mfirstuc.sty v2.09: sentence and title casing

Nicola L.C. Talbot

Dickimaw Books

dickimaw-books.com

2025-04-03

This document is also available as HTML (mfirstuc-manual.html).

0

The mfirstuc package was originally part of the glossaries bundle (added to glossaries v1.12 in 2008) for use with commands like \Gls , which need to have the first letter converted to uppercase (sentence-case), but as the commands provided by mfirstuc may be used without glossaries, the two were split into separately maintained packages in 2015 (mfirstuc v2.0 and glossaries v4.18).

Version 2.08 has reimplemented \makefirstuc using Lagrange Commands. If there are any compatibility issues, you can rollback to v2.07:

\usepackage{mfirstuc}[=2.07]

Contents

1	Sen	tence-Case	1
	1.1	Examples	1
	1.2	Sentence Case Commands	4
	1.3	Exclusions	8
	1.4	Blockers	10
	1.5	Mappings	12
	1.6	Package Options	13
2	Title	-Case	14
	2.1	Excluding Words From Case-Changing	17
	2.2	PDF Bookmarks	20
	2.3	Examples	21
3	Misc	cellaneous	26
	3.1	Saving Exclusions, Blockers and Mappings in the aux File	26
	3.2	All-Caps	27
	3.3	UTF-8	27
Sy	mbol	s	29
GI	ossar	у	30
Co	mma	nd Summary	32
	Com	mand Summary: Symbols	32
		mand Summary: C	32
		mand Summary: E	32
	Com	mand Summary: G	33
		mand Summary: I	33
		mand Summary: M	33
		mand Summary: X	36
In	day		37

1 Sentence-Case

There are two sentence-case commands provided: \MFUsentencecase and \makefirstuc. A summary of the principle features of the two commands is given in Table 1.1.

Table 1.1: Principle features of \MFUsentencecase and \makefirstuc

Feature	\MFUsentencecase	\makefirstuc
Can expand?	✓	X
Supports exclusions?	✓	✓
Supports blockers?	×	✓
Supports mappings?	×	✓
Semantic commands must be robust?	✓	×

1.1 Examples

Text only (leading UTF-8 now supported):

```
\makefirstuc{\(\delta\) | \(\delta\) | \(\de
```

Leading punctuation is ignored:

```
\makefirstuc{`word'} / 'Word'/'Word'
\MFUsentencecase{`word'}
```

However, if the punctuation character is followed by a blocker or mapping command, the punctuation will need to be excluded otherwise the command will be treated as an exclusion. If possible, use semantic markup instead of directly including the punctuation.

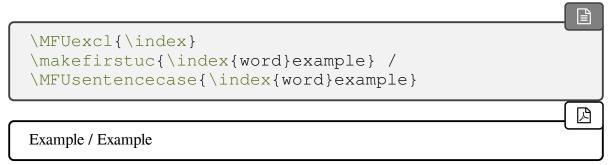
Fragile commands need to be protected with \MFUsentencecase:

```
\newcommand{\alert}[1]{\textcolor{red}{\textbf{#1}}}
\makefirstuc{\alert{\elite}} /
\MFUsentencecase{\protect\alert{\elite}}
```

囚

Élite / Élite

Exclusions are supported by both \makefirstuc and \MFUsentencecase:



Blockers are only supported by \makefirstuc. They are treated as exclusions with \MFUsentencecase, which produces a different result:

```
\MFUblocker{\nameref}
\makefirstuc{\nameref{sec:sentencecase} section} /
\MFUsentencecase{\nameref{sec:sentencecase} section}

Sentence-Case section / Sentence-Case section
```

Mappings are only supported by \makefirstuc. They are treated as exclusions with \MFUsentencecase, which produces a different result:

```
\newrobustcmd{\secref}[1]{section~\ref{#1}}
\newrobustcmd{\Secref}[1]{Section~\ref{#1}}
\MFUaddmap{\secref}{\Secref}
\makefirstuc{\secref{sec:sentencecase} example} /
\MFUsentencecase{\secref{sec:sentencecase} example}
Section 1 example / section 1 example
```

Argument expansion: \MFUsentencecase expands its argument and then skips exclusions whereas \makefirstuc parses its argument without expansion and then passes the relevant content to \glsmakefirstuc. Note that different results will occur with the expanded and unexpanded package options. For example:

```
\usepackage[expanded] {mfirstuc}
  \newcommand{\testa}{sample}
  \newcommand{\testb}{\testa\ test}
  \begin{document}
  \makefirstuc{\testb} / \xmakefirstuc{\testb} /
  \emakefirstuc{\testb} / \MFUsentencecase{\testb}
  \end{document}
                                                                  囚
  Sample test / Sample test / Sample test / Sample test
With the default unexpanded option the result is:
  sample test / sample Test / Sample test / Sample test
Note that this is different from the result in earlier versions of mfirstuc.
  \usepackage{mfirstuc}[=v2.07]
  \newcommand{\testa}{sample}
  \newcommand{\testb}{\testa\ test}
  \begin{document}
  \makefirstuc{\testb} / \xmakefirstuc{\testb} /
  \emakefirstuc{\testb}
  \end{document}
This produces:
  SAMPLE TEST / SAMPLE test / Sample test
If this old result is required, either use rollback or try the grabfirst option.
 Argument expansion with mappings (and the default unexpanded option):
                                                                  \newrobustcmd{\secref}[1]{section~\ref{#1}}
  \newrobustcmd{\Secref}[1]{Section~\ref{#1}}
  \MFUaddmap{\secref}{\Secref}
  \newcommand{\testa}{\secref{sec:sentencecase}}
```

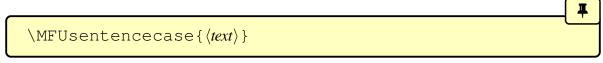
\makefirstuc{\testb} / \xmakefirstuc{\testb} /

\newcommand{\testb}{\testa\ example}

\emakefirstuc{\testb} / \MFUsentencecase{\testb}

section 1 example / section 1 Example / Section 1 example

1.2 Sentence Case Commands



This simply uses \text_titlecase_first:n, which is provided by the LaTeX3 kernel. Note that this fully expands the argument. If you use this command, ensure that your semantic commands are robust. For example:

```
\newrobustcmd{\alert}[1]{\textcolor{red}{\textbf{#1}}
}}
\MFUsentencecase{\alert{sample} text}
Sample text
```

Note that the following will fail:

```
\newcommand{\alert}[1]{\textcolor{red}{\textbf{#1}}}
\MFUsentencecase{\alert{sample} text}
```

This is because the expansion that's performed before the case-change will result in:

```
\textcolor{Red}{\textbf{sample}} text
```

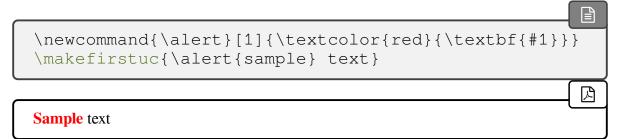
This triggers an error since Red isn't a recognised colour name.

```
\makefirstuc{\langle text\rangle}\}

This is the original command provided by mfirstuc, which was written to assist sentence-casing
```

This is the original command provided by mfirstuc, which was written to assist sentence-casing in the glossaries package and had to take markup into account. It parses its argument to try to determine which part needs the case-change. The actual case-change is performed by \gls-makefirstuc.

As from v2.08, the definition of \makefirstuc has been updated to use LATEX3 code, but the argument is still parsed, which allows for non-robust semantic commands, and now also takes blockers and mappings into account. For example:



The rules used when parsing $\mbox{makefirstuc} \{\langle text \rangle\}$ are as follows:

- 1. if $\langle text \rangle$ is empty do nothing;
- 2. if $\langle text \rangle$ consists of a single item or starts with a group do \glsmakefirstuc{ $\langle text \rangle$ };
- 3. if $\langle text \rangle$ starts with a command $\langle cs \rangle$:
 - a) if $\langle cs \rangle$ is \protect, skip it and continue to the next step where the command $\langle cs \rangle$ under inspection is now the one that follows \protect:
 - b) if $\langle cs \rangle$ has been identified as a blocker, just do $\langle text \rangle$ (that is, no case-change is applied);
 - c) if $\langle cs \rangle$ isn't followed by a group then do \glsmakefirstuc{ $\langle text \rangle$ };
 - d) otherwise assume the format is $\langle cs \rangle \{ \langle arg \rangle \} \langle following \rangle$ (where $\langle following \rangle$ may be empty) and then:
 - i. if $\langle cs \rangle$ has been mapped to $\langle Cs \rangle$ then do $\langle Cs \rangle$ { $\langle arg \rangle$ } $\langle following \rangle$;
 - ii. if $\langle cs \rangle$ has been identified as an exclusion, do $\langle cs \rangle \{ \langle arg \rangle \} \setminus \{ \langle following \rangle \}$;
 - iii. otherwise do $\langle cs \rangle$ { \makefirstuc { $\langle arg \rangle$ } } $\langle following \rangle$.
- 4. otherwise do $\glsmakefirstuc\{\langle text\rangle\}$.

Note that the above algorithm uses recursion. The termination points are: don't implement a change, replace a command with another command, or apply the case-change via \glsmake-firstuc.

For convenience, the following commands are also provided:

\xmakefirstuc{\langle text\rangle}

This is simply a shortcut for:

```
\verb|\expandafter\makefirstuc\expandafter{$\langle \textit{text}\rangle$}|
```

This ensures the first token in $\langle text \rangle$ is expanded before being passed to \makefirstuc.

```
\emakefirstuc{\langle text\rangle}
```

This fully expands its argument before passing it to \makefirstuc. For example:

```
\newcommand{\test}[1]{sample #1}
\newcommand{\tmp}{\test{text}}
\makefirstuc{\tmp} /
\xmakefirstuc{\tmp} /
\emakefirstuc{\tmp}
sample text / sample Text / Sample text
```

If you don't have any mappings or blockers set up, it's simpler to use \MFUsentencecase instead of \emakefirstuc.

With $\ensuremath{\verb|lefirstuc|}$, mappings and blockers will only be detected if they are robust or protected, otherwise they will expand too soon to be detected.

```
\glsmakefirstuc{\langle text\rangle}
```

This command is used by \makefirstuc to perform the case-change. The "gls" prefix is for historical reasons as the original code was part of the glossaries package. With the default unexpanded option, this command is defined as:

```
\MFUsentencecase{\unexpanded{\langle text\rangle}}
```

The use of \unexpanded inhibits the expansion of $\langle text \rangle$ that would otherwise occur. This allows \makefirstuc to maintain as much backward-compatibility with version 2.07 as possible. This is particularly important when using the glossaries package with a style that automatically converts the description to sentence-case, as there may well be documents with complex descriptions that contain fragile commands.

The expanded package option will redefine \glsmakefirstuc to use \MFUsentencecase without \unexpanded. The grabfirst package option will redefine

 $\label{thm:constraint} $$ \glsmakefirstuc to use \mbox{$\mbox{$mfugrabfirstuc to emulates the behaviour of $\glsmakefirstuc in $v2.07$ and below without using rollback.}$

```
\mfugrabfirstuc{\langle text\rangle} avoid where possible
```

This robust command is provided to emulate v2.07 and its use is discouraged. Note that you may still have different results compared to v2.07, so you may need to use rollback instead. (Note that the grabfirst option additionally redefines \mfirstucMakeUppercase to \MakeUppercase, which you will also need.)

This applies \mfirstucMakeUppercase to the first token in $\langle text \rangle$ and leaves the remainder unchanged. This won't work if $\langle text \rangle$ starts with a UTF-8 character unless you use X-ILATEX or LuaLATEX, so avoid using this method where possible. For example:

```
\renewcommand{\mfirstucMakeUppercase}[1]{%
\MakeUppercase{#1}%
}
\mfugrabfirstuc{sample}
```

This is equivalent to:

```
\mfirstucMakeUppercase{s}ample
```

Whereas

```
\renewcommand{\mfirstucMakeUppercase}[1]{%
\MakeUppercase{#1}%
}
\mfugrabfirstuc{\(\elline\)}
```

only works with XHATEX or LualATEX.



As with earlier implementations of \makefirstuc that tried to only apply the casechange to the first token, this won't produce the desired result if the argument starts with a punctuation character.

For example:

\mfugrabfirstuc{`sample'}

This is equivalent to:

\mfirstucMakeUppercase(`)sample'

so no case-change is applied.

1.3 Exclusions



Exclusions will apply to all LATEX3 case-changing commands, so will also affect \mfirstucMakeUppercase, but this is generally desirable.

Ŧ

\MFUexcl $\{\langle cs \rangle\}$

Identifies the text-block command $\langle cs \rangle \{ \langle arg \rangle \}$ as an exclusion. That is, one that should have its argument excluded from a case-change.

Internally, \MFUexcl adds \(\cangle cs \) to LATEX3's exclusion list \l_text_case_exclude _arg_tl, which means that \MFUsentencecase will skip the command and its argument and apply the case-change to the following content. Some common commands (\begin \cite \end \label and \ref) are automatically added by the LATEX3 kernel. The exclusions are also recognised by \makefirstuc when it parses its argument.



The command \NoCaseChange is automatically treated as an exclusion by the LATEX3 case-changing commands with recent LATEX kernels. You may want to add it as a blocker for \makefirstuc and keep \MFUskippunc for exclusions.

Note that \ensuremath is added as an exclusion by mfirstuc otherwise \make-firstuc will pass its argument to \MFUsentencecase, which will cause a problem.

Another way of identifying content that should be excluded with \makefirstuc is to encapsulate it with:



 $\MFUskippunc\{\langle content \rangle\}$

This is a robust command that simply does its argument and is automatically added to the exclusion list. This may be used for cases where the excluded command isn't a simple text-block

command that only has one mandatory argument or where leading punctuation is followed by a blocker or mapping.

The command name stems from earlier versions where leading punctuation needed to be skipped. This is no longer necessary, but the command remains as a general purpose robust exclusion command.

For example, \glsadd (provided by glossaries) should have its argument skipped, since its argument is a label, so the following will ensure that $\glsadd\{\langle label\rangle\}$ will be skipped, and the case-change will be applied to the following text for both $\mbox{MFUsentence}$ and $\mbox{makefirstuc}$.

```
\MFUexcl{\glsadd}
\makefirstuc{\glsadd{ex}some text}
\MFUsentencecase{\glsadd{ex}some text}
```

This will be equivalent to:

```
\glsadd{ex}Some text
```

However, \glsadd can take an optional argument which, if present, will cause a problem. For example:

```
\MFUexcl{\glsadd}
\makefirstuc{\glsadd[counter=section]{ex}some text}
\MFUsentencecase{\glsadd[counter=section]{ex}
some text}
```

Both commands will cause the following error:

```
! Package glossaries Error: Glossary entry `[' has not been defined.
```

This is because the open square bracket is assumed to be the argument of \glsadd, so this effectively becomes:

```
\glsadd{[}Counter=section] exsome text
\glsadd{[}Counter=section] exsome text
```

which means that now [is considered the label and everything that follows is just text.

For this situation to work, you need to encapsulate the problematic content with an exclusion command, such as \NoCaseChange or \MFUskippunc:

```
\makefirstuc{\MFUskippunc{\glsadd[counter=section]
{ex}}some text}
```

This will also work with \MFUsentencecase because mfirstuc automatically adds \MFUskippunc to the exclusion list.



As from glossaries v4.50, \glsadd is automatically added as an exclusion, but be aware of the problem with using the optional argument, as described above.

Punctuation characters are skipped by \MFUsentencecase. For example:

```
\MFUsentencecase{`word'} 'Word'
```

However, leading punctuation will result in \makefirstuc passing its entire argument to \MFUsentencecase (since the argument doesn't start with $\langle cs \rangle \{ \langle arg \rangle \}$), which means that \makefirstuc won't detect any blocker or mapping.

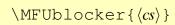
For example:

```
\MFUblocker{\nameref}
\newcommand*{\qt}[1]{``#1''}
\makefirstuc{``\nameref{sec:sentencecase} section''}
/
\makefirstuc{\MFUskippunc{``}\nameref
{sec:sentencecase} section''} /
\makefirstuc{\qt{\nameref{sec:sentencecase} section}}
}
"Sentence-Case section"/"Sentence-Case section"
```

If you are using a package such as csquotes, bear in mind that if the command is followed by an asterisk (a "starred command") or an optional argument then it won't fit the expected $\langle cs \rangle$ { $\langle arg \rangle$ } format and unexpected results will occur.

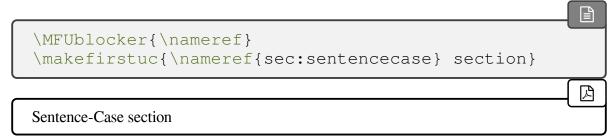
1.4 Blockers

A blocker is a command that prevents any case-change if it occurs at the start of \makefirst-uc or after the argument of an exclusion. Blockers are not supported by \MFUsentence-case but are instead treated as exclusions.

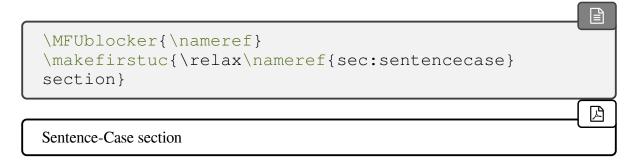


Identifies $\langle cs \rangle$ as a blocker. Since blockers aren't supported by \MFUsentencecase, this automatically implements \MFUexcl { $\langle cs \rangle$ } to protect its argument from \MFUsentencecase, but it won't prevent subsequent content from being changed.

For example:



In the following, the blocker isn't detected by \makefirstuc because the content doesn't start with $\langle cs \rangle$ { $\langle arg \rangle$ } or \protect $\langle cs \rangle$ { $\langle arg \rangle$ }. This means that the entire content is passed to \MFUsentencecase which treats \nameref as an exclusion:



If a blocking command is inside the definition of another command, it won't be visible to \makefirstuc unless the other command is expanded before applying \make-firstuc. For example, by using \xmakefirstuc or \emakefirstuc. If it doesn't get expanded until after it has been passed to \MFUsentencecase, then it will be treated as an exclusion instead.

Recent LATEX kernels provide \NoCaseChange and automatically add it as an exclusion. You may want to add it as a blocker for \makefirstuc. For example:

```
\makefirstuc{\ensuremath{\alpha}-particle} /
\makefirstuc{$\alpha$-particle} /
\MFUsentencecase{\ensuremath{\alpha}-particle} /
\MFUsentencecase{$\alpha$-particle}
```

```
\MFUblocker{\NoCaseChange}%
\makefirstuc{\ensuremath{\alpha}-particle} /
\makefirstuc{\NoCaseChange{$\alpha$}-particle} /
\MFUsentencecase{\ensuremath{\alpha}-particle} /
\MFUsentencecase{\NoCaseChange{$\alpha$}-particle}
```

Note that mfirstuc adds \ensuremath as an exclusion so \makefirstuc will skip it. Whereas the example above that starts with \$\\$ will have the entire argument passed to \MFU-sentencecase, which skips the maths content. In the first paragraph, \NoCaseChange is only an exclusion, but in the second paragraph it's also a blocker.

Exclusion: α -Particle / α -Particle / α -Particle / α -Particle Blocker: α -Particle / α -Particle / α -Particle

1.5 Mappings

A mapping indicates that one command should be substituted for another, instead of applying a case-change. The assumption is that the substituted command should perform the case-change instead. Mappings are not supported by \MFUsentencecase but are instead treated as exclusions.

```
\MFUaddmap\{\langle cs1 \rangle\}\{\langle cs2 \rangle\}
```

Identifies a mapping which indicates that $\mbox{makefirstuc}$ should replace $\langle csI \rangle$ with $\langle cs2 \rangle$ and not apply a case-change. This automatically implements:

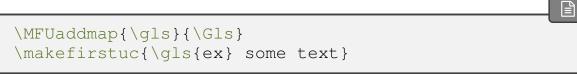
```
\MFUexcl{\langle csI \rangle}\MFUblocker{\langle cs2 \rangle}
```

This means that $\langle cs2 \rangle$ is identified as a blocker (since it's assumed to already be a sentence-case command) and $\langle cs1 \rangle$ is identified as an exclusion to protect its argument from \MFUsentencecase, which doesn't support mappings.

Ð

If either $\langle csI \rangle$ or $\langle cs2 \rangle$ is empty, no mapping is established, but an exclusion or blocker will be set for the non-empty argument.

For example (with glossaries):



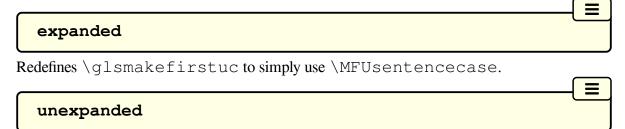
This will be converted to:

```
\Gls{ex} some text
```

Note that this and similar mappings are automatically added in glossaries v4.50+ and glossaries –extra v1.49+.

If a mapped command is inside the definition of another command, it won't be visible to \makefirstuc unless the other command is expanded before applying \make-firstuc. For example, by using \xmakefirstuc or \emakefirstuc. If it doesn't get expanded until after it has been passed to \MFUsentencecase, then it will be treated as an exclusion instead.

1.6 Package Options



Redefines \glsmakefirstuc to use \MFUsentencecase without expanding its argument.

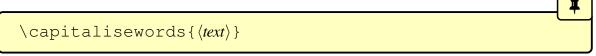


Redefines \glsmakefirstuc to use \mfugrabfirstuc and also redefines \mfirst-ucMakeUppercase to use \MakeUppercase.

This option is best avoided and is only provided to emulate the v2.07 behaviour of \gls makefirstuc. However, you may still have different results compared to v2.07, so you may need to use rollback instead.

2 Title-Case

The title-case commands are designed to convert the first letter of each word in a phrase to uppercase. These commands are robust.

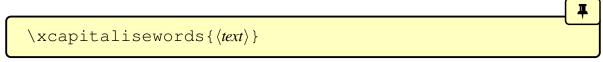


This command applies a sentence-case command to each word in $\langle text \rangle$ where the space character is used as the word separator. Note that it has to be an ordinary space character, not another form of space, such as \sim or \space. Note that no expansion is performed on $\langle text \rangle$. For example:

See §2.1 for excluding words (such as "of") from the case-changing.

This isn't the same as the \LaTeX 3 command \text_titlecase:n, which converts the first letter to uppercase and all other letters to lowercase.

For convenience, there are shortcut commands if expansion is required before parsing the argument:

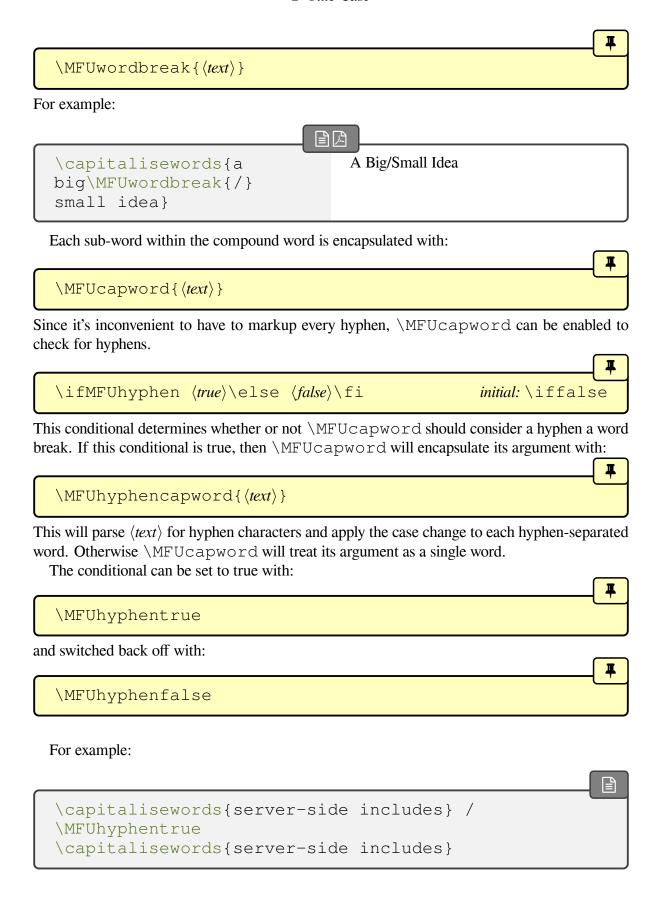


This is a shortcut for:

```
\expandafter\capitalisewords\expandafter{\langle text\rangle}
\ecapitalisewords\langle \langle text\rangle \rangle
```

This fully expands $\langle text \rangle$ before passing it to \capitalisewords.

The parser used by \capitalisewords first splits up the text on each space character. Each of these space-separated words may actually be a compound, so further parsing is performed on each "word". The divisions within the compound word should be marked up with:



囚

圖

囚

modifier: *

Server-side Includes / Server-Side Includes

The actual case-change of each word is performed with:

 $\verb|\MFUcapwordfirstuc|{|}\langle word\rangle||$

This defaults to $\mbox{\mbox{\it makefirstuc}} \{\langle word \rangle\}$ but may be redefined to use $\mbox{\it MFUsentence-}$ case, if preferred.

Hyphens and \MFUwordbreak must be visible to the parser that searches for word breaks. This means they won't be detected if they are within a group or in the definition of a command.

Formatting for the entire phrase must go outside \capitalisewords (unlike \makefirstuc). For example:

\capitalisewords{\emph{a sample phrase}} /
\emph{\capitalisewords{a sample phrase}}

A sample phrase / A Sample Phrase

If your phrase is likely to contain formatting commands, you can instead use:

 $\colon \colon \colon$

where $\langle text \rangle$ may be just words (as with \capitalisewords):

 $\colon \colon \colon$

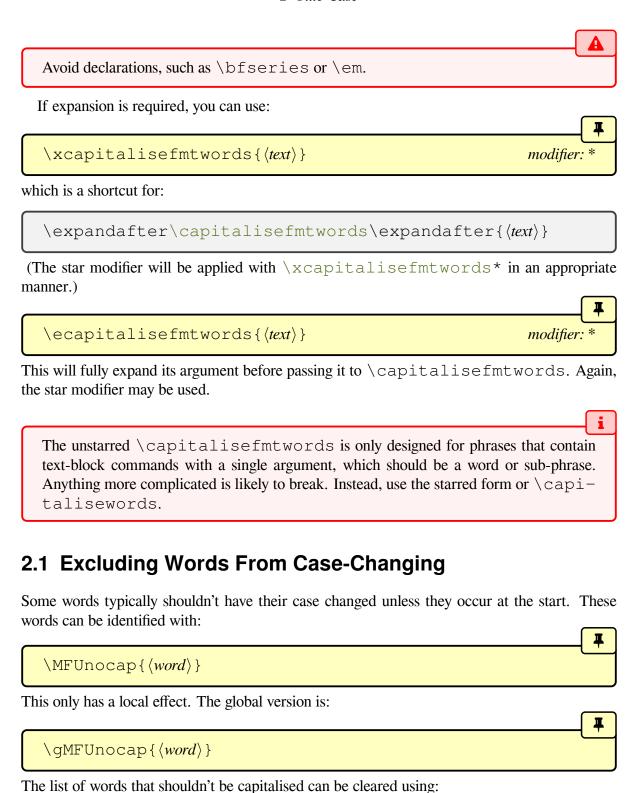
or may be entirely enclosed in a formatting command in the form:

 $\colon \colon \colon$

or contain formatted sub-phrases:

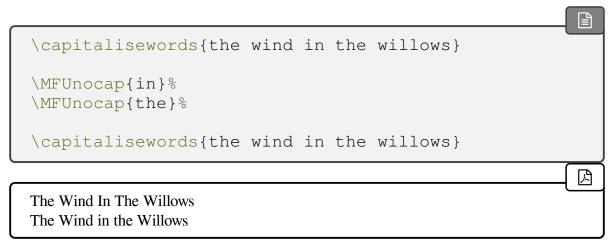
 $\colonerright{\colonerright} \colonerright{\colonerright{\colonerright}\colonerright{\colonerright{\colonerright}\colonerright{\colonerright$

The starred form only permits a text-block command at the start of the phrase. See §2.3 for examples.

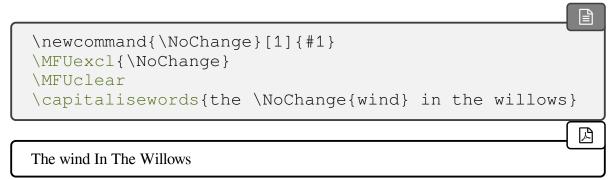


\MFUclear

For example:



Since the case-change is ultimately performed by \makefirstuc, you can also use an exclusion to prevent an individual word from being changed. For example:



This can also work if you redefine \MFUcapwordfirstuc to use \MFUsentence-case provided the exclusion command doesn't expand (so \NoChange would need to be protected or made robust in the above example).

Exceptions only apply to (non-leading) whole words or words separated with \MFU-wordbreak but not to parts of a hyphenated word that are split by \MFUhyphen-capword.

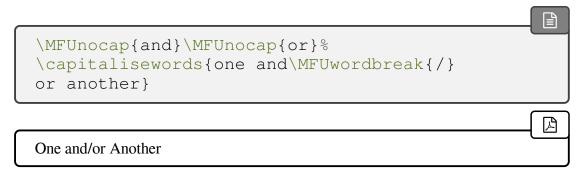
Examples:

1. Exceptions aren't applied if \MFUwordbreak occurs before the first space.

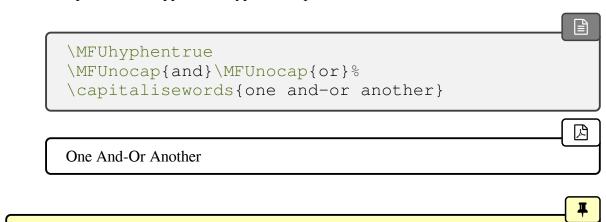


A/The Something

2. Exceptions are applied for non-leading words:



3. Exceptions aren't applied for hyphenated parts:



\usepackage{mfirstuc-english}

The supplementary package mfirstuc—english loads mfirstuc and uses \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. Note that you need to explicitly load mfirstuc—english if you require it. There's no automatic language detection performed by mfirstuc.

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 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 mfirstuc:

```
\RequirePackage{mfirstuc}
```

Now add all your \MFUnocap commands. For example:

```
\MFUnocap{de}
```

At the end of the file add:

```
\endinput
```

Put the file somewhere on T_EX'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.

2.2 PDF Bookmarks

If you are using hyperref and want to use \capitalisewords or \capitalisefmtwords (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 \tex-orpdfstring command. For example:

```
\chapter{\texorpdfstring \( \capitalisewords{a book of rhyme}}\% TeX \( {A Book of Rhyme}\% PDF \)
```

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

```
\pdfstringdefDisableCommands{%
  \let\capitalisewords\@firstofone
}
```

The same applies to \makefirstuc. You can, however, use the expandable \MFUsentencecase. So you may prefer:

```
\pdfstringdefDisableCommands{%
  \let\capitalisewords\MFUsentencecase
  \let\makefirstuc\MFUsentencecase
}
```

See the hyperref manual for further details.

2.3 Examples

1. Text only:

2. Excluding words (see §2.1):

```
\MFUnocap{of} A Little Book of Rhyme \capitalisewords {a little book of rhyme}
```

3. \space isn't recognised as a word boundary:

```
\capitalisewords{a book of rhyme.}
\capitalisewords{a book\space of rhyme.}
```

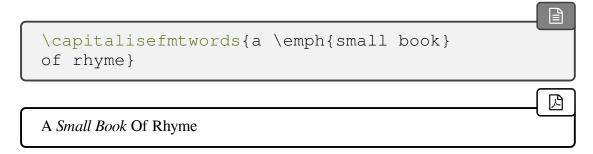
A Book Of Rhyme.
A Book of Rhyme.

4. Phrase entirely enclosed in a formatting command:

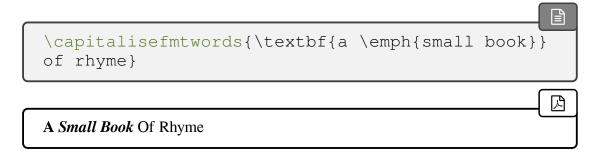
```
\capitalisefmtwords{\emph{a small book of rhyme}}

A Small Book Of Rhyme
```

5. Sub-phrase enclosed in a formatting command:



6. Nested text-block commands:

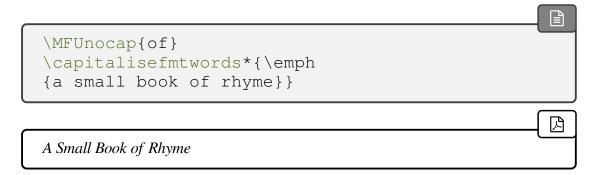


7. Formatting and case-change exception (see §2.1):

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

A Small Book of Rhyme

8. Starred form:

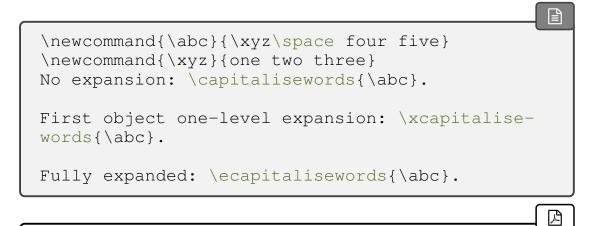


9. The starred form also works with just text (no text-block command):

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

A Small Book of Rhyme
```

10. Expansion:



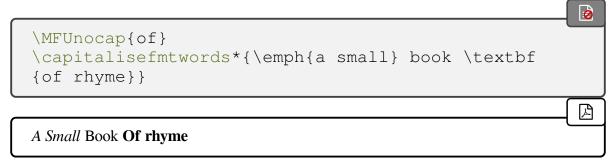
No expansion: one two three four five.

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 \xcap-italisewords, 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.

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:



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}

A Very small Book of Rhyme
```

In this case \emph{very small} is considered a single word.

Grouping causes interference. As with all the commands described here, avoid declarations.

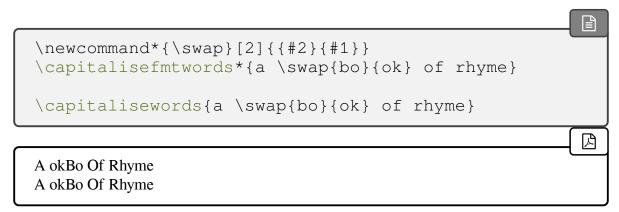
```
\capitalisefmtwords{{\bfseries a \emph{small book}} of rhyme}
```

As a general rule, it's better to define semantic commands rather than directly using font commands and declarations within the document.

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

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

However it can work with the starred form and the simpler \capitalisewords:



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

3 Miscellaneous

3.1 Saving Exclusions, Blockers and Mappings in the aux File

If the exclusions, mappings and blockers are required by some external tool, the information can be saved in the aux file.

\MFUsaveatend

This saves the information at the end of the document using a delayed write. This means that it can register all information identified throughout the document, but there's a chance the document may end before the write takes place (for example, if the last page only contains floats). This command may be counteracted by:

\MFUsave

This saves the information using a protected write at the point where this command occurs, which may be too soon if additional exclusions, mappings or blockers are identified later. This command will counteract any instance of \MFUsaveatend, regardless of whether or not \MFUsaveatend comes before or after \MFUsave.

The associated aux commands are listed below. In each case, a definition is provided in the aux file that does nothing.

 $\verb|\decompactor| @ mfu@ excls { \langle \textit{exclusions} \rangle } |$

Lists all exclusions. For example:

\@mfu@excls{\begin \cite \end \label
\ref \cite \NoCaseChange \ensuremath
\MFUskippunc \gls \glspl}

 $\ensuremath{\verb|deckers||} \ensuremath{\verb|deckers||} \ensuremath{\verb|deckers||} \ensuremath{\verb|deckers||}$

Ŧ

Lists all blockers. For example:

```
\@mfu@blockers{\Gls \Glspl }

\@mfu@mappings{\(\lambda\text{mappings}\right)\}
```

Lists all mappings as a $\langle key \rangle = \langle value \rangle$ list. For example:

```
\@mfu@mappings{ {\gls }= {\Gls }, {\glspl }= {\Glspl }}
```

3.2 All-Caps

```
\verb|\mfirstucMakeUppercase{$\langle \textit{text} \rangle$}|
```

This command was originally used to perform the actual conversion to uppercase and was defined to use command \MakeUppercase. The glossaries package (before v4.50) formerly loaded the textcase package and redefined \mfirstucMakeUppercase to use \MakeText-Uppercase, which was better than \MakeUppercase.

The textcase package has been deprecated as from 2022 and it now simply sets \Make-TextUppercase to \MakeUppercase because the new kernel now defines \Make-Uppercase to use the newer LATEX3 command \text_uppercase:n. Although that command is expandable, \MakeUppercase is robust.

As from v2.08, mfirstuc now defines \mfirstucMakeUppercase so that it uses \text $_{titlecase_first:n}$ directly, rather than indirectly through \makeUppercase, which means that it's now expandable. However, \mfirstucMakeUppercase is no longer used by mfirstuc except in \mfugrabfirstuc, which is provided to emulate v2.07. Note that the grabfirst option will also redefine \mfirstucMakeUppercase to use \makeUppercase.

3.3 UTF-8

file en-

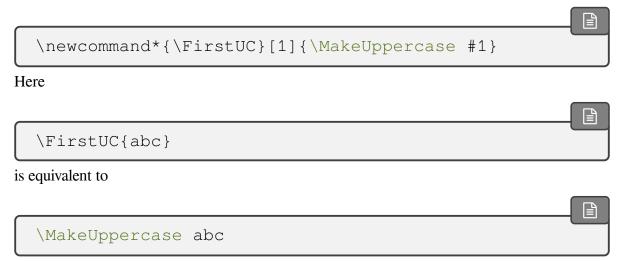
See Binary Files, Text Files and File Encodings^a if you are confused about how file encodings, such as UTF-8, relate to text files.

"dickimaw-books.com/blog/binary-files-text-files-and-file-encodings/

This section only applies to rollback or the use of \mfugrabfirstuc (implemented via the grabfirst option).

Prior to version 2.08, the case-change applied by the \glsmakefirstuc command worked by utilizing the fact that, in most cases, T_EX 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, \frac12 instead of \frac{1}{2} or x^2 instead of x^2 , although this practice is discouraged by some.)

A simplistic version of the original \glsmakefirstuc command is:



and since \MakeUppercase requires an argument, it grabs the first token (the character "a" in this case) and uses that as the argument so that the result is "Abc". This behaviour can be achieved with \mfuqrabfirstuc.

Unfortunately, this will fail if the content starts with a UTF-8 character and you are using pdfLTEX, where each octet of the UTF-8 character is a separate token. This isn't a problem with XELTEX and LuaLTEX which both treat the entire multibyte character as a single token.

LATEX3 now provides \text_titlecase_first:n to convert the first character to uppercase, so now the case-change can be applied with:

```
\ExplSyntaxOn
\text_titlecase_first:n{\(\epsilon\)}
\ExplSyntaxOff
```

This is exactly what \MFUsentencecase does without the need to switch on LATEX3 syntax. The difference with the default definition of \glsmakefirstuc is that it prevents its argument from being expanded.

Symbols

Symbol	Description
Ŧ	The syntax and usage of a command, environment or option etc.
i	An important message.
•	Prominent information.
	LATEX code to insert into your document.
	Problematic code which should be avoided.
凸	How the example code should appear in the PDF.
,0	Text in a transcript or log file or written to STDOUT or STDERR.
Un	Code that requires a native Unicode engine (X\(\frac{1}{2}\)ETEX or Lual\(\frac{1}{2}\)TEX).
=	An option that doesn't take a value.
A	A warning.

Glossary

Blocker

A command that prohibits case-changing. If encountered by \makefirstuc, it won't attempt to apply any case-changing. See §1.4.

Exclusion command

A command whose argument should be skipped by the case-changer. The content that follows the command should have its case-changed instead. See §1.3.

Exclusion word

A word that shouldn't have its case changed by title-case commands, unless the word occurs at the start.

Mapping

A command that should be substituted by another, if encountered by \makefirstuc, instead of applying a case-change. See §1.5.

Semantic command

Essentially, this is a command associated with a particular element, idea or topic that hides the font and other stylistic formatting inside its definition. For example, Latin taxonomy is usually displayed in italic. Explicitly using font commands, for example

```
\textit{Clostridium}
```

is syntactic markup. Whereas defining a command called, say, \bacteria that displays its argument in italics is a semantic command. The actual styling is hidden and the command relates specifically to a particular concept.

```
Syntactic: \textit{Clostridium}
\newrobustcmd*{\bacteria}[1]\emph{#1}%
Semantic: \bacteria{Clostridium}
```

The end result is the same:

人

Syntactic: *Clostridium* Semantic: *Clostridium*

The advantage with semantic commands is that it's much easier to change the style, simply by adjusting the command definition. Note that I've used \newrobustcmd to make the semantic command robust as the style commands can cause a problem if they expand too soon.

Sentence-case

Content that should appear at the start of a sentence that needs to have its first significant character converted to uppercase. See §1.

Title-case

Content that needs to appear in a title that should have each significant word converted to sentence-case. See §2.

Unicode Transformation Format (8-bit) (UTF-8)

A variable-width encoding that uses 8-bit code units. This means that some characters are represented by more that one byte. XalateX and LualateX treat the multi-byte sequence as a single token, but the older LateX formats have single-byte tokens, which can cause complications, although these have mostly been addressed with the newer kernels introduced over the past few years. Related blog article: Binary Files, Text Files and File Encodings.¹

dickimaw-books.com/blog/binary-files-text-files-and-file-encodings/

Symbols

\@mfu@blockers{\langle blockers\rangle} mfirstuc v2.08+

§3.1; 24

An aux file command that lists all the identified blockers.

 $\ensuremath{\verb|Qmfu@excls||} {\ensuremath{|excls||} {\ensuremath{|e$

An aux file command that lists all the identified exclusions.

 $\label{eq:continuous_continuous} $$ \left(\begin{array}{c} \text{mappings} \\ \end{array} \right) $$ mfirstuc v2.08+ $$$

An aux file command that lists all the identified mappings as a $\langle key \rangle = \langle value \rangle$ list.

C

\capitalisefmtwords{\langle text\rangle} modifier: * mfirstuc v2.03+ \quad \quad \quad 2; 15

Similar to \capitalisewords but for content that may contain formatting commands.

\capitalisewords { $\langle text \rangle$ } mfirstuc v1.06+

Applies title-case to $\langle text \rangle$. This will apply \MFUcapword to each word in $\langle text \rangle$ (unless the word has been identified as an exclusion word and doesn't occur at the start of $\langle text \rangle$).

Ε

\ecapitalisefmtwords { $\langle text \rangle$ } modifier: * mfirstuc v2.03+

§2; 16

Fully expands $\langle \textit{text} \rangle$ and passes it to \capitalisefmtwords including the * modifier, if provided.

\ecapitalisewords { $\langle text \rangle$ }

mfirstuc v1.10+

§2; 13

Fully expands (*text*) before passing it to \capitalisewords.

\emakefirstuc{ $\langle text \rangle$ }

mfirstuc v1.10+

§1.2; 5

Fully expands $\langle text \rangle$ before passing it to \makefirstuc.

G

\glsmakefirstuc{\langle text\rangle}

mfirstuc v1.05+

§1.2; 6

Used by \makefirstuc to perform the case-change. This uses \MFUsentencecase $\{\langle text \rangle\}$ but by default will prevent its argument from being expanded.

\gMFUnocap{\langle word\rangle}

mfirstuc v1.09+

§2.1; 16

Globally adds $\langle word \rangle$ to the exclusion word list for \capitalisewords.

ı

\ifMFUhyphen $\langle true \rangle$ \else $\langle false \rangle$ \fi initial: \iffalse mfirstuc v2.03+

§2; 14

Conditional that determines whether or not hyphens should be considered word boundaries by the title-case commands.

M

mfirstuc v1.0+

§1.2; 4

\makefirstuc{\langle text\rangle}

Converts \(\lambda text\)\) to sentence-case internally using \glsmakefirstuc to perform the casechange, but first parses $\langle text \rangle$ to deal with special cases. $\mbox{\tt mfirstucMakeUppercase} \{\langle \textit{text} \rangle\}$ mfirstuc v1.07+ **§3.2**; 25 Converts $\langle text \rangle$ to uppercase. Now only used in \mfugrabfirstuc. **\MFUaddmap** $\{\langle cs1 \rangle\}$ $\{\langle cs2 \rangle\}$ mfirstuc v2.08+ **§1.5**; 11 Identifies a mapping from the command $\langle cs1 \rangle$ to command $\langle cs2 \rangle$. **\MFUblocker** $\{\langle cs \rangle\}$ mfirstuc v2.08+ **§1.4**; 10 Locally identifies $\langle cs \rangle$ as a blocker command. **\MFUcapword** $\{\langle text \rangle\}$ mfirstuc v2.03+ **§2**; 14 Uses either \MFUhyphencapword or \MFUcapwordfirstuc depending on \if-MFUhyphen. **\MFUcapwordfirstuc**{\langle word\rangle} mfirstuc v2.07+ **§2**; 14

Used by \MFUcapword and \MFUhyphencapword to apply the case-change to the given word. This just does \makefirstuc $\{\langle text \rangle\}$ by default.

\MFUclear mfirstuc v1.09+ \\$2.1; 16

Locally clears the exclusion word list for \capitalisewords.

MFUexcl $\{\langle cs \rangle\}$ mfirstuc v2.08+

Locally identifies $\langle cs \rangle$ as an exclusion command.

\mfugrabfirstuc{\langle text\rangle} \ (avoid where possible)	mfirstuc v2.08+
Provide to emulate \glsmakefirstuc in v2.07 and below, but results to v2.07. This will attempt to grab only the first character of uppercase, leaving the rest of $\langle text \rangle$ unchanged. This won't work if $\langle text \rangle$ character, unless you are using X3LATEX or LuaLATEX.	of $\langle text \rangle$ and convert it to

\MFUhyphencapword $\{\langle text \rangle\}$

mfirstuc v2.07+

***** §2; 14

Used by \MFUcapword to apply \MFUcapwordfirstuc to each hyphen-separated word in $\langle text \rangle$, if hyphens should indicate a word-break.

\MFUhyphenfalse

mfirstuc v2.03+

§2; 14

§1.2; 6

Sets \ifMFUhyphen to false.

\MFUhyphentrue

mfirstuc v2.03+

§2; 14

Sets \ifMFUhyphen to true.

\MFUnocap{\langle word \rangle}

mfirstuc v1.09+

§2.1; 16

Locally adds $\langle word \rangle$ to the exclusion word list for \capitalisewords.

\MFUsave

mfirstuc v2.08+

§3.1; 24

Saves the list of exclusions, blockers and mappings to the aux file (if required by some external tool). This command sets itself to \relax so it doesn't repeat the action if used multiple times, and counteracts any use of \MFUsaveatend.

\MFUsaveatend

mfirstuc v2.08+

§3.1; 24

Saves the list of exclusions, blockers and mappings to the aux file (if required by some external tool) at the end of the document. This command sets itself to \relax so it doesn't repeat the action if used multiple times, but it can be overridden by \MFUsave.

\MFUsentencecase { $\langle text \rangle$ } mfirstuc v2.08+	§ 1	1.2; 4

Converts $\langle text \rangle$ to sentence-case with expansion.

$$\label{eq:mfuskippunc} $$ \MFUskippunc {$\langle content \rangle$} $$ mfirstuc v2.07+$$

§2; 13

An exclusion command.

\MFUwordbreak {
$$\langle text \rangle$$
 } mfirstuc v2.07+

Used to markup a character or command that should be treated as a word break by the title-case commands.

X

\xcapitalisefmtwords{\langle text\rangle}	modifier: * mfirstuc v2.03+	§2 ; 15
---	-----------------------------	----------------

Shortcut for \expandafter\capitalisefmtwords\expandafter $\{\langle text \rangle\}$ including the * modifier, if provided.

```
\xcapitalisewords { \langle text \rangle } mfirstuc v1.06+
```

Shortcut for \expandafter\capitalisewords\expandafter $\{\langle text \rangle\}$.

```
\makefirstuc \{\langle text \rangle\} mfirstuc v1.01+
```

Shortcut for \expandafter\makefirstuc\expandafter $\{\langle text \rangle\}$.

Index

Symbols	\emakefirstuc §1.2 ; 5, 6, 10, 12 \end 7, 24
~	\ensuremath
\MFUsaveatend \@mfu@mappings \\$3.1; 25, see also \MFUsave& \MFUsaveatend	file formats aux
В	glossaries – extra package
\begin	\Gls
C	\glsmakefirstuc \ §1.2 ; 2, 4, 5, 6, 12, 25, 26
\capitalisefmtwords §2; 15, 16, 19, see also \capitalisewords, \MFUnocap, \gMFUnocap & \ifMFUhyphen \capitalisewords §2; 13, 15, 16,	\Glspl 24, 25 \glspl 24, 25 \gMFUnocap \\$2.1; 16, see also \MFUnocap & \MFUclear grabfirst \\$1.6; 3, 6, 12, 25
19, 23, see also \capitalisefmtwords, \MFUnocap, \gMFUnocap &	hyperref package
\ifMFUhyphen \cite	<pre>I \ifMFUhyphen</pre>
${f E}$	L

Index

\l_text_case_exclude_arg_tl	N
7 M	\nameref
\makefirstuc §1.2 ; Table 1.1; a, 1, 2, 4, 5-12, 15, 17, 19	P
\MakeTextUppercase 25 \MakeUppercase 6, 12, 25, 26 mapping Table 1.1; 1-5, 8, 9, 11, 12, 24, 25 mfirstuc-english package \$2.1; 18 mfirstuc package a, 3, 4, 8, 9, 11, 18, 25 \mfirstucMakeUppercase \$3.2; 6,	\pdfstringdefDisable- Commands 19 \protect 5 punctuation 1,8,9
7, 12, 25 \MFUaddmap §1.5 ; 11, see also	\ref
\MFUexcl & \MFUblocker \MFUblocker \\$1.4; 10, see also \MFUexcl & \MFUaddmap \MFUcapword \\$2; 14 \MFUcapwordfirstuc \\$2; 14, 17, see also \MFUcapword & \MFUhyphencapword	S semantic command
\MFUclear \\$2.1; 16, see also \gMFUnocap & \MFUnocap \MFUexcl \\$1.3; 7, 10, see also \MFUblocker & \MFUaddmap	\textsup \te
\mfugrabfirstuc \\$1.2; 6, 12, 25, 26 \MFUhyphencapword \\$2; 14, 17, see \also \ifMFUhyphen & \MFUcapword	\text_uppercase:n 25 title-case 12
\MFUhyphenfalse \\$2; 14 \MFUhyphentrue \\$2; 14 \MFUnocap \\$2.1; 16, 18, see also \gMFUnocap & \MFUclear \MFUsave \\$3.1; 24 \MFUsaveatend \\$3.1; 24 \MFUsentencecase \\$1.2; Table 1.1; 1,	\unexpanded 6 unexpanded
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	\xcapitalisefmtwords \\$2; 15, 16 \xcapitalisewords \\$2; 13, 22 \xmakefirstuc \\$1.2; 5, 10, 12