatic sequencing, be it by numbers or by letters. The end separator is `\smallbreak`.

Verbatims have default ! as escape character. These verbatims are called semi-transparent. File verbatim inclusion goes via `\bluefileverbatim`. Pascal fragments can be marked up by bracketting the fragment by `\beginpascal` end `\endpascal`, without inserting any additional markup into the code — the formatting macros are loaded behind the scenes.

Binary trees can be marked up via `\beginbintree` and `\endbintree`. The specification of data for a table goes via `\data`. Presentation of a table can be varied by using the attributes `\framed` and

[4] A Russian version is in progress.

[5] The macros used to typeset *Concrete Mathematics* (Knuth 1988).

[6] Maybe I should provide these and others in a small file which can be customized more easily. Maybe I can even use `\everyjob` and prompt for the personalized data, with an instruction at the end that `\everyjob` inactivate itself.

`\ruled`. Math markup has been extended by compatible automatic forward cross-referencing. The creation of pictures has been separated from the reuse of pictures. For creation of pictures by TEX alone, Knuth's `manmac` boxes and his subset of the picture environment has been adopted. Additionally, I have developed some turtle graphics macros, which allow, for example, the typesetting of fractals or (rotated) trees, all within TEX alone.

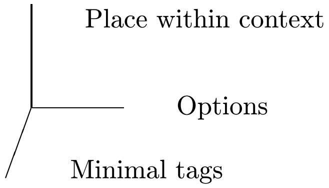
Design considerations

Specifications are needed for:

- look-and-feel in print — typographic design
- markup language — syntax and semantics
- coding style — two-part macros, handling of options, and so on

Typographic design. For the look-and-feel of the result of a blue script in two-column I took over the *TUGboat* layout.

Markup language. I like the orthogonal approach; for example, to treat the following elements:



independently from one other.

Conventions for control sequences. The general conventions include:

- `\begin<tag>`, `\end<tag>` pairs, which take global options via `\every<tag>` and local options via `\this<tag>`[7]
- `\<tag>`, the equivalent one-part macro
- `\blue<tag>`, minimal variant
- token variables `\this<tag>` and `\every<tag>` for user guidance (also called options)
- token variables `\pre<tag>` and `\post<tag>` for controlling positioning within context
- token variables to convey information, i.e., the title
- `\pasteup<tag>` to position the formatted text element within its context

More specifically, the following examples show the control sequences for the section heads:

```
\blue<tag>...  \<tag>{...}  \begin<tag>
                               ...
                            \end<tag>
```

optionally preceded by

```
\every<tag>{...}     \this<tag>{...}
```

Where the place within context (see below) differs from the default, supply:

```
\pre<tag>{...}       \post<tag>{...}
```

Options For compuscripts, it is useful to specify whether the latest version is at hand or not. Another option is to specify `\onecol`.[8] With verbatims it is handy to supply catcode changes and to specify a verbatim input file. Pictures can take different values from the default for their size, and some pictures allow visibility levels.

Place within context Formatting can be seen as placing elements within context. To parameterize the place within context is a good approach because it provides the user or style designer with the hooks needed for the look-and-feel of the result. A good example of this is `\head` and such, where the user can set the whitespace via their versions of the commands `\prehead...` and `\posthead...` . The same can be done for verbatims.

For tables the `\postbtable` token variable is already used for measurements of the table. Usually I place the `\vbox` (which results from my `btable` macro) in a math display, which is consistent with the examples in *The TEXbook*.

I'm still considering how to format tables at the beginning of a text and how to paste up these boxes in the right place; this is similar to the list of references or my supporting ToC. The difference with references is that we have usually many tables, and because box registers are limited available, we might end up in trouble. The next best idea is to also have the tables stored in a database and then load them from there selectively, similar to what I already do with figures.

For the moment there is nothing special about figures to be included in the database, as long as they obey the syntax of a database element. For figures I have not yet provide a `\prefigure` or `\postfigure`. The pictures are delivered in an `\hbox`.

Changing the defaults Token variables can be changed in two ways:

- assign new values to the token variables
- extend the token variable with your extras

An example of the latter follows, showing cases of appending to `\post<tag>` and then proceeding to `\pre<tag>`:

```
\post<tag>\ea{\the\post<tag>
              <your extensions>}
%
```

[7] Consistent with Knuth's `\every`⟨*tag*⟩{...}.

[8] Default is `\twocol` in the note format.

```
\def\preadd{<your extensions>}
\pre<tag>\ea{\ea\preadd\the\pre<tag>}
```

Coding

In this part some ideas behind the coding are supplied. In the Paradigm series of articles, various cases — such as two-part macros and one-part on top, handling of options, coding of items, headings, and so on — were treated.

Conventions. Again, to follow Knuth's lead, especially as done in `manmac`, I have favored the so-called two-part macros at the lowest level.

Starting with the rootname `<tag>`, composite macro names have been created, using prefixes such as `blue`, `begin`, `this`, `every`, `pasteup`, `pre`, `post` and `end`, and postfixes such as `box` and `name`. For example, the user will find `\bluehead`, `\head`, `\beginhead`, `\endhead`, `\prehead` and `\posthead`. However, not all possible combinations exist; for example, `\thishead` is not provided.

For language-dependent names I have introduced definitions of the form `\<tag>name`, for example, which is for use within the environments `keywords`, `abstract` and `references`.[9]

Where available macros have been used, I have added an extra level, to comply with my markup conventions. When the original macro name had to be preserved for the outer level the name was changed, usually by using a prefix pointing to the source.

Beginning of the BLUe script. Design goals include:

- several compuscripts can be processed in one run
- preliminary matter can be provided in any order
- start with a rule in print
- set title in bold
- set authorname indented
- store address information to be set at the end
- set keywords in smaller type
- set abstract 'narrowed' and in smaller type
- set (small) table of contents
- title and issue are used in running headline
- running foot contains draft, page number and copyright

The typesetting of the preliminary matter is done by `\beginscript`.

Headers. Design goals include:

- discourage setting the title alone at the end of a page
- typeset (flexible) vertical space before (large) and after (medium) the title
- gobble spaces at the beginning of the title
- set the title in bold face (and the current size) unindented
- don't indent the first line of the subsequent text

Templates for the coding of heads come from *The TEXbook*'s `\beginsection` and `manmac`'s `\beginchapter`.

The above is only part of the story. In `\report` I also needed to reuse the titles for the Table of Contents, Table of Examples, and the running heads. I could not combine storing and processing on the fly. The best I could achieve was to allow for changes in category codes now and then. This is handy for math material appearing in titles; however, because the circumflex is used for other purposes, its catcode has to be temporarily changed to 7 in titles.

Selective loading. The specifications must:

- load specified parts
- prevent side-effects (work within a group)
- separate typesetting from paste-up
- cope in an elegant way with typos in macro names

At the heart of the coding lies the principle that specified control sequences are redefined. This is in contrast with common usage, which seeks to prevent a format or style file from being overwritten. My use is the opposite: I specify the names I want to have selected via a definition and explicitly require this definition to be overridden.

The list element tag — also called an active list separator — is fundamental. The idea is that this list element tag can be defined such that we can either use its arguments to form a definition — that is, the macro is loaded — or we can just skip them.

One-pass handling of references. The goal is to:

- load only specified references from a database
- store these
- associate the names with their sequence number for cross-referencing
- separate typesetting from paste-up
- cope with typos in the macros names in an elegant way

Various steps have to be integrated to:

- maintain a database of references

[9] This is an aside because I consider English the lingua franca for scientific communication.

- provide a references macro to perform the tasks of selecting and formatting
- specify the list of names as argument to the macro[10]
- load selectively from the database
- format the selected entries
- prevent side-effects
- redefine the names of the selected entries by their number and store these globally
- provide a paste-up tag

I have made use of the FIFO (first in first out) paradigm, so an unspecified number of entries can be supplied.

Practicalities

When a compuscript is in the proofing phase we have to get rid of overfull and underfull `\hboxes` and `\vboxes`. A pragmatic approach has been summarized lucidly by Phil Taylor (1993): overfull `\hboxes` usually result from words which extend beyond the righthand margin.

Get rid of overfull `\hboxes`. In writing in (LA)TEX about (LA)TEX — especially in narrow columns — long control sequence names often yield problems. Generally, the hyphenation algorithm is not in action for such 'words.' Let us assume that these words are set in `\tentt`, then we can activate hyphenation and allow for some room via:

```
\hyphenchar\tentt='055
%and room for example via
\tolerance500
\hbadness=499
\hfuzz=5pt
```

Get rid of underfull `\hboxes` and `\vboxes`. These generally require adjustments to the text itself. Underfull `\vboxes` are usually the result of boxes which don't fit on the page and are therefore moved to the next page or column, leaving an underfull column or page behind them. We can decide to let these elements float via:

```
\topinsert%or \midinsert
<box material>
\endinsert
```

or otherwise adjust or rewrite the text. As yet, no `\raggedbottom` is provided.

What is not included?

No special hyphenations or font selection in the spirit of NFSS has been included, nor has the use of encapsulated PostScript been treated in the "Publishing with TEX" user's guide. As well, there are no macros to handle tables larger than the page size.[11] Of course, this list is by its very nature incomplete, but it rounds out the description of things one wants to know.

Related work

First a disclaimer. It is not possible to really compare `blue.tex` with other formats because each has its own goals. What follows has to be read as a rough indication of how `blue.tex` relates to the works mentioned.

In a sense LATEX provides the functionalities I'm after; however, LATEX is too complex to master for me, and I don't like its coding, or its appearance in print using the standard styles. LATEX aims at a general audience, while BLUe's format is customized for my own purposes. Because of LATEX's complexity, customization is too time-consuming. Moreover, it is subject to change — it is only quasi-static — and from the results I have seen so far, its complexity has increased, alas, and the end to changes not in sight. LATEX does not worry about adaptations to other formats because people involved with it consider LATEX a (de facto) standard. The concept of active documents is not in there, apart from what Knuth has already provided, albeit under different names.

A quick survey of other macro sets revealed that each has its own deficiencies, and none included all the functionalities I was looking for. Doob's macros for typesetting his "Gentle Introduction to TEX" (1990) are too limited, especially when we consider the life-cycle of documents. Berry's `eplain` builds upon `plain` alone. Spivak provides much of LATEX's functionalities in LAMS-TEX, but he avoids `plain`'s math markup. The infotex macros of the Free Software Foundation look as if they had been designed bottom-up. What is the use of a meta-parsing macro — very clever, there is no question about that — if we can do without this functionality in ordinary formats? No attempt for simplicity, nor a set of common markup tags as a foundation to build upon.

The functionality of *TUGboat*'s output routine lies at the heart of the page makeup of `blue.tex`. However `blue.tex` goes much further than the *TUGboat* styles in, for example, the markup for pictures, references, (bordered) tables, verbatims,

[10] It is used twice: for selecting and for formatting.

[11] It is not hard to provide a row separator which would allow splitting of the table, I guess.

math cross-referencing, indexing on-the-fly, all designed for use with AnyTEX; that is, it is not tied up with `plain` exclusively.[12] Add to that the way references and pictures are selected from a database and one would have to agree that `blue.tex` is a leap forward. The coding of options is less monolithic, and the setup via modules with thin interfaces more flexible. `blue.tex` is not a goal per se. It has an eye open to change in general, and adaptation towards other contexts in particular.

Furthermore, as far as I'm aware of, nobody has paid sufficient attention to how to cope with the balance of a *personalized* system in relation to the outer world: using a stable format as basis, adaptable to requirements from the world outside when the need arises.

Availability

`blue.tex` comprises some 118KBytes, `fmt.dat` is 40K, `tools.dat` is 60K, `lit.dat` takes 78K, and `pic.dat` another 38K; `address.dat` is still very small. The transparencies macros have been incorporated into the database `fmt.dat`, and, as of March 1995, they are together with the letter and report formats. Although this paper bears the title "BLUe's Format," the filenames are in lowercase letters only: `blue.tex`, `fmt.dat`, `tools.dat`, `lit.dat`, `pic.dat`, `address.dat`, and the article `fmt.art`, in addition to the PWT user's guide. All files available[13] on the NTG's 4AllTEX (double) CD-ROM from 1995, CTAN (in `info/pwt`, and from the NTG and GUST fileservers.

Looking back

The hardest thing has been to not mix up all the influences I have been exposed to. To stand back and decide about simple coding conventions was not easy, especially when the material has not yet been mastered completely. When to use token variables and when to use definitions is also difficult when trying to prevent confusion at the user level. It is tempting to introduce parameter separators, but I have refrained from those in the outer-level macros, and have instead used the straightforward parameter mechanism. Having gained some practice with the `\this<tag>` idea, I am still very pleased with it, especially because of its local character and absence of parsing overhead.

Looking forward

BLUe's format is stable. I use it whenever I have to typeset notes, letters, submisions for TEX bulletins, or transparencies. I guess it will grow along with my needs. The "Publishing with TEX" user's guide is also stable, although now and then some parts get improved. For some time to come, I'll be working on METAFONT/MetaPost. After that I'll embark on Literare Programming — who knows how that will influence BLUe's format.

Acknowledgements

I owe the TEX community much for borrowing free material to build upon, starting with what Don Knuth gave to the world. In particular I would like to thank the NTG for allowing me to bring out earlier versions of my papers in the MAPs publications; I would also like to thank GUST for having invited me to lecture at BachoTEX94 about `manmac`, and thank the attendees for having asked for a `manmac` user's guide. Thank you!

Conclusions

With respect to format or style development, macro writing is a special way of programming; to embed this within the realm of software engineering is a real challenge, and much needed.

It is still wishful thinking to hope for a full-fledged simple format which is easy to use, and whose marked up copy can be easily transformed into a representation suitable for other situations. `blue.tex` comes close, mainly because I have thought more than twice about what Knuth has provided, before writing anything of my own, and because of my handful of outer-markup tags (very few indeed). Therefore, I conclude that:

– a variety of tools is needed
– `blue.tex`≡ `manmac` + *TUGboat* abbreviations and output routine + variant formats + add-ons
variant formats ≡ report + letter + transparencies
add-ons ≡ bordered table + crossrefs + `gkppic` + turtle graphics + verbatim suite + index set + . . .

[12] For example, I have made the index macros separately available to cooperate with AnyTEX.

[13] Macros are shareware. The price is the minimum of US$25 and the local price of 20 loaves of bread, in dollars. CyrTUG members can join for free.

- the principles include consistency, portability, longevity, flexibility, intelligibility, extensibility, and correctness
- user documentation is paramount
- when attempting to keep up with Knuth, be realistic

When you consider book formatting you might consider customizing `manmac`.

My added value. I have integrated `manmac` and developments since, such as:

- separation of specification and paste-up
- specification of options via `\this<tag>{...}` and `\every<tag>{...}`
- parameterizations of `\pre<tag>{...}` and `\post<tag>{...}`
- formats, tools, addresses, literature and graphics databases
- one-pass bibliography handling
- formatting references in the spirit of the AMS
- cross-referencing of math and bibliography items
- outer and inner markup separation
- a verbatim suite
- a bordered table macro
- a conversion tool for outer markup tags
- report, letter and transparencies formats, in addition to the default note
- one- and two-column options for notes
- one-pass indexing on-the-fly
- ... various other add-ons

Last but not least, a user's guide and TEXnical documentation are part of BLUe's format system.

With respect to coding I have added the paradigm — a real pearl, to paraphrase Bentley[14] — to systematically and automatically add a one-part macro with the same functionalities on top of a two-part macro. My thesis is that the users can benefit from `blue.tex` to typeset like craftsmen, to achieve the quality of Knuth.

The extra bonus is stability, and simplicity as well. The disadavantage is investment in learning which, by trial-and-error, takes a substantial amount of time. It all has to do with your attitude, whether you believe that TEX proper will serve a lifetime, and whether you like to invest in learning plain and `manmac` as a basis. The reward is freedom. You are no longer dependent upon the gurus for what-and-when. Furthermore, I adhere to De Vinne's adage that "The last thing to learn is simplicity."

The user is encouraged to peruse *The TEXbook*, and to also look at the markup in the file. I hope you will arrive at the same conclusion as I have: Knuth's markup is unsurpassed. Then simply practise Knuth's minimal markup, have fun and all the best.

References

Adams, D. *The Hitchhiker's Guide to the Galaxy*. New York, Harmony Books, 1989.

Berry, K. `eplain`. Available from CTAN(`/macros /eplain`).

De Vinne, T.L. *Invention of Printing*. 1876. New York: Hart, 1969.

Doob, M. *A Gentle Introduction to TEX*. *TEXniques* 13, Providence, Rhode Island: TEX Users Group, 1990.

Knuth, D.E. *The TEXbook*. Reading, MA: Addison-Wesley, 1986.

Graham R.L, D.E. Knuth, O. Pastashnik. *Concrete Mathematics*. Reading, Mass.: Addison-Wesley, 1988.

Spivak, M.D *LAMS-TEX, The Synthesis*. Texplorator, 1989.

Taylor, P. "A Pragmatic Approach to Paragraphs." *TUGboat*, 14,2 (1993), pages 138–140.

van der Laan, K. "Paradigms — Headache?" MAPS 94.2 (1994), pages 212 – 214.

van der Laan, K. "Paradigms — Plain's Items Extended." MAPS 94.2 (1994), pages 210 – 211.

van der Laan, K. "Paradigms — Two-Part Macros." MAPS 95.1 (1995), pages 200 – 204.

van der Laan, K. Publishing with TEX: BLUe's Selection. Garnwerd, Holland, 1995. Available from CTAN (`info/pwt`).

[14] Jon Bentley writes a regular column, "Programming Pearls", for the *Communications of the ACM.*