BLUe's Letters

With love

Kees van der Laan

Hunzeweg 57,
9893 PB Garnwerd, The Netherlands
cgl@rc.service.rug.nl

Abstract

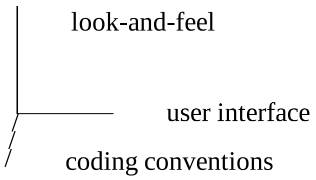
The backgrounds, use, design, and coding of BLUe's Letters format have been discussed. The purpose is to format a letter, merge it with address(es) from a database, and typeset it all with the appropriate background such as logo and the like, completely within TEX. Separate labels can be obtained too, either specified by name or searched for by pattern.

Keywords: Active list separator, addresses, address labels, compatible extension, databases, data integrity, education, fifo, lazy evaluation, letter, list element tag, macro writing, mail-merge, no-nonsense, pattern matching, plain TEX, reusable software parts, set macros, software engineering.

1 Introduction

What makes a letter different from a note? From an article? A key issue is the merging of the background (logo and the like), with address(es) from a database, and the letter proper. Of course the look-and-feel is much different too, although scientific communication was done by letter in the old days, instead of via articles in journals for an anonymous audience. This sets the scene when we like TEX to take care of it all.[1]

When designing a format three issues have to be dealt with, preferably as a mutual orthogonal set



In the sequel I'll talk about the

- use (and customization)
- look-and-feel
- user interface, and
- coding.

The `\letter` format is part of BLUe's format system, which is a personalized format and compatible with plain TEX.

I started from Knuth's (example) letter format, *The TEXbook* Appendix E. Knuth could do with the following handful of tags

```
\letterhead \address \body \closing \endletter
```

If this is such a great approach, how come that it has not been taken over? IMHO, with all respect, it was too difficult to customize. Moreover, at that time many a TEXie was busy porting the system to the various platforms. Nowadays I would like to

- store addresses in a database, and use these from there
- have a simpler letterhead
- let `\headline` take care of the headers
- use fewer fonts
- use the known sender — remember BLUe's format is yours, it knows about your context
- use window envelopes
- introduce token variables for `\subject`, `\yourreference`, and `\ourreference`
- make positioning of the address flexible, such that it can move to adjust to your window envelope
- adapt `\makelabel`[2] to work independenly with the (address.dat) database (Address labels are handy for carbon copies among other things)
- substitute `\beginscript` for `\beginletter` and ditto for the `\endscript`.[3]

2 Notations and definitions

`\ea` denotes `\expandafter`. `\nx` denotes `\noexpand`.

An important notion is what I call 'list element tag.' This element is a chameleon and takes any actual meaning to process the list elements. It is very close to the concept

[1] The advantage of handling all by TEX is that we don't have to bother with jargon from other tools. Another advantage is the stability, because TEX has been frozen.

[2] I named it `\makelabels`.

[3] Well, I decided to allow for the aliases `\beginletter` and `\endletter` too.

of active list separator. Knuth used this already[4] in *The TeXbook* Appendix D.2 List macros, the `\\`s, with the annotation 'But in fact, the `\\` separators are enormously useful, because we can define `\\` to be any desired one-argument macro, and then we can execute the list!' I prefer the name list element tag instead of (active) separator, because it does not really separate — the first element is different — moreover, the idea is not restricted to one argument. I have used the list element tag with name `\lst`. Agreed, to use the list element tag is unusual and confusing at first, but once you get the hang of it, you will appreciate it. It is an eye-opener.[5] Knuth has used the list element tag, `\ansno`, for formatting the answers of the exercises for *The TeXbook*. `\ansno` takes two arguments ended by period . and colon : respectively, and followed by the answer proper, typeset on the fly.

3 Use

The scripts for typesetting a letter vary with whether a typed-in letter or a stored letter will be merged. It also depends on the number of addressees. Extremes are a letter addressed to one person, or a letter sent to the complete database.

3.1 Typesetting letters

Below push-the-button scripts have been supplied by example. I assume that the addresses are available in the address.dat database,[6] as a list of elements, each obeying the syntax: list element tag, `\lst`, followed by a name tag, and a group. For addresses this comes down to the following.

```
\lst\<name>{<salutename>
\\<fullname>
\\<affiliation>
\email{...}
\phone{...}
\fax{...}  }
```

For `<tag>` I take the name with initials. `<salutename>` is the name to be used after 'Dear' or so. `<fullname>` is the name as it will appear on the envelope, and in the headline of the follow pages. The affiliation can be split over several lines — visual markup — but do precede each line by `\\` — a lower level list element tag :-) — to denote for the time being a new line in the result. I adopted the convention to supply country names in capitals. Consistency facilitates the use of the `\search` macro, to match patterns.

Example *(Typeset a typed-in letter)*

```
\input blue.tex    %Personalized format
\letter            %knows logo, sender etc.
\subject{...}
\ourreference{...} %just numbers
\yourreference{...}
\addresses{\<name>}%to load address(es)
\addressee{\<name>}
\beginscript
\dear              %uses \addresseename
....
\sincerely         %knows about you
\cc ...            %set \item{cc.} ...
\ps ...            %set \item{P.S.} ...
\appendix{<appendixtitle>}
<appendixmaterial>
\endscript
```

The `\appendix` control sequence takes one argument. It starts a new page, but continues page numbering. `\beginletter` is an alias for `\beginscript`, analogous for `\endletter`.

Example *(Typeset a stored letter)*

If the letter proper is stored in letter.tex, then the outer-level markup looks even simpler.

```
\input blue.tex \letter
\subject{...}
\ourreference{...} %just numbers
\yourreference{...}
\letterto{\<name>...}
```

If we wish to send the letter to the complete database replace the last line by `\lettertoall`.

3.2 Making address labels

Although the addresses are integrated with the letter and suitable positioned for window envelopes, the separate address lables can be handy for carbon copies.

Example *(Some or all labels)*

```
\input blue.tex \letter

\makelabels{\knuthde\ntg}
\bye
```

All address labels from the database, specified in token variable `\addressfile`, can be obtained via

```
\input blue.tex \letter
\makealllabels
\bye
```

Labels which match a pattern emerge after

```
\input blue.tex \letter
\search{RUSSIA}
\makesearchlabels
\bye
```

[4] Of course. . . .

[5] The history is that I became familiar with the list element tag when working on the Tower of Hanoi in TeX. Since then I also used the fifo paradigm and could get rid of the list element tag at the expense of recursion. Now I combine the best of both worlds. The user only has to specify the list of names, and when reuse is in sight the macros fill the token variable `\namelst` with the names and the list element tags inserted, to facilitate later processing. When I started to work on handling references I first provided a list of definitions and loaded all those definitions from file. At that time TeX definitions were appropriate as entries. Only when I considered to load selectively, I found to use the `\def` as list element tag. Finally, I realized that the `\def` is confusing and dangerous and returned to the idea of lists preceded by a less vulnerable list element tag. The double backslash is so heavily used for e-o-l that I refrained from following Knuth on that. Furthermore, I found it convenient to associate a name to each list element.

[6] The file `\addressesfile` contains the name of the address database.

3.3 Extending the address.dat database

Example *(Entry of address.dat database)*

```
\lst\ntg{NTG

\\Nederlandstalige \TeX{} Gebruikersgroep
\\Postbus 394
\\1740 AJ Schagen
\\The Netherlands
\email{ntg@nic.surfnet.nl}
\phone{}
}
```

To add an address just adhere to the syntax, and include the new address in address.dat, in the right order, i.e., alphabetically sorted.

Check for data integrity

In order to make sure TEX can scan your address.dat, after you have added addresses, make a table of contents as follows.

Example *(Table of contents of address.dat)*

```
\input blue.tex

\contentsdatabase{address}
\bye
```

As a result there are no pages of output (that is OK!), and the file contentsaddress has been made. Note that I have already included `\newwrite \tocaddress` in blue.tex.

4 Customization

When starting to use BLUe's format system a user is asked to personalize the format. Name and affiliation have been customized, of course. For letters the logo is relevant, next to the bank account or so which can be supplied in the toks variable `\businessaccount`.

```
\businessaccount{Giro: C.G. van der Laan
   no {\oldstyle1321224}}
%
\def\logo{\copy\ntglogobox}
%
\def\ntglogo{\vbox{%
  \hbox{{\calx N}ederlandstalige}
  \hbox{\hskip1em{\calx T}\kern-.2ex
        \raise-.5ex\hbox{E}\kern.1exX}
  \hbox{\hskip2em{\calx G}ebruikersgroep}}}
\setbox\ntglogobox\ntglogo
```

The skip variables `\haoffset`, and `\vaoffset` — mnemonics horizontal address offset, vertical address offset — can be used to adjust the address to the window of your window envelopes.

To allow for your address database, say `<name>.dat`, include

```
\addressesfile{<name>}
\newtoks\toc<name>
\immediate\openout\toc<name>=contents<name>
```

If your file which contains the letter is different from letter.tex provide `\letterfile{<filewithletter>}`. Customization towards a different design is beyond the scope of this paper.

5 Look-and-feel

A letter is characterized by a first page with a special header part, an address window, letter beginning, and footer. On follow pages it must be clear which first page each follow page is supposed to follow.

5.1 Letter

The following is a sketch of the design. Just the rough outlines.

I decided to set a logo left upper and sender affiliation right upper. Then follows, separated by a line, reserved space — let us call this a window — to typeset the address in. This is followed by reference information left and date right. The letter opening starts with a salute followed by the letter proper, possibly split over several pages. At the end the signature of the sender is set right. The back matter consists of P.S., cc., or Appendix with a title. The first page footline is separated by a partial horizontal rule below which the business information has been set.

A follow page takes a modest headline with addressee left upper and subject, our reference, and date right upper. The footline contains just the page number.

5.2 Address label

Each label is set with the address in the middle and the sender affiliation left at the bottom of a window of 6×13 cm.[7]

6 User interface

How should the markup language for the user look like? In agreement with BLUe's format the letter is a script with info elements preceding the script proper. I consider addresses also information, similar to the references of an article. The addresses can be loaded selectively from the database via

```
\beginaddresses
\<name>...        or   \addresses{\<name>...}
\endaddress
```

The use of the `\script` tags is in agreement with blue.tex.

6.1 Letter

The information tags appear before `\beginscript`, and can be looked upon as token variables.[8]

```
\addresses{...}   %load list of addresses
\subject{...}
\ourreference{...}%digits because I fancy
                  %frisky \oldstyle
\yourreference{...}
```

The blue collar workers read

[7] Well it depends. `\twocol` will yield smaller labels.

[8] They can be supplied in any order. This a general aspect of BLUe's format system.

```
\addressee{...}   %splits off name
\beginscript...\endscript
```

In the letter proper ordinary TEX markup can be used, next to the definitions[9]

```
\dear
\sincerely
```

The back matter has the markup tags

```
\ps
\cc
\appendix{...}
```

Special cases are handled by

```
\letterto{...}    %list of name tags
\lettertoall
```

6.2 Address label

The markup tags read

```
\makelabels{...}  %name tags
%
\search{...}
\makesearchlabels
%
\makealllabels
```

7 Coding

In BLUe's format system two-part macros are the starting point. A one-part macro is provided on top of it. Specification of information and actual typesetting have been separated.

7.1 Handling addresses

In `\beginaddresses` the names are first defined and then overloaded by their entry from address.dat, the (default) file specified in `\addressesfile`. Also the token variable `\namelst` will contain the list of the names of the loaded elements each preceded by the list element tag, `\lst`, to facilitate execution of the list.

```
\def\beginaddresses#1\endaddresses{%
  \def\process##1{\ifx\undefined##1
    \let##1\addresserror\else
    \message{\Dash\string##1
         already loaded.\Dash}\fi
  \namelst\ea{\the\namelst\lst##1}}
  \fifo#1\ofif%end defining all names
  \loadselectivefrom{\the\addressesfile}}
%with on top the trivial variant
\def\addresses#1{\beginaddresses
               #1\endaddresses}
%and auxiliary
\def\addresserror{Address not in databasea
    (Sorry).\loaderror{Addresses}}
```

`\loadselectivefrom` reads as follows.

```
\def\loadselectivefrom#1{%#1 address or lit
 \def\lst##1{\ifx##1\undefined\ea\gobble
    \else\ea\gdef\ea##1\fi}\input #1.dat}
```

7.2 The markup for a letter

We have two aspects the user macros, and the page makeup.

User macros

`\letterto` executes `\processletter` for each address supplied as argument. `\processletter` provides the script and takes the information it needs. It also splits the address. Note that the `\letterto` was needed to alleviate for the user the trouble to retain values for the next letter, to be processed in the same run. This is induced by my use of `\headline`, which is changed for follow pages.

`\lettertoall` sends the letter to all addresses from the address database as specified in `\addressesfile`. For all those addresses `\processletter` is executed.

`\addressee` takes the first two lines apart from the address entry for `<salutename>` and `<fullname>`, and sets the address in the affiliation box.

```
\def\letterto#1{{\everyscript{\notlastscript}
   \storedvsize\vsize
   \storedheadline\headline
   \storedfootline\footline
   \beginaddresses#1\endaddresses
   \let\lst\processletter\the\namelst}}
%
\def\lettertoall{{\everyscript{\notlastscript}
   \def\lst##1{\processletter}%pick address
   \input\the\addressesfile.dat\relax}}
%
\def\addressee#1{\ea\splitaddress#1\egroup}
%
%with at the lower level
%
\def\processletter#1{{%#1 name or address
   \headline\storedheadline
   \footline\storedfootline
   \vsize\storedvsize
   \addressee{#1}
   \beginscript
   \input\the\letterfile\relax
   \endscript}}
%
\def\splitaddress#1\\#2\\{\addresseename{#1
   \unskip}\fullname{#2\unskip}\setbox
   \affiliationbox=\hbox\bgroup#2\\}
```

Page makeup

usually implies the output routine, OTR for short. However, for this application I could do without modifying the OTR.

In contrast to the letter format as supplied in Appendix E of *The TEXbook*, I heavily used `\headline` and `\footline`. The big deal is the headline. It consists of a vbox of height 3cm, with logo and sender affiliation, followed by an `\hrule`, and the address window. Note that `\headline` is redefined for use on subsequent pages. This implies that the original `\headline` has to be stored for the next letter. This holds too for `\footline`. Note that the `\vsize` for follow pages is changed in `\footline`.

[9] I don't know of a trick to remember that `\dear` is a definition and not a token variable to represent the language dependent word 'Dear.' Perhaps it is good to remind that the letter format knows about the addressee, once the name of the address has been supplied.

```
\headline{\line{\vbox to3cm{%
\line{\logo\hss
  \vbox{\hsize.33\hsize\small
   \the\address\\\the\netaddress}
}\kern3pt\hrule\vss}\hss}%
\addresseewindow
%and for follow pages
\global\headline{\line{\vbox
  to3cm{\vss%Implicit vspace
  \line{\tenrm To: \the\fullname
   \hss\today/\the\subject/\the\crowner\
   \oldstyle\the\ourreference}\kern2pt\hrule
   \vss}}}%end follow \headline
}%end \headline
%
\footline{\line{\vbox{%
  \kern\baselineskip\hrule\kern.5ex
  \hbox{\strut\the\businessaccount}}\hss}
%and for follow pages
  \global\vsize19cm
  \global\footline{\line{\null
   \hss--{\oldstyle\the\count0}--\hss
   }}%end follow \footline
}%end \footline
```

The `\addresseewindow` sets the address in a window of 4cm height and width `\hsize`. The positioning is biased by the values of the skip variables `\vaoffset`, and `\haoffset`.[10] This is followed by the information elements.

```
\def\addresseewindow{\line{%
\vbox to 4cm{%    %Window height Dutch
                  %envelopes
\vskip\vaoffset   %To shift address vert.
\leftskip\haoffset%To shift address hor.
\unhbox\affiliationbox
\vss}\hss}%end line
\line{\hbox to\longindentation
   {\hbox to8ex{Subject\hss}:
     \the\subject\hss}\today\hss}
\line{\hbox to8ex{\small Our Ref\hss}:
   \the\crowner\
   \oldstyle\the\ourreference\hss}
\line{\hbox to8ex{\small Your Ref\hss}:
   \the\yourreference\hss}
}
```

For completeness the following

```
\def\beginscript{\lastscript
   \the\everyscript\the\thisscript
\begingroup\pageno1 \null
\vskip3\bigskipamount
}%end \beginscript
%alias
\let\beginletter\beginscript
%
\def\endscript{\smallskip
  \vfil\eject\endgroup
  \tracingstats1
  \stop\thisscript{}}
%alias
\let\endletter\endscript
%
\def\dear{Dear \the\addresseename,\bigskip}
%
%To be replaced by your salutation
\def\sincerely{{\bigskip
   \parindent\longindentation
   Sincerely,
   \medskip
   \the\author\vskip3\bigskipamount}}
%
\def\ps{\bigskip\small\item{P.S.}}
\def\cc{\bigskip\small\item{cc.}}
\def\appendix#1{\newpage\tenpoint
   \centerline{\bf\the\appendixname\ #1}
   \bigskip}
```

Defaults are

```
\onecol%Because in blue.tex \twocol default
\addressesfile{address}
\searchfile{address}
\letterfile{letter}
\def\email#1{}\def\phone#1{}
%Separation headline and rest
\vsize13cm%First page
\hsize13cm\pagewd\hsize
\hoffset1cm
\parindent0pt
\generalindent2pc
\interlinepenalty1000
\longindentation.667\hsize
\storedvsize\vsize
\storedheadline\headline
\storedfootline\footline
\raggedbottom
```

7.3 The markup for an address label

Address labels can be obtained by `\makelabels` with the names as arguments. Another possibilty is to use `\search` with a pattern as argument — which will yield the address names, each preceded by `\lst`, in `\namelst`, and defines the names with associated list element as replacement text — together with `\makesearchlabels`. The last possibility is to use `\makealllabels`.

```
\def\makelabels#1{\vsize=28cm%
   \headline{}\footline{}%
   \beginaddresses#1\endaddresses
   \let\lst\processlabel\the\namelst}
%
\def\makesearchlabels{\vsize28cm%
   \headline{}\footline{}%
   \let\lst\processlabel\the\namelst}
%
\def\makealllabels{\vsize28cm%
   \headline{}\footline{}%
   \def\lst##1{\processlabel}%
   \input\the\addressesfile.dat\relax
}
%
%with at the lower level
%
\def\processlabel#1{\addressee{#1}%
   \boxit{\kern1cm\vbox to3.5cm
      {\noindent\leftskip.33\hsize
       \hsize.9\hsize
       \unhbox\affiliationbox\vss}
   {\smallskip\small
        \leftskip\generalindent
         \the\author\\
         \the\address\bigskip}
}\smallskip}%end \processlabel%;nonum
```

[10] Note the subtle use of `\unhbox` for the affiliation box.

7.4 Table of contents of database

A stepping stone application of Knuth's list element tag, see *The TEXbook* Appendix D, is provided in the macro below to yield a list of all the names of the entries in a database.

```
\def\contentsdatabase#1{%#1 pic lit address
\ea\let\ea\name\csname toc#1\endcsname
\immediate\openout\name=contents#1
\def\lst##1##2{\immediate\write\name{\nx##1}}
\input #1.dat\relax}
%with auxiliary
\newwrite\toc<#1>
```

Explanation. The list element tag — for these class I databases `\lst` — gets the meaning to extract the `\<name>` of each list entry, and store these in the file `contents<#1>`.

7.5 Search by pattern

The idea is that entries from the database specified in `\addressesfile` can be located via searching for a pattern. Of the found entries the names are collected in the token variable `\namelst`, with each name preceded by `\lst` to facilitate processing later. Moreover the names are defined with the associated list element proper as replacement text, i.e., the spotted entries are loaded.

```
\def\search#1{\def\loc##1##2{%
   \def\locate####1##1####2\end
    {\ifx\empty####2\empty\foundfalse
    \else\foundtrue\fi}\ea\locate##2.##1\end}
   \def\lst##1##2{\loc{#1}{##2}\iffound
     \immediate\write16{\nx##1}%log file
     \namelst\ea{\the\namelst\lst##1}
     \def##1{##2}%define found element
    \fi
   }\input\the\searchfile.dat\relax}
```

8 Test program

The following is in use by me to test the `\letter` format. When you change some parts insert the changed macros after the first line. I assume that a prototype letter has been stored in letter.tex.

```
\input blue \letter
\letterto{\knuthde\grinevaoa}
\bye
\lettertoall
\bye
\makelabels{\knuthde\grinevaoa}
\bye
\makealllabels
\bye
\searchfile{address}
\search{RUSSIA}
\makesearchlabels
\bye
\contentsdatabase{address}
\bye
```

9 Summary of tags

Personalize these tags, make it your crib. The numbers refer to the page numbers in fmt.dat — BLUe's format database of formats — when printed by pgfile.tex, or via the script as provided in the beginning of fmt.dat.

Token variables already available in blue.tex are

- `\vaoffset` and `\haoffset`, to position the address
- `\letterfile`, which contains the name of the file of the letter, and
- `\addressesfile`, which contains the name of the address database.

```
%Address
%  \beginaddresses...........9-19
%  \endaddresses...............21
%  \addresses...............23-24
%Letter
%  \addressee...............26-27
%  \letterto..............101-107
%  \lettertoall...........109-110
%lower level
%  \splitaddress............51-54
%  \processletter.........151-158
%Header and footer
%  \headline..............201-214
%  \footline..............216-224
%lower level
%  \addresseewindow.......251-266
%  \vaoffset, \haoffset...254-254
%Labels
%  \makelabels............301-304
%  \makealllabels.........306-309
%  \makesearchlabels......311-313
%lower level
%  \processlabel..........351-360
%Composition
%  \beginscript...........401-406
%  (\beginletter)............407
%  \endscript.............409-412
%  (\endletter)...............413
%  \dear......................508
%  \sincerely.............502-506
%  \ps........................510
%  \cc........................511
%  \appendix..............512-514
```

10 Acknowledgements

Many a suggestion with respect to simple markup tags in the user interface was done by Erik Frambach. Discussions with Herman Haverkort on the TEX-NL network contributed to a clearer notion of the list element tag. Jos Winnink proofed as usual. Thank you!

11 Conclusion

More or less to complete BLUe's format the `\letter` format emerged. While working on it, it was fun to experience that the database mechanisms as developed for references could be reused in this context.

References

The TEXbook and LATEX user's guide are omni-present and not explicitly listed. For my works consult lit.dat, via the use of `\search` for example.