# Contents

1 Introduction  
2 Capitalising the First Letter of a Word  
3 Capitalise the First Letter of Each Word in a Phrase or Sentence (Title Case)  
   3.1 PDF Bookmarks  
   3.2 Excluding Words From Case-Changing  
4 UTF-8  
Index
1 Introduction

The mfirstuc package was originally part of the glossaries bundle for use with commands like \Gls, but as the commands provided by mfirstuc may be used without glossaries, the two have been split into separately maintained packages.

The commands described here all have limitations. To minimise problems, use text-block style semantic commands with one argument (the text that requires case-changing), and avoid scoped declarations.

Here are some examples of semantic commands:

1. Quoted material:
   \newcommand{\qt}[1]{``#1''}
   (or use the cquotes package). With this, the following works:
   \makefirstuc{\qt{word}}
   This produces:
   “Word”
   Whereas
   \makefirstuc{`word''}
   fails (no case-change and double open quote becomes two single open quotes):
   “word”

2. Font styles or colours:
   \newcommand*{\alert}[1]{\textcolor{red}{#1}}
   Now the following is possible:
   \makefirstuc{\alert{word}}
This produces

Word

Note that \makefirstuc{\textcolor{red}{word}} fails (with an error) because the case-changing interferes with the label.

Define these semantic commands robustly if you intend using any of the commands that fully expand their argument (\emakefirstuc, \ecapitalisewords and \ecapitalisefmtwords).
2 Capitalising the First Letter of a Word

A simple word can be capitalised just using the standard \LaTeX upper casing command. For example,

\MakeUppercase word

but for commands like \Gls the word may be embedded within the argument of another command, such as a font changing command. This makes things more complicated for a general purpose solution, so the \texttt{mfirstuc} package provides:

\begin{verbatim}
\texttt{\makefirstuc{〈stuff〉}}
\end{verbatim}

This makes the first object of \texttt{〈stuff〉} upper case unless \texttt{〈stuff〉} starts with a control sequence followed by a non-empty group, in which case the first object in the group is converted to upper case. **No expansion is performed on the argument.**

If \texttt{〈stuff〉} starts with a control sequence that takes more than one argument, the case-changing will always be applied to the *first* argument. If the text that requires the case change is in one of the other arguments, you must hide the earlier arguments in a wrapper command. For example, instead of \texttt{\textcolor{red}{text}} you need to define, say, \texttt{\red} as \texttt{\color{red}} and use \texttt{\red{text}}.

Examples:

• \texttt{\makefirstuc{abc}} produces Abc.

• \texttt{\makefirstuc{\emph{abc}}} produces Abc (\MakeUppercase has been applied to the letter “a” rather than \emph). Note however that

\begin{verbatim}
\makefirstuc{〈〈em  abc〉〉}
\end{verbatim}

produces ABC (first object is \texttt{〈\em  abc〉} so this is equivalent to \texttt{\MakeUppercase{〈\em  abc〉}}, and

\begin{verbatim}
{\makefirstuc{〈em  abc〉}}
\end{verbatim}

produces abc (\em doesn’t have an argument therefore first object is \em and so is equivalent to \texttt{{\MakeUppercase{〈em〉}abc}}).
• \makefirstuc{"\'a}bc} produces Ábc.
• \makefirstuc\ae\ bc} produces Æbc.
• \makefirstuc\{\ae}bc} produces Æbc.
• \makefirstuc\{\a}bc} produces Âbc.

Note that non-Latin or accented characters appearing at the start of the text should be placed in a group (even if you are using the inputenc package). The reason for this restriction is detailed in Section 4.

New to version 2.04: There is now limited support for UTF-8 characters with the inputenc package, provided that you load datatool-base (at least v2.24) before mfirstuc (datatool-base is loaded automatically with newer versions of glossaries). If available mfirstuc will now use datatool-base's \dtl@getfirst@UTFviii command which is still experimental. See the datatool manual for further details.

\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage{datatool-base}[2016/01/12]% v2.24+
\usepackage{mfirstuc}
\begin{document}
\makefirstuc{élite}
\end{document}

(Package ordering is important.)

In version 1.02 of mfirstuc, a bug fix resulted in a change in output if the first object is a control sequence followed by an empty group. Prior to version 1.02, \makefirstuc\{\ae{}bc} produced æBc. However as from version 1.02, it now produces Æbc.

Note also that
\newcommand{\abc}{abc}
\makefirstuc{\abc}

produces: ABC. This is because the first object in the argument of \makefirstuc is \abc, so it does \MakeUppercase{\abc}. Whereas:
\newcommand{\abc}{abc}
\expandafter\makefirstuc\expandafter{\abc}

6
produces: Abc. There is a short cut command which will do this:
\makefirstuc{(stuff)}

This is equivalent to \expandafter\makefirstuc\expandafter{(stuff)}. So
\newcommand{\abc}{abc}
\makefirstuc{\abc}

produces: Abc.

\makefirstuc only performs one level expansion on the first object in its argument. It does not fully expand the entire argument.

As from version 1.10, there is now a command that fully expands the entire argument before applying \makefirstuc:
\emakefirstuc{(text)}

Examples:
\newcommand{\abc}{\xyz a}
\newcommand{\xyz}{xyz}
No expansion: \makefirstuc{\abc}.
First object one-level expansion: \xmakefirstuc{\abc}.
Fully expanded: \emakefirstuc{\abc}.


If you use mfirstuc without the glossaries package, the standard \MakeUppercase command is used. If used with glossaries, \MakeTextUppercase (defined by the textcase package) is used instead. If you are using mfirstuc without the glossaries package and want to use \MakeTextUppercase instead, you can redefine
\glsmakefirstuc{(text)}

For example:
\renewcommand{\glsmakefirstuc}[1]{\MakeTextUppercase #1}

Remember to also load textcase (glossaries loads this automatically).
3 Capitalise the First Letter of Each Word in a Phrase or Sentence (Title Case)

New to mfirstuc v1.06:

\capitalisewords{\text{text}}

This command applies \makefirstuc to each word in \text{text} where the space character is used as the word separator. Note that it has to be a plain space character, not another form of space, such as ~ or \space. Note that no expansion is performed on \text{text}. See Section 3.2 for excluding words (such as "of") from the case-changing.

The actual capitalisation of each word is done using (new to version 2.03):

\MFUcapword{\text{word}}

This just does \makefirstuc{\text{word}} by default, but its behaviour is determined by the conditional:

\ifMFUhyphen{\text{word}}

If you want to title case each part of a compound word containing hyphens, you can enable this using

\MFUhyphentrue

or switch it back off again using:

\MFUhyphenfalse

Compare

\capitalisewords{server-side includes}

which produces:

Server-side Includes

with
which produces:

Server-Side Includes

Formatting for the entire phrase must go outside \textbf{\texttt{server-side includes}} (unlike \texttt{\textbf{makefirstuc}}). Compare:

which produces:

A sample phrase

with:

which produces:

A Sample Phrase

As from version 2.03, there is now a command for phrases that may include a formatting command:

\texttt{\textbf{\texttt{capitalisefmtwords{phrase}}}}

where \texttt{(phrase)} may be just words (as with \texttt{\textbf{capitalisewords}}) or may be entirely enclosed in a formatting command in the form

or contain formatted sub-phrases

Avoid scoped declarations.

\texttt{\textbf{\texttt{capitalisefmtwords}}} is only designed for phrases that contain text-block commands with a single argument, which should be a word or sub-phrase. Anything more complicated is likely to break. Instead, use the starred form or \texttt{\textbf{capitalisewords}}.

The starred form only permits a text-block command at the start of the phrase. Examples:
1. Phrase entirely enclosed in a formatting command:

\textbf{a small book of rhyme}

\textbf{A Small Book Of Rhyme}

2. Sub-phrase enclosed in a formatting command:

\textbf{a small book} of rhyme

A Small Book Of Rhyme

3. Nested text-block commands:

\textbf{a} \emph{small book} of rhyme

A Small Book Of Rhyme

4. Indicating words that shouldn't have the case changed (see Section 3.2):

\textbf{a} \emph{small book} of rhyme

A Small Book of Rhyme

5. Starred form:

\textbf{a} \emph{small book of rhyme}

A Small Book of Rhyme
produces:

*A Small Book of Rhyme*

6. The starred form also works with just text (no text-block command):

```
\MFUnocap{of}
capitalisefmtwords*{a small book of rhyme}
```

produces:

*A Small Book of Rhyme*

If there is a text-block command within the argument of the starred form, it’s assumed to be at the start of the argument. Unexpected results can occur if there are other commands. For example

```
\MFUnocap{of}
capitalisefmtwords*{\emph{a small} book \textbf{of rhyme}}
```

produces:

*A Small Book Of rhyme*

(In this case \textbf{of rhyme} is considered a single word.) Similarly if the text-block command occurs in the middle of the argument:

```
\MFUnocap{of}
capitalisefmtwords*{a \emph{very small} book of rhyme}
```

produces:

*A Very small Book of Rhyme*

(In this case \emph{very small} is considered a single word.) Grouping causes interference:

```
capitalisefmtwords*{a \emph{small book}} of rhyme
```

produces:

*A Small book Of Rhyme*

As with all the commands described here, avoid declarations. For example, the following fails:

```
capitalisefmtwords*{\bfseries a \emph{small book}} of rhyme
```
produces:

\textit{a Small book Of Rhyme}

Avoid complicated commands in the unstarred version. For example, the following breaks:

\begin{verbatim}
\newcommand*{\swap}[2]{\{#2\}{#1}}
\capitalisefmtwords{a \swap{bo}{ok} of rhyme}
\end{verbatim}

However it works okay with the starred form and the simpler \texttt{\capitalisewords}:

\begin{verbatim}
\newcommand*{\swap}[2]{\{#2\}{#1}}
\capitalisefmtwords*{a \swap{bo}{ok} of rhyme}
\end{verbatim}

\begin{verbatim}
\capitalisewords{a \swap{bo}{ok} of rhyme}
\end{verbatim}

Produces:

\texttt{A okBo Of Rhyme}
\texttt{A okBo Of Rhyme}

Note that the case change is applied to the first argument.

This is a short cut for \texttt{\expandafter\capitalisewords{\text}}. As from version 1.10, there is now a command that fully expands the entire argument before applying \texttt{\capitalisewords}:

\begin{verbatim}
\ecapitalisewords{\text}
\end{verbatim}

There are also similar shortcut commands for the version that allows text-block commands:

\begin{verbatim}
\xcapitalisefmtwords{\text}
\end{verbatim}

The unstarred version is a short cut for \texttt{\expandafter\capitalisefmtwords\expandafter{\text}}. Similarly the starred version of \texttt{\xcapitalisefmtwords} uses the starred version of \texttt{\capitalisefmtwords}.

For full expansion:

\begin{verbatim}
\ecapitalisefmtwords{\text}
\end{verbatim}

Take care with this as it may expand non-robust semantic commands to replacement text that breaks the functioning of \texttt{\capitalisefmtwords}. Use robust seman-
tic commands where possible. Again this has a starred version that uses the starred form of \capitalisefmtwords.

Examples:

\newcommand{\abc}{\xyz\space four five}
\newcommand{\xyz}{one two three}

No expansion: \capitalisewords{\abc}.
First object one-level expansion: \xcapitalisewords{\abc}.
Fully expanded: \ecapitalisewords{\abc}.

produces:

No expansion: ONE TWO THREE FOUR FIVE.
First object one-level expansion: ONE TWO THREE four Five.
Fully expanded: One Two Three Four Five.

(Remember that the spaces need to be explicit. In the second case above, using \xcapitalisewords, the space before “four” has been hidden within \space so it’s not recognised as a word boundary, but in the third case, \space has been expanded to an actual space character.)

Examples:

1. \capitalisewords{a book of rhyme.}
   produces: A Book Of Rhyme.

2. \capitalisewords{a book\space of rhyme.}
   produces: A Book of Rhyme.

3. \newcommand{\mytitle}{a book\space of rhyme.}
   \capitalisewords{\mytitle}
   produces: A BOOK OF RHYME. (No expansion is performed on \mytitle.)
   Compare with next example:

4. \newcommand{\mytitle}{a book\space of rhyme.}
   \xcapitalisewords{\mytitle}
   produces: A Book of Rhyme.
   However
   \ecapitalisewords{\mytitle}
   produces: A Book Of Rhyme. (\space has been expanded to an actual space character.)

5. \newcommand*{\swap}[2]{\{#2\}\{#1\}}
   \capitalisewords{a \swap{bo}{ok} of rhyme}
\ecapitalisewords{a \swap{bo}{ok} of rhyme}
produces:

A okBo Of Rhyme
A OKbo Of Rhyme

This is because the argument of \ecapitalisewords is fully expanded before being passed to \capitalisewords so that last example is equivalent to:

\capitalisewords{a \{ok\}\{bo\} of rhyme}

### 3.1 PDF Bookmarks

If you are using hyperref and want to use \capitalisewords, \capitalisefmtwords or \makefirstuc (or the expanded variants) in a section heading, the PDF bookmarks won't be able to use the command as it's not expandable, so you will get a warning that looks like:

Package hyperref Warning: Token not allowed in a PDF string (PDFDocEncoding):
(hyperref) removing `\capitalisewords'

If you want to provide an alternative for the PDF bookmark, you can use hyperref's \texorpdfstring command. For example:

\chapter{\texorpdfstring{\capitalisewords{a book of rhyme}}{A Book of Rhyme}}

Alternatively, you can use hyperref's mechanism for disabling commands within the bookmarks. For example:

\pdfstringdefDisableCommands{%
\let\capitalisewords@firstofone
}

See the hyperref manual for further details.

### 3.2 Excluding Words From Case-Changing

As from v1.09, you can specify words which shouldn't be capitalised unless they occur at the start of (text) using:
This only has a local effect. The global version is:

\texttt{\textbackslash gMFUnocap\{word\}}

For example:

\texttt{\capitalisewords{the wind in the willows}}
\texttt{\MFUnocap{in}}\%
\texttt{\MFUnocap{the}}\%
\texttt{\capitalisewords{the wind in the willows}}

produces:

The Wind In The Willows
The Wind in the Willows

The list of words that shouldn’t be capitalised can be cleared using

\texttt{\MFUclear}

You can also simply place an empty group in front of a word if you don’t want that specific instance to be capitalised. For example:

\texttt{\MFUclear \capitalisewords{the \{}wind in the willows\}}

produces:

The wind In The Willows

This is also a useful way of protecting commands that shouldn’t be parsed. For example:

\texttt{\capitalisewords{this is section \{}\nameref{sec:nocap}\}.}}

produces

This Is Section Excluding Words From Case-Changing.

(No case-changing is applied to \texttt{\nameref{sec:nocap}}. It just happens to already be in title case.)

The package \texttt{mfirstuc-english} loads \texttt{mfirstuc} and uses \texttt{\MFUnocap} to add common English articles and conjunctions, such as “a”, “an”, “and”, “but”. You may want to add
other words to this list, such as prepositions but, as there's some dispute over whether prepositions should be capitalised, I don't intend to add them to this package.

If you want to write a similar package for another language, all you need to do is create a file with the extension .sty that starts with

\NeedsTeXFormat{LaTeX2e}

The next line should identify the package. For example, if you have called the file \texttt{mfirstuc-french.sty} then you need:

\ProvidesPackage{mfirstuc-french}

It's a good idea to also add a version in the final optional argument, for example:

\ProvidesPackage{mfirstuc-french}[2014/07/30 v1.0]

Next load \texttt{mfirstuc}:

\RequirePackage{mfirstuc}

Now add all your \texttt{\MFUnocap} commands. For example:

\MFUnocap{de}

At the end of the file add:

\endinput

Put the file somewhere on \TeX's path, and now you can use this package in your document. You might also consider uploading it to CTAN in case other users find it useful.
4 UTF-8

The \makefirstuc command works by utilizing the fact that, in most cases, \TeX
doesn't require a regular argument to be enclosed in braces if it only consists of a
single token. (This is why you can do, say, \( \frac{1}{2} \) instead of \( \frac{1}{2} \), or \( x^2 \) instead of \( x^2 \), although some users frown on this practice.)

A simplistic version of the \makefirstuc command is:

\newcommand*{\FirstUC}[1]{\MakeUppercase #1}

Here

\FirstUC{abc}

is equivalent to

\MakeUppercase abc

and since \MakeUppercase requires an argument, it grabs the first token (the char-
acter “a” in this case) and uses that as the argument so that the result is: Abc.

The glossaries package needs to take into account the fact that the text may be
contained in the argument of a formatting command, such as \acronymfont, so
\makefirstuc has to be more complicated than the trivial \FirstUC shown above,
but at its basic level, \makefirstuc uses this same method and is the reason why, in
most cases, you don’t need to enclose the first character in braces. So if

\MakeUppercase \textit{stuff}

works, then

\makefirstuc{\textit{stuff}}

should also work and so should

\makefirstuc{\texttt{foo}(\textit{stuff})}

but if

\MakeUppercase \textit{stuff}

doesn’t work, then neither will

\makefirstuc{\textit{stuff}}

nor
\makefirstuc{\texttt{\textbackslash foo\{stuff\}}}
This works correctly when compiled with \LaTeX. This means that `\texttt{\textbackslash makefirstuc\{\textbackslash ABC\} will work} provided you use `\LaTeX \texttt{and the fontspec package}.

Version 2.24 of datatool-base added the command `\texttt{\textbackslash dtl@getfirst@UTF\textbackslash Viii} which attempts to grab both octets. If this command has been defined, \texttt{mfirstuc} will use it when it tries to split the first character from the rest of the word. See the datatool documented code for further details.
# Index

<table>
<thead>
<tr>
<th>C</th>
<th>\texttt{\textbackslash capitalize}words</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{\gls}</td>
<td>\texttt{\glsmakefirstuc}</td>
<td>7</td>
</tr>
<tr>
<td>\texttt{\gls/gMUnocap}</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>datatool-base package</td>
<td>6, 19</td>
</tr>
<tr>
<td>E</td>
<td>\texttt{\ecapitalizefmtwords}</td>
<td>12</td>
</tr>
<tr>
<td>\texttt{\ecapitalizewords}</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>\texttt{\emakefirstuc}</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>fontspec package</td>
<td>19</td>
</tr>
<tr>
<td>\texttt{\MakeFirstUc}</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>\texttt{\mfirstuc}</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>\texttt{\mfirstuc 英文}</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>\texttt{\textcase}</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>\texttt{\textcase 英文}</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>\texttt{\gMFUcapword}</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>\texttt{\gMFUClear}</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>\texttt{\gMFUHyphenfalse}</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>\texttt{\gMFUHyphentru}</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>\texttt{\gMUnocap}</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>\texttt{\textcase 英文}</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>\texttt{\Emakefirstuc}</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>\texttt{glossaries package}</td>
<td>6, 7, 17</td>
<td></td>
</tr>
<tr>
<td>\texttt{\glsmakefirstuc}</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>\texttt{\gMUnocap}</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>\texttt{\gMFUHyphenfalse}</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>\texttt{\gMFUHyphentru}</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>\texttt{\gMUnocap}</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>\texttt{\textcase 英文}</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>\texttt{\Emakefirstuc}</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>