

# The `textcase` package\*

David Carlisle†

2004/10/07

## 1 Introduction

`\MakeTextUppercase` and `\MakeTextLowercase` are versions of the standard `\MakeUppercase` and `\MakeLowercase` that do not change the case of any math sections in their arguments.

```
\MakeTextUppercase{abc\ae\ \ ( a = b \ ) and $\alpha \neq a$  
or even \ensuremath{x=y} and $\ensuremath{x=y}$}
```

Should produce:

ABCÆ  $a = b$  AND  $\alpha \neq a$  OR EVEN  $x = y$  AND  $x = y$

Version 0.03 of this package incorporates some changes suggested by Donald Arseneau so that as well as math mode, the arguments of `\cite`, `\label` and `\ref` are also prevented from being uppercased. So you can now go

```
\MakeTextUppercase{%  
Text in section~\ref{intro}, about \cite[pp 2--4]{bbb}}
```

which produces

TEXT IN SECTION 1, ABOUT [1, PP 2–4]

If, instead, the standard `\MakeUppercase` were used here, the keys ‘intro’ and ‘bbb’ would be uppercased and generate errors about undefined references to `INTRO` and `BBB`.

Sometimes there may be a special section of text that should not be uppercased. This can be marked with `\NoCaseChange`, as follows.

```
\MakeTextUppercase{%  
Text \NoCaseChange{More Text} yet more text}
```

which produces

TEXT More Text YET MORE TEXT

---

\*This file has version number v0.07, last revised 2004/10/07.

†Reorganisation to support skipping a wider class of things than just math done by Donald Arseneau

`\NoCaseChange` has other uses. If for some reason you need a tabular environment within an uppercased section, then you need to ensure that the name ‘tabular’ and the preamble (eg ‘ll’) does not get uppercased:

```
\MakeTextUppercase{%
  Text \NoCaseChange{\begin{tabular}{ll}}%
                    table&stuff\goes&here
  \NoCaseChange{\end{tabular}}
  More text}
```

which produces

```
TEXT TABLE STUFF MORE TEXT
      GOES  HERE
```

## 2 Features and Foibles

### 2.1 Nested text

The commands defined here only skip math sections and `\ref` arguments if they are not ‘hidden’ inside a `{ }` brace group. All text inside such a group will be made uppercase just as with the standard `\MakeUppercase`.

```
\MakeTextUppercase{a b {c $d$} $e$}
```

produces

```
A B C D e
```

Of course, this restriction does not apply to the arguments of the supported commands `\ensuremath`, `\label`, `\ref`, and `\cite`.

If you cannot arrange for your mathematics to be at the outer level of brace grouping, you should use the following basic technique (which works even with the standard `\MakeUppercase` command). Define a new command that expands to your math expression, and then use that command, with `\protect`, in the text to be uppercased. Note that if the text being uppercased is in a section title or other moving argument you may need to make the definition in the document preamble, rather than just before the section command, so that the command is defined when the table of contents file is read.

```
\MakeTextUppercase{%
  Text \fbox{$a=b$ and $x=y$}}%

\newcommand{\mathexprone}{$a=b$}
\newcommand{\mathexprtwo}{$x=y$}
\MakeTextUppercase{%
  Text \fbox{\protect\mathexprone\ and \protect\mathexprtwo}}%
```

which produces

```
TEXT 

|                     |
|---------------------|
| $A = B$ AND $X = Y$ |
|---------------------|


TEXT 

|                     |
|---------------------|
| $a = b$ AND $x = y$ |
|---------------------|


```

## 2.2 Citations

As documented above, `\cite` and `\ref` commands are not uppercased by `\MakeTextUppercase`. If you are using a non-numeric citation scheme you may want the replacement text for `\cite` to be uppercased.

It is difficult to arrange that `\MakeTextUppercase` uppercases such text, not least because this would lead to interaction with the many bibliography packages which redefine `\cite` one way or another. One possibility to achieve this is to use Donald Arseneau's `cite` package and to locally redefine `\citeform` to add `\MakeUppercase` around the final text string produced by `\cite`.

```
\MakeTextUppercase{%
  Text \cite{bbb} and \cite{ccc}}

{\renewcommand\citeform{\MakeUppercase}\MakeTextUppercase{%
  Text \cite{bbb} and \cite{ccc}}}
```

which produces<sup>1</sup>

```
TEXT [1] AND [David Carlisle 1997]
TEXT [1] AND [DAVID CARLISLE 1997]
```

## 2.3 overload Option

By default the package only defines new commands, `\MakeTextUppercase` and `\MakeTextlowercase` as described above. You may wish to redefine the standard `\MakeUppercase` and `\MakeLowercase` commands to be the same as these new commands. So that for example headings in the book class have this new feature without any further redefinition. You may use the package option `[overload]` in which case these new definitions will be overloaded on to the existing command names.

## References

[1] Something numeric

[David Carlisle 1997] Something textual

## 3 Implementation

1 `(*package)`

`\@uclnotmath` This is the main macro of this package. It is basically a copy of `\MakeTextUppercase` and `\MakeTextLowercase` from the L<sup>A</sup>T<sub>E</sub>X kernel, modified slightly so that they can share code (that modification could be done to the standard versions as well) and then further changed to skip certain features like math mode and `\label` arguments.

The arguments are:

`#1`: Extra commands to apply for case changing. Used to locally redefine `\i` and `\j` for uppercasing.

---

<sup>1</sup>This is faked, so this document does not rely on `cite.sty` being installed

#2: Either ##1##2 or ##2##1 to control the order in which `\let` is applied to the pairs of control sequences in `\@uclclist`.

#3: `\uppercase` or `\lowercase`.

#4: The text to be upper (or lower) cased.

```
2 \def\@uclcnotmath#1#2#3#4{\begingroup
```

Run extra commands (currently just to redefine `\i` and `\j`).

```
3     #1%
```

Locally set `\( \)` to be just `$ $`, so that the math skipping code can be simplified, just to look for `$`.

```
4     \def\({$}\let)\(%
```

Set up the ‘non-math’ things that also have to be skipped.

```
5     \def\NoCaseChange##1{\noexpand\NoCaseChange{\noexpand##1}}%
```

```
6     \@nonchange\label
```

```
7     \@nonchange\ref
```

```
8     \@nonchange\ensuremath
```

`\cite` a bit trickier, as we want to uppercase any optional argument. This will fail if the optional argument contains a brace group, but should catch most cases.

text `\cite[page 1]{foo}` more text ends up as

```
\uppercase{text \toks@{\cite[page1]}}%
```

```
\the\toks@{foo}%
```

```
\uppercase{ more text}
```

```
9     \def\cite##1##{\toks@{\noexpand\cite##1}\@citex}%
```

```
10    \def\@citex##1{\NoCaseChange{\the\toks@{##1}}%
```

(`\@citex` is a scratch macro here, not a redefinition of the existing `\@citex`.)

The following is essentially taken from `\MakeUppercase`. Recursively execute `\reserved@a` to `\let` the pairs in `\@uclclist`. The strange construction with `\@gobble` at the end just gobbles the final recursive call.

Incidentally, packages should not use the `\reserved@...` scratch macros, which are ‘reserved’ for use within the L<sup>A</sup>T<sub>E</sub>X kernel, but (a) this code is essentially a copy from the kernel, and (b) I’m allowed to break the rules, so there.

```
11    \def\reserved@a##1##2{\let#2\reserved@a}%
```

```
12    \expandafter\reserved@a\@uclclist\reserved@b{\reserved@b\@gobble}%
```

Expand everything first so that the ‘skipping’ code can see what to skip and so that tokens are revealed to `\uppercase`. This makes the argument ‘moving’. The `$\valign$` is just a fake math expression used to terminate the parsing done by `\@skipmath`.

```
13    \protected@edef\reserved@a{\endgroup
```

```
14        \noexpand\@skipmath#3#4$\valign$}%
```

```
15    \reserved@a}
```

`\@nonchange`

```
16 \def\@nonchange#1{\def#1##1{\NoCaseChange{#1{##1}}}}
```

`\NoCaseChange` For hiding arbitrary text from `\uppercase`. This innocuous definition is used for any occurrence of `\NoCaseChange` in text that is not passed to `\MakeTextUppercase`. For example a section heading may be uppercased, but the toc entry may not. It is also used for nested definitions of `\cite` etc, where

the `\NoCaseChange` is inserted by expansion, but not removed as it is hidden by the brace group.

```
17 \let\NoCaseChange\@firstofone
```

```
\@skipmath #1: operation \uppercase / \lowercase
#2: text up to first (next) $
#3: first math mode material (or \valign sentinel)
18 \def\@skipmath#1#2#3${%
19   \@skip@nonchange#1#2\NoCaseChange\valign
20   \ifx\valign#3%
21     \else
22       $#3$%
23     \expandafter\@skipmath\expandafter#1%
24   \fi}
```

```
\@skip@nonchange#1: \uppercase or \lowercase
#2: Text up to the first command (such as \cite) made ‘safe’. or the first use of
\NoCaseChange.
25 \def\@skip@nonchange#1#2\NoCaseChange#3${%
26   #1{#2}%
27   \ifx\valign#3%
28     \else
29       #3%
30     \expandafter\@skip@nonchange\expandafter#1%
31   \fi}
```

```
\MakeTextUppercase Put it all together. Arrange for i and j to uppercase, and to read the pairs in the
ucase list ‘forwards’.
32 \DeclareRobustCommand\MakeTextUppercase{%
33   \@uclnotmath{\def\i{I}\def\j{J}}{##1##2}\uppercase}
34 \protected@edef\MakeTextUppercase#1{\MakeTextUppercase{#1}}
```

```
\MakeTextLowercase probably will never be used, but it is easy to implement. Use \lowercase and
read the pairs in the uppercase list ‘backwards’.
35 \DeclareRobustCommand\MakeTextLowercase{%
36   \@uclnotmath}{##2##1}\lowercase}
37 \protected@edef\MakeTextLowercase#1{\MakeTextLowercase{#1}}
```

`\MakeUppercase`

```
\MakeLowercase 38 \DeclareOption{overload}{%
39   \expandafter\let\csname MakeUppercase \expandafter\endcsname
40     \csname MakeTextUppercase \endcsname
41   \expandafter\let\csname MakeLowercase \expandafter\endcsname
42     \csname MakeTextLowercase \endcsname}
```

```
43 \ProcessOptions
```

```
44 \</package>
```