

mfirstuc.sty v1.08: uppercasing first letter

Nicola L.C. Talbot

Dickimaw Books

<http://www.dickimaw-books.com/>

2013-11-04

The glossaries bundle is supplied with the package mfirstuc which provides the command:

`\makefirstuc`

`\makefirstuc{<stuff>}`

This makes the first object of *<stuff>* uppercase unless *<stuff>* starts with a control sequence followed by a non-empty group, in which case the first object in the group is converted to uppercase. Examples:

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

`\makefirstuc{{\em abc}}`

produces *ABC* (first object is `{\em abc}` so equivalent to `\MakeUppercase{\em abc}`), and

`{\makefirstuc{\em abc}}`

produces *abc* (`\em` doesn’t have an argument therefore first object is `\em` so equivalent to `{\MakeUppercase{\em}abc}`).

- `\makefirstuc{{\’a}bc}` produces *Ábc*.
- `\makefirstuc{\ae bc}` produces *Æbc*.
- `\makefirstuc{{\ae}bc}` produces *Æbc*.
- `\makefirstuc{{\ä}bc}` produces *Äbc*.

Note that non-Latin or accented characters appearing at the start of the text must be placed in a group (even if you are using the inputenc package) due to expansion issues.

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}
```

produces: Abc. There is a short cut command which will do this:

```
\xmakefirstuc \xmakefirstuc{\<stuff>}
```

This is equivalent to `\expandafter\makefirstuc\expandafter{\<stuff>}`. So

```
\newcommand{\abc}{abc}
\xmakefirstuc{\abc}
```

produces: Abc.

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

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

```
\glsmakefirstuc \glsmakefirstuc{\<text>}
```

For example:

```
\renewcommand{\glsmakefirstuc}[1]{\MakeTextUppercase #1}
```

Remember to also load textcase (glossaries loads this automatically).

New to mfirstuc v1.06:

`\capitalisewords`

`\capitalisewords{<text>}`

This command apply `\makefirstuc` to each word in `<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>`.

`\xcapitalisewords`

`\xcapitalisewords{<text>}`

This is a short cut for `\expandafter\capitalisewords\expandafter{<text>}`.

If you are using `hyperref` and want to use `\capitalisewords` or `\makefirstuc` (or `\xcapitalisewords/\xmakefirstuc`) 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. See the `hyperref` manual for further details.

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`, so `<text>` consists of just one "word".) Compare with next example:

4. `\newcommand{\mytitle}{a book\space of rhyme.}`
`\xcapitalisewords{\mytitle}`

produces: A Book of Rhyme.