Mail merge with the use of formlett.sty

Zhuhan JIANG
Department of Mathematics Statistics and Computing Science, University of New England, Armidale N.S.W. 2351, Australia. *Email: zhuhan@neumann.une.edu.au*

(Completed on 12 January 1995)

In this article, the author explains how to use a form-letter or mail merge style `formlett.sty`, designed for plain TEX and LATEX or LATEX2e. `formlett.sty` supports different parameter input methods, parameter naming and defaulting mechanism, as well as facilities for previewing parameter positions and printing labels.It is written for the purpose of being powerful, robust and above all easy to use.

Introduction

Our purpose here is to describe a comprehensive implementation of a macro system, handling form letters or mail merge under plain TEX or LATEX 2.09 or LATEX2e. The main objective is to provide an easy way to output many form letters with their own parameters, with or without the use of multiple files. There will be a coherent and simple format for putting parameters inside a form letter, with a number of helping facilities for such as naming parameters and previewing their positions. A minimum support for printing mailing labels is also provided.

The concept of macros [1] for form letters is not new: there already exist macros in this connection such as **`merge`**, **`textmerg`** and **`address`** to name a few, see [2,3] for further details. Our stress here is therefore laid on the ease to use, along with the power and the robustness of the macros.

Basic format

Suppose we would like to write a form letter for mail merge in the following form

`<<FULL NAME>>`
`<<ADDRESS-01>>`
`<<ADDRESS-02>>`

Dear `<<GIVEN NAME>>`,

We have been looking for `<<MISSING ITEM>>` *for quite a while without any luck, could you help us out? If so, please ring* `<<PHONE NUMBER>>`.

Cheers! *Michael*

where the text inside the double arrow brackets are to be replaced by the parameters specific to each individual letter. If we number these parameters one by one in a single group, then the letter template will be organised as

$\boxed{1-1}$
$\boxed{1-2}$
$\boxed{1-3}$

Dear $\boxed{1-4}$,

We have been looking for $\boxed{1-5}$ *for quite a while without any luck, could you help us out? If so, please ring* $\boxed{1-6}$.

Cheers! *Michael*

and we need just 6 parameters to personalise an individual letter. A simplest example of supplying a set or a *cluster* of these parameters to output a merged form letter is to use

```
\moreletter
  Mrs L Stenson;\#1-20 Sunset Street;%
  Hills, Norway;Louise;a Bible;220-8888!
```

For more letters to be merged this way, just use again the command **`\moreletter`** in the above format, in which each parameter is separated by a semicolon '`;`' and the whole parameter cluster is ended by an exclamation mark '`!`'. The actual code in LATEX for making the above form letter, along with an extra merged letter, may read as follows

```
0001  \documentstyle[formlett]{article}
0002  \begin{document}
0003  \beginletter
0004    \NOPAGENUMBERS\parindent=0pt
0005    \paras[1]\par\paras[2]\par
0006    \paras[3]\par\medskip
0007    Dear \paras[4], \par\medskip
0008    We have been looking for
0009    \paras[5] for quite a while
0010    without any luck, could you help
0011    us out? If so, please ring
0012    \paras[6]. \par\medskip
0013    Cheers!\hfill Michael\vfill\eject
0014  \endletter
0015
0016  \moreletter % letter one
0017    Mrs L Stenson;\#1-20 Sunset Street;%
0018    Hills, Norway;Louise;a Bible;220-8888!
0019  \moreletter % letter two
0020    S Wales;UMIST;Manchester;Sir or Madam;%
0021    a \TeX\ package{+}manual;225-9905!
0022  \end{document}
```

The line numbers in the above and later on are for quick reference only: they are not a part of the program code. The letter template in lines 4-13 is defined by the `\beginletter` and `\endletter` pair in line 3 and line 14, with each use of `\moreletter` generating a new merged letter. We note that command `\paras[`*m*`]` represents the m^{th} merged parameter (of the first group, see later), and the command `\NOPAGENUMBERS` will remove the page numbering for both LaTeX and plain TeX. We also note that if one replaces lines 1-2 by `\input formlett.sty`, then the program will be compilable under plain TeX as well.

Although the mail merge mechanism in the above will be sufficient for most simple applications, `formlett` provides in fact many more features. We shall thus in the followings introduce the main features one by one, from the most useful to the less likely encountered.

To start with, we notice that in the above example there are exactly two parameters to be used for the address. This is inconvenient to say the least, even if we allocate say three more extra parameters for the purpose. This naturally calls for the separation of the whole *cluster* of parameters into several *groups*, so that the numbering of the parameters in every group will be resynchronised. We thus use `\paras[`*m*`][`*n*`]` to denote the m^{th} parameter of the n^{th} group of a whole cluster of merging parameters. In fact we may use `\blockparas[`*m*`][`*n*`][`*pre*`][`*post*`]` to represent parameters in the n^{th} group, from the m^{th} to the last parameter of that group, where the tokens denoted by *pre* and *post* are those to be added in front and behind respectively each of the legitimate parameters. The defaults for *pre* and *post* are `\noindent` and `\par` respectively. It is therefore more sensible to specify the merging parameters of the earlier example via

`<<FULL NAME>>`
`<<ADDRESS-etc>>`

Dear `<<GIVEN NAME>>`*,*

We have been looking for `<<MISSING ITEM>>` *for quite a while without any luck, could you help us out? If so, please ring* `<<PHONE NUMBER>>`*.*

Cheers! *Michael*

where `<<ADDRESS-etc>>` will be produced by `\blockparas[2][1]`, i.e. all the parameters of the first group, starting from the 2nd parameter. We note that the missing square-bracketed macro parameters for `\blockparas` are automatically provided: in fact macro parameters in `formlett` are systematically defaulted. Hence for example, `\paras[1][1]` is equivalent to any of the following commands

`\paras`, `\paras[1]`, `\paras[][1]`,
`\paras[1][]`, `\paras[][]`

As for the more precise position of the merging parameters, they can always be viewed via the command `\preview`, which in this case gives the following positioning

$\boxed{\text{1 – 1}}$

$\boxed{\text{1 – 2 (+): macro:->\textbackslash noindent <<paras>> macro:->\textbackslash par}}$

Dear $\boxed{\text{2 – 1}}$,

We have been looking for $\boxed{\text{2 – 2}}$ *for quite a while without any luck, could you help us out? If so, please ring* $\boxed{\text{2 – 3}}$.

Cheers! *Michael*

This way, the mail merge for the above letter template can also be done (in plain TeX) in the following more flexible way

```
\input formlett.sty
\beginletter
\NOPAGENUMBERS\parindent=0pt
\noindent{\it\paras[1]}\par
\blockparas[2]\par\bigskip
Dear \paras[1][2],\par\medskip
We have been looking for
\paras[2][2] for quite a while
without any luck, could you help
us out? If so, please ring
\paras[3][2]. \par\medskip
Cheers!\hfill Michael\vfill\eject
\endletter \preview

\beginpilemode
  Mrs L Stenson;\#1-20 Sunset Street
  Hills, Norway+Louise;a Bible;220-8888!

  S Wales;UMIST;Manchester+%
  Sir or Madam;a \TeX\space
  package{+}manual;225-9905!
\endpilemode
\end
```

We note that by default we used plus sign '+' to separate different groups inside a cluster of parameters, and each cluster inside a `\beginpilemode` and `\endpilemode` pair will output a merged letter, as if it were produced by command `\moreletter` followed by that cluster of parameters. The output of the first merged letter in fact reads

Mrs L Stenson
#1-20 Sunset Street Hills, Norway

Dear Louise,

We have been looking for a Bible for quite a while without any luck, could you help us out? If so, please ring 220-8888.

Cheers! *Michael*

It is often desirable to supply parameters line by line, i.e. one line as one parameter. Hence the two merged letters produced by lines 25-47 can also be made by replacing the content included in the `\beginpilemode` and `\endpilemode` pair (lines 39-46) by the following new code

```
0048 \blockmarks
0049 \beginblockmode
0050 Mrs L Stenson
0051 \#1-20 Sunset Street
0052 Hills, Norway
0053 -----
0054 Louise
0055 a Bible
0056 220-8888
0057 ======
0058
0059 S Wales
0060 UMIST
0061 Manchester
0062 ----
0063 Sir or Madam
0064 a \TeX\ package+manual
0065 225-9905
0066 ====
0067 \endblockmode
0068 \defaultmarks
```

where obviously we have used lines with '`----`' to separate groups and used lines with '`====`' to end a cluster of parameters. This was done by the use of `\blockmarks` which changes the group delimiter '`+`' and cluster delimiter '`!`' into the above two new ones, with the `\defaultmarks` at the end setting them back. If the whole cluster of parameters are categorised into a single (first) group, then we may even use a block of consecutive nonempty lines to provide the merging parameters. This is done by the use of command pair `\beginlinemode` and `\endlinemode`. Hence we may also conduct the above mail merge in for instance LaTeX2e by

```
0069 \documentclass{article}
0070 \usepackage{formlett}
0071 \begin{document}
0072   \beginletter
0073     \NOPAGENUMBERS\parindent=0pt
0074     \paras[4]\par\blockparas[5]
0075     \par\medskip
0076     Dear \paras[1], \par\medskip
0077     We have been looking for
0078     \paras[2] for quite a while
0079     without any luck, could you help
0080     us out? If so, please ring
0081     \paras[3]. \par\medskip
0082     Cheers!\hfill Michael\vfill\eject
0083   \endletter
0084
0085 \beginlinemode
0086 Louise
0087 a Bible
0088 220-8888
0089 Mrs L Stenson
0090 \#1-20 Sunset Street
0091 Hills, Norway
0092
0093 Sir or Madam
0094 a \TeX\ package+manual
0095 225-9905
0096 S Wales
0097 UMIST
0098 Manchester
0099 \endlinemode
0100 \end{document}
```

Notice that we have changed the numbering order of the parameters so that the number of address lines output by `\blockparas` is more flexible.

Advanced features

The advanced features to be discussed below are mainly for the purpose of (i) writing *long* form letter; (ii) making *many* different form letters; (iii) reading mail merge parameters that are in *crude* ASCII form; and (iv) generating mail labels.

It is in general better to keep the letter template, i.e. the content between `\beginletter` and `\endletter` like those in lines 4-13, in a separate file, say `myletter.let`, and use `\inputletter{myletter.let}` to load in the letter template. The advantage of this over the previously used method is that the letter template can now be arbitrarily large without any risks of causing computer memory problem.

Suppose we have a collection of letter templates, we may wish to remind ourselves what each parameter represents. For this purpose, we may add at the top of the letter template a command `\paranames` *cluster*`!`, where *cluster* is a cluster of parameters each giving a name to the corresponding parameter. Command `\paranames` takes its macro parameters in the same way `\moreletter` does. After a letter template is read in, command `\showparas` will output a form letter with its parameters replaced by the corresponding names. The mainly italic passage containing lines 23-24, for instance, is the output of `\showparas` for the letter template in lines 102-109 given explicitly later on.

We may also use `\paradefaults` *cluster*`!` to provide default parameters, i.e. the parameters to replace the empty or not entered ones, and use `\loaddefaultparas` inside letter template to activate the parameter defaulting. We note that in general commands `\paranames` and `\loaddefaultparas`, if present, should be at the top of the letter template, with `\loaddefaultparas` below the command `\paranames`. Hence for our previous mail merge, we may re-do it via

```
0101   \input formlett.sty
0102   \beginletter
0103   \paranames              % optional
0104     \tt<<FULL NAME>>;\tt<<ADDRESS-etc>>;%
0105    +\tt<<GIVEN NAME>>;\tt<<MISSING ITEM>>;%
0106     \tt<<PHONE NUMBER>>!
0107   \loaddefaultparas       % optional
0108    {letter details given in lines 27-36}
0109   \endletter
0110   \preview \showparas
0111
0112   \paradefaults        % optional
0113     To whom this may concern
0114    +Sir or Madam;something;%
0115     061-225-9905!
0116
0117 \blockmarks
0118 \beginrawblockmode{}
0119 Mrs L Stenson
0120 #1-20 Sunset Street
0121 Hills, Norway
0122 ------
0123 Louise
0124 a Bible
0125 220-8888
0126 =========
0127
0128 ......
0129
0130 Above empty line active
0131 \endrawblockmode
0132 \defaultmarks
0133 \end{document}
```

We note that in *blockmode* with `\blockmarks`, a line with '`....`' marks the immediate start of a parameter block, whether or not the lines immediately following it are empty or not. Also that in *rawblockmode* enclosed by `\beginrawblockmode` and `\endrawblockmode`, all characters are input in their original ASCII form. In other words, the special characters for TeX will be ignored here.

If the merging parameters are given by exactly m lines per cluster, as is often the case when they are produced by a database utility, then we may use the `\begindatamode` and `\enddatamode` pair to mark the beginning and the end of the merging parameters. For more details, see the Appendix at the end.

We may keep the letter template and the merging parameters in two separate files. For instance, if we save lines 103-108 to file `myletter.let`, lines 39-46 or lines 48-68 to file `myletter.adr`, then we can produce via LaTeX the mail merge by

```
0134   \documentstyle[formlett]{article}
0135   \begin{document}
0136   \inputletter{myletter.let}
0137   \showparas \preview  % utility demo
0138   \paradefaults To whom it may concern!
0139   \inputfile{myletter.adr}
0140   \beginlabels % 1st group as address
0141   \inputfile{myletter.adr} % for labels
0142   \endlabels
0143   \end{document}
```

where the lines 140-142 will generate the address labels using the first group (as is intended here) of parameters. More precise format of `\beginlabels` and etc can be found in the Appendix.

Another way of making labels, if no separate files for the parameters are present, is to re-read the document itself and make labels instead of merged form letters in the second reading. We may thus use `\input formlett.sty \initstyle` `[`*styles*`]{article}{`*preamble*`}` to replace `\documentstyle [formlett,` *styles*`]{article}` *preamble* `\begin{document}`, which will be valid for LaTeX but ignored for TeX , and will enable one to use `\labelsquit` at the end to read in the current document again with all the letters there converted into the corresponding labels. Though `\initstyle` is valid for plain TeX, LaTeX2.09 as well as LaTeX2e, `\initclass` will generate `\documentclass` instead of `\documentstyle` when LaTeX2e is the processing environment. If you only want to execute certain commands the first time round (i.e. before `\labelsquit` re-reads the file again), use `\firstread{`*commands*`}` for this purpose. For more detailed use and examples, read the *programmer's document* attached to the style file `formlett.sty` and the example files it generates. In fact, many details and extra features that are not contained in this article may be found there.

We already noted that if a form letter is large, we have to use `\inputletter` to load in the template. However, it is often still desirable to keep everything inside a single file, while allowing it to make new (scratch) files when it is being compiled. We may thus use for example

```
\beginfile{myletter.let}
{letter template of lines 103-108}
\endfile
```

to create a letter template file `myletter.let`. Likewise, the merging parameters can be saved to another run-time created file `myletter.adr`.

We have so far only used the default delimiters '`;`', '`+`' and '`!`' to separate single, group and cluster of parameters, with temporary change via `\blockmarks` to '`....`', '`----`' and '`====`' respectively. However we may change the delimiters to any characters by `\delimiters`$\{S\}\{G\}\{C\}$, where S, G and C are the tokens to delimit single, group and cluster of parameters respectively, and change the delimiters back to the default by `\defaultmarks` or equivalently by `\delimiters{;}{+}{!}`.

To conclude this section, let us finally look below at an 'ideal' and 'finished' mail merge application code,. It is currently in a form compilable under LaTeX2e, and is also valid for plain TeX if the first three lines (lines 144-146) are removed.

```
0144 \documentclass{article}
0145 \usepackage{formlett}  % LaTeX2e
0146 \begin{document}
0147
0148 %%%%  MAKE file scr@tch@.let
0149 \beginfile{scr@tch@.let} % letter content
0150 \paranames <<name>>;<<address>>+<<items>>!
0151 \loaddefaultparas
0152 \parindent=0pt\blockparas\bigskip\bigskip
0153 Dear \paras, \par\bigskip
0154 This part is typically the letter content.
0155 It now displays all the items in the 2nd
0156 parameter group. They are \par
0157 \blockparas[][2] \vfill\eject
0158 \endfile
0159
0160 %%%%  MAKE file scr@tch@.adr
0161 \beginfile{scr@tch@.adr} % address file
0162 \blockmarks
0163 \beginblockmode
0164 T Teng
0165 UMIST
0166 Manchester M60 1QD
0167 =====
0168
0169 Z Jiang
0170 UNE
0171 Armidale NSW 2351
0172 Australia
0173 ------
0174 Email: zhuhan@neumann.une.edu.au
0175 =====
0176 \endblockmode
0177 \defaultmarks
0178 \endfile
0179
0180 % MAIN BODY
0181 \paradefaults To Whom It May Concern
0182   +No further info available{!}!
0183 \inputletter{scr@tch@.let}
0184 \inputfile{scr@tch@.adr}% for letters
0185 \beginlabels
0186 \inputfile{scr@tch@.adr}% for labels
0187 \endlabels
0188 \end{document}
```

We note that if we replace lines 144-146 by `\input formlett.sty \initstyle{}{}`, then the mail merge can be processed by either plain TeX or LaTeX (including LaTeX2e). It is likewise if `\initstyle` in the replaced line is replaced furthermore by `\initclass`, except that now the native LaTeX2e environment is preserved, when applicable, rather than turning it into the compatibility mode of LaTeX2.09.

Programmer's tips

First we note that all the macro parameters that are to appear inside square brackets of `formlett`'s commands are defaulted, just like how `\paras[`m`][`n`]` is defaulted earlier on. The actual default values can be found in the Appendix.

For those wizard users who want to do everything their own way, we point out that if for instance the 3rd letter parameter of the 2nd group of a cluster is given as `<A>`, then `\LET2*3~` will contain `\b@group\relax<A>\e@group` right after a cluster is read in. `\DEF2*3~`, on the other hand, contains the corresponding default parameter in the same fashion. Furthermore, the command `\checkparas[`m`][`n`]{LET}` will copy the content of `\paras[`m`][`n`]`, minus the 'wrapping' extra tokens `\b@group\relax` and `\e@group`, to `\cachedata` and set `\ifemptyparas` to true or false depending on whether the content is empty or not. This way, a user may even change the characteristics of his letter template by first testing the content of the supplied individual parameters. However, we note that if `\loaddefaultparas` is executed, then the `LET` array, when some of its elements are not supplied, will contain the corresponding elements of the `DEF` array. Hence care must be exercised under such circumstances, when interpreting the `\cachedata` generated by `\checkparas[`m`][`n`]{LET}`. If necessary, we may use `\delparadefaults` to delete current default parameter array `DEF` so as to conduct `\checkparas{LET}` more precisely.

There are also a number of generic macro utilities in `formlett`, including a user stack and a multidimensional array mechanism.

`formlett` was written for all TeX dialects, its format is thus more close to that of plain TeX. In fact, we deliberately avoided mixing up `formlett` with the standard letter environment of LaTeX. This is largely due to the fact that the limited facilities of LaTeX letter environment are easy to come by anyway, and are not really worth writing buckled code to make `formlett` a type of extension of the environment. Nevertheless, one can still use the letter environment of LaTeX inside the form letter template. Besides, the applicability of `formlett` is not restricted for making form *letters*; it can also be used to merge other type of articles or passages.

References

[1]. Knuth D E, *The TeXbook*, Reading, Mass., Addison-Wesley, 1992.

[2]. Piff M, Text merges in TEX and LATEX, *TUGboad*, 13(4):518, 1993.

[3]. Damrau J and Wester M, Form letters with 3-across labels capacity, *TUGboat*, 13(4):510, 1991.

Appendix

In the followings, we give a brief summary of the main commands given by `formlett` version 2.1.

Let m and n be numbers, p, q and r be dimensions, A, B, S, G, C and T be tokens, and X be a box. Furthermore, we shall denote by R a full set of letter parameters ended by '`!`', with '`;`' separating parameters inside a same group and '`+`' separating different parameter groups. We moreover denote R by F, when '`; + !`' there can be replaced by 'S G C' respectively if `\delimiters`$\{S\}\{G\}\{C\}$ is issued. In the commands tabulated below, the macro parameters contained in squared brackets support default. In particular, the defaults are m=`1`, n=`1`, p=`8truecm`, q=`1.5em`, r=`3pt`, A=`\noindent` and B=`\par`.

`\paras`$[m][n]$	m^{th} parameter of n^{th} group
`\blockparas`$[m][n][A][B]$	m^{th} to the last parameter of n^{th} group, each preceded by A and followed by B, wrapped by {} if B=`\relax`
`\addressparas`$[m][n][p][q]$	m^{th} to the last parameter of n^{th} group, each put into a box of width p with indent q for wrapped portions
`\loaddefaultparas`	fill empty parameters with defaults
`\checkparas`$[m][n]\{T\}$	m^{th} parameter of n^{th} group copied to `\cachedata`; `\ifemptyparas` is true if element is empty; T is often `LET` or `DEF`
`\moreletter` F	use parameters F to output a new letter
`\paranames` R	use R as parameter names
`\paradefaults` R	use R as default parameters
`\delparadefaults`	delete default parameters

`\delimiters`$\{S\}\{G\}\{C\}$
`\defaultmarks`
`\blockmarks`
	use S, G, C as delimiters
	use '`; + !`' as delimiters
	use '`....`', '`----`', '`====`' as delimiters

`\preview`	highlight parameter positions
`\showparas`	display parameter names, if any
`\beginfile`$[T]\{$*file.ext*$\}$ `\endfile`	write text verbatim to file *file.ext* (empty implies `scr@tch@.tex`), nonempty T replaces `\endfile` to mark last full line
`\inputletter`$\{$*file.ext*$\}$ `\inputfile`$\{$*file.ext*$\}$	input letter content input *file.ext*

`\beginletter` `\endletter`	delimiters for letter content (template)
`\beginpilemode` `\endpilemode`	normal letter parameters cluster-wise
`\beginblockmode` `\endblockmode`	for line-by-line blocks of parameters, empty lines active within each cluster
`\beginlinemode` `\endlinemode`	for line-by-line parameters, empty lines delimit clusters
`\beginrawblockmode`$\{T\}$ `\endrawblockmode`	raw text mode; nonempty T replaces `\endrawblockmode` to mark end
`\beginrawlinemode`$\{T\}$ `\endrawlinemode`	raw text parameters and active spaces etc
`\begindatamode`$[T]\{m\}$ `\enddatamode`	m raw text lines for one form letter

`\PAGENO=1`
`\NOPAGENUMBERS`
`\textbox`$[p]\{text\}$
`\boxmore`$[r]\{X\}$
	page number reset to 1
	no page numbers
	text into box of width p
	add borderline to box X at a distance r

`\addressbox`$[p][q]\{text\}$	*text* into box of width p, with wrapped options indented by q
`\beginlabels`$[a][b][c][d][e][f]$ `\endlabels` *defaults:* a=`20pt`, b=`\tt\raggedright`, c=`1`, d=`1`, e=`2.6in`, f=`2em`	form letters become labels: address taken from c^{th} to last parameter of d^{th} group, with width e, indent f, borderspace a and font toks b
`\firstread`$\{T\}$	toks T will not be read if the file is re-read via `\labelsquit`
`\initstyle`$[a]\{b\}\{c\}$	initiation for `\labelsquit`,with styles a, documentstyle b and preamble c
`\initclass`$[a][o]\{b\}\{c\}$	similar to `\initstyle` (o is LATEX2e options), but retains native LATEX2e when applicable
`\labelsquit`$[a][b][c][d][e][f]\{$*file.ext*$\}$ *defaults:* see that for `\beginlabels`	quit after converting letters to labels by reading the current document or *file.ext*