BLUe's Verbatim— the selection

Kees van der Laan

Abstract

A suite of macros for verbatim mode has been provided, which are simple, concise and flexible.
The functionalities are: verbatim text, and file verbatim inclusion, with 'options' for numbering and enabling of metacode.
Options can be supplied via the toks variable `\thisverbatim`, and globally via `\everyverbatim`.
The place within context can be handled via the defs `\preverbatim` and `\postverbatim`.
The macros are context independent and can be used at the inner level with AnyTEX.

Keywords: Education, escape character, formatting, generic style, semi-transparent, TEX, verbatim mode.

1   Introduction

This is the third paper[1] in the series where I select a BLUe-way of formatting with TEX from the multitude of confusing possibilities. In the previous papers I have addressed the audience. To be honest the real reason is that the articles function for me as a mental aid, a summary of what I have once understood.

When writing about (LA)TEX in TEX verbatims are needed. By verbatim text I mean that what is keyboarded is formatted in teletype font with spaces and end-of-lines obeyed. The special characters of TEX are set with catcode 12. The TEXbook provides in Appendix D.3 Verbatim listing

- `\listing`, 380, for file verbatim with line numbers added in the typeset result
- `\verbatim`, 381, for inline verbatim, with an alternative coding on page 382
- `\beginverbatim ... \endverbatim`, 382, for typesetting everything in between in verbatim, except for `\endverbatim`, the termination tag
- typesetting 'tabs verbatim (that is variable number of spaces),' really advanced.

LATEX provides the functionalities

- `\verb`*<char><verbatim text><char>*, for inline verbatim, and
- `\beginverbatim ... \endverbatim`, for a displayed result.

TUGboat.sty provides a fancy set of macros which can't be used easily in other contexts. The functionalities are rich, however. The best and most complete I have seen.[2] See my TUGboat BLUes, for an elaborate discussion.

The aim of this paper is to select a simple, flexible, practical, and generic suite.

2   What is the problem?

The problem is apparently the incompleteness of the available macros for practical purposes. Be it that options are lacking, that there is no file verbatim, or that the encoding is dependent upon the general mechanisms of the environment.

For BLUe the problem is the selection of a practical and general suite, which is simple, intelligible, flexible, and generic. With the latter I mean that it can be used within any context, be it plain, manmac, LATEX, or $\mathcal{AMS}$-TEX, to name but the most notorious flavours of TEX.

In keeping it simple I sacrificed uniform markup conventions. I chose to start verbatims with `\verbatim` but to end the *mode* via `<escapechar>verbatim`. This looks strange. I think it does not have to be perceived as such, especially when we realize that we can assume some TEX knowledge with authors who write in TEX about (LA)TEX, and who will appreciate 'exceptions to the rule.'

3   Design

Functionally, I needed in practice[3]

```
%1. To handle via default ! escape char
%    -- other fonts
%    -- footnotes
%    -- emc (enable metacode)
\thisverbatim={\emc
   \def\ftxt{Footnote text typed in
           on more than one line.}}
\verbatim
Some <meta code> and
blah, blah, ...    !it
Now text in italics!tt
and back again in tt
footnote!footnote*!bgroup !ftxt!egroup
!endverbatim
```

---

[1] The first two papers are BLUe's Bibliography, MAPS 93.2, and BLUe's Transparencies, MAPS 94.1. Projected are BLUe's Pictures and BLUe's Format. BLU stands for Ben Lee User of the TEXbook fame. BLUe is its cousin, adopted by me. The idea is that like in Math the basics are explained to jump off from.

[2] I don't know why ligatures are not handled correctly in tugboat.sty (for example ?', TB 381).

[3] These examples can also function as samples.

```
%
%2. To handle numbering and
%    verbatim file inclusion
\everyverb={\numvrb}
\thisverbatim={%
    \catcode'\!=12 %is default escape char
    \catcode'\|=12 %is used for inline verb
    \input vrb.tex
    \makeescape\*}%To terminate
\verbatim
Extras after file
*endverbatim
%
%3. To restart (line)numbers
\thisverbatim={\vrblin0 }%Or \numvrb
\verbatim
Just some text with
ligatures such as ?' switched off (TB 381).
line numbers restarted via
\vrblin0 (in general via \numvrb).


After two blank lines.
!endverbatim
%
%4. Inline verbatim via minimal tags
%Minimal | tag for inline verbatim
\makeactive\|
\def|{\bgroup\setupverbatim
        \the\everyverbatim
        \the\thisverbatim
        \def|{\egroup\thisverbatim{}}}
\thisverbatim{\emc}
Before |<inline verbatim>| after.
```

Basic to the design is the concept of `\this<foo>`, which is analogous to Knuth's `\every<foo>`. With respect to switching the verbatim mode off, I adopted Knuth and Levy's use of the escape character. The mode is finished essentially by `!egroup`. This is a royal road towards simple coding. An escape character is often needed after all, so why not make use of it?

When I looked carefully in TUGboat.sty's verbatim coding I found that the verbatim text is stored as parameter, and therefore restricted. In manmac the verbatim text is set within a `\vbox`, and therefore practically also limited. I thought of line-by-line processing but abandoned the idea once I appreciated the idea of

　　verbatim mode
　　`\everypar`, and
　　the use of the escape character

for termination.

Furthermore, I like the idea of hooks via `\preverbatim` and `\postverbatim`, to provide the user the opportunity to supply code to be executed before and after. In practice I hardly used the latter possibilities, however.

A bonus which comes with the escape character is that it can also be used to control the counter for the line number in numbered file verbatims. For example the line numbers can be suppressed for parts or appropriately initialized, as demonstrated in the listing of vrb.tex at the end of this paper. This approach is very useful when the macro file is appended by its contents with references to the line numbers in order to locate macros.[4]

All what has to be done is to include `%!numvrb` in the file when one likes the next line to start numbered, or to include `%!vrblin100` when one likes to give the line counter the special value 100. To complete it one can suppress the numbering via inclusion of `%!nonum`. This approach does not hinder the original purpose, because it is done via comment lines, and extra comment lines are anyhow wanted to group code parts. Ça va sans dire that this looks nice to me.

4　Conclusion

The macros obey the adage:

　　　prototype, revise and rewrite.

My alternative codings to the verbatims of TUGboat.sty—as released in TUGboat BLUes—can be seen as the prototypes. vrb.tex is an improvement with respect to simplicity, flexibility, speed, independence from the context, and the little use of resources. Confusing was that `\char'174` is the vertical bar in ASCII and `\tt`, but the `\Dash` in `\rm`.

I too found verbatims quite confusing. Now I feel confident, because I understand the process and I only have to remember the specialities of ! and |. To end the verbatim by *<escape char>*`endverbatim` is also not hard to remember. As usual, things you do yourself look easy. It is only hoped that BLUe will feel the same way about it. My case rest.

References

The TEXbook and the LATEXbook are not explicitly mentioned, because I assume that these are omni-present.

[1] Knuth D.E, S Levy (1987): The CWEB System of Structured Documentation. (FTP: labrea.stanford.edu, in directory /pub/cweb)

[2] Laan C.G van der (1994): TUGboat BLUes—how TEXies do it. MAPS 94.1.

[3] Salomon D (priv.comm.)

[4] Schöpf R (1989): A new implementation of the LATEX verbatim[*] environments. *TUGboat* 11, no. (2), 284–296.

Appendix: The file vrb.tex

Note that ! and | have the meaning of escape character and starting the inline verbatim, respectively. The ! can be obtained via !!, in verbatim mode, and via just ! outside. The vertical bar can be obtained via !vrt in verbatim mode, and via `\vrt` outside. On the other hand when verbatims contain |-s, it is easier to change the catcode of | into 12 via `\thisverbatim`.

```
%vrb.tex Feb 1994 Version 1.0
%Author: C.G. van der Laan, Hunzeweg 57,
%    9893PB Garnwerd
%    The Netherlands
%    05941-1525, cgl@risc1.rug.nl.
%Purpose: Verbatim macros via plain TeX,
```

---

[4] In studying Manmac, the TUGboat styles I missed the table of contents with references to the line numbers of the code.

```
  %          to be used with AnyTeX.
  %
1 %User toks variables
2 \newtoks\thisverbatim
3 \newtoks\everyverbatim
4 %
5 %User customization
6 \let\preverbatim\medskip
7 \let\postverbatim\medbreak
8 %
101 %User 'options': \numvrb
102 \newcount\vrblin
103 \def\numvrb{\vrblin0
104    \everypar{\advance\vrblin1
105      \llap{\sevenrm\the\vrblin\quad}}}
106 \def\nonum{\everypar={}}
107 %            : \emc
108 \def\makeescape#1{\catcode'#1=0 }
109 \def\makeactive#1{\catcode'#1=13 }
110 {\makeactive\<
111  \gdef\emc{\makeactive\<%
112    \def<##1>{$\langle##1\rangle$}}}
113 %
201 %User macro
202 \def\verbatim{\preverbatim\begingroup
203    \tt\setupverbatim
204    \the\everyverbatim\relax
205    \the\thisverbatim\relax
206    \verbatimgobble}
207 %
208 \def\endverbatim{\endgroup\postverbatim
209    \thisverbatim={}}
210 %
211 \def\setupverbatim{\makeactive\'%
212    \let\!=!\makeescape\!%Knuth&Levy
213    \def\par{\leavevmode\endgraf}%TB381
214    \obeylines\uncatcodespecials
215    \obeyspaces}
216 %
217 {\obeyspaces\global\let =\
219  \obeylines\gdef\verbatimgobble#1^^M{}%
219  \makeactive\' \gdef'{\relax\lq}}%TB381
```

```
220 %
221 \def\uncatcodespecials{\def\do##1{%
222    \catcode'##1=12 }\dospecials}
223 %
251 %Minimal | tag for inline verbatim
252 \def\vrt{{\tt\char'\|}}\makeactive\|
253 \def|{\bgroup\tt\setupverbatim
254      \the\everyverbatim\relax
255      \the\thisverbatim\relax
256      \def|{\egroup\thisverbatim{}}}
257 \endinput %14/2/94 cgl@risc1.rug.nl
258 %
```

Contents
Newtoks
%
History of changes
Febr 1994   Release Version 1.0