# The dlfltxbmarkup package*

Lars Madsen†

March 4, 2010

This package implements the `\markup` features that are used extensively in Madsen (2010).

## Requirements

This package depends on several *memoir* internals so it can only be used together with the *memoir* class. Further requirements are `keyval` and `ragged2e`.

## Usage

The `\markup` macro is intended to be used for categorising various types of keywords. These can then be automatically written in the text as well as in the (outer) margin and added to the index.

`\markup` `\markup[⟨keys⟩]{⟨text⟩}`

The key list is a mix of control keys and user defined category keys. How to created the user defined keys are explained later. The control keys are listed below.

`notxt`
: do not write anything in the text

`nowr` alias for `notxt`

`nomk` do not write anything in the margin

`noidx`
: do not add anything to the index

`idxit`
: index entry in italic (i.e. page number in italic)

`idxbf`
: index entry in bold face

`idxn` normal index entry

`addtospvrt=`⟨*number*⟩
: to be used in situations where we use `\sidepar` instead of `\marginpar`. This key will move the text further down in the margin. It is a unit-less number (`\onelineskip` is the unit).

`vaddtosp=`⟨*number*⟩
: alias for `addtospvrt`

`forcesidepar`
: forces the use of `\sidepar` instead of `\marginpar`

*Version: 0.6

†Email: daleif@imf.au.dk

The package has one option »loadsampleconfig« which will auto load the »dlfltxbmarkup-sample.cfg« sample file (which actually contains the categories that I use for the creation of Madsen (2010)).

### Defining your own categories

The main macro for defining your own category keys are

\felineKeyGenerator

\felineKeyGenerator{⟨*key*⟩}{⟨*description*⟩}{⟨*margin code*⟩}{⟨*index code*⟩}{⟨*text code*⟩}

In each of the last three arguments »#1« will refer to the mandatory argument given to \markup. The reason for adding the ⟨*description*⟩ part will be explained later. For the code to be used in the three last arguments we have a few helper macros that might be useful

\cs

\cs{⟨*text*⟩}

for writing the name of LaTeX macros.

\css

\css{⟨*text*⟩}

like \cs but to be used in the index, it will typeset the »\« in the left margin.

\felineWriteInMargin

\felineWriteInMargin{⟨*text*⟩}

\NoMarginparAvail

Depending on whether \NoMarginparAvail is true or not, it will use \sidepar (when \marginpar is unavailable) and \marginpar otherwise (unless of course the »forcesidepar« key is in effect). A \marginpar can for example not be used from inside a ***framed*** or ***shaded*** environment. So in such an environment one can just make sure that \NoMarginparAvail is set to true and then we will still have some thing in the margin. It is recommended to use

\strictpagechecktrue

\strictpagechecktrue

in order to get more rigorous side checking. Internally we use \checkoddpage and \ifodd to test whether we are in the left or right margin. We also use

\felineMarginAdjustment

\felineMarginAdjustment

to ensure that the margin text has the correct adjustment. The default setting is

```
\newcommand\felineMarginAdjustment{%
  \ifoddpage\RaggedRight\else\RaggedLeft\fi}
```

In this documentation which is not twosided, it has been redefined to be \RaggedLeft.

For index entries we recommend the use of

\felineIndexCmd

\felineIndexCmd

which initially is defined to be \index. The »idxit« control key redefines it to be \itindex, which is

```
\newcommand\itindex[1]{\index{#1|textitindex}}
```

where `\textitindex` is just `\textit`. Similar for »`idxbf`«.

Let us now create a category key for LaTeX packages.

```
\felineKeyGenerator{sty}%
  {for registering package names}%
  {\felineWriteInMargin{\foreignlanguage{english}{#1}}}%
  {\felineIndexCmd{#1 (package)@#1 (package)}%
    \felineIndexCmd{packages!#1}}%
  {\foreignlanguage{english}{\texttt{\hyphenchar\font=‘\-#1}}}
```

The rather odd last line is to ensure that the package name is written under English hyphenation rules, and the »`\hyphenchar\font=‘\-`« is to ensure that we can get hyphenation in the typewriter font. As you can see we create two index entries, one for the name it self (marking it as the name of a package) and one as a sub item to a list of mentioned packages.

Here is another key for mathematical symbols

```
\def\felinenameuse#1{\@nameuse{#1}}
\felineKeyGenerator{msym}%
  {mathematical symbols}
  {\felineWriteInMargin{\cs{#1} \textnormal{($\@nameuse{#1}$)}}}%
  {\felineIndexCmd{#1@\protect\css{#1} ($\protect\felinenameuse{#1}$)}}%
  {\cs{#1}}
```

`\@nameuse` is used to transform the name of the macro into the macro it self (for those that didn't already know `\@nameuse`). The shortcut `\felinenameuse` is used here because `\@nameuse` might not be permissible in the index (because of the »`@`«.

You can place these keys either in your preamble or in »`dlfltxbmarkup.cfg`« which will be auto loaded if found. If you have used the package option »`loadsampleconfig`« then »`dlfltxbmarkup-sample.cfg`« will be loaded at the very end of the package, i.e. *after* »`dlfltxbmarkup-sample.cfg`«.

The default category key to be used by `\markup` is controlled by

`\felineStandardKey`

```
\felineStandardKey
```

Just redefine it to be the key you want. Notice that by using »`loadsampleconfig`« the default key is set to »`macro`«.

### The descriptions

For a long project like Madsen (2010) the list of category keys may grow quite big, and one might loose track of the keys and what they should be used for. Under normal use of the this package, the description is ignored. But internally in `\felineKeyGenerator` we call the macro `\felineMarkupDescription` with all five arguments. By default it does nothing. But the user can define it to do something before loading the package.

`\felineMarkupDescription`

Located with this manual you should find the file »dlfltxbmarkup-showkeys.tex« which will print a new document showing they keys and their description. If you move a % in that file you will get a list of the keys used for Madsen (2010).

## Bibliography

Lars Madsen. Introduktion til LaTeX. http://www.imf.au.dk/system/latex/bog/, 2010. The current version of the book is 3rd edition beta.