

Documented Code For mfirstuc v2.07

Nicola L.C. Talbot

Dickimaw Books

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

2021-10-15

Contents

1 Introduction	1
2 Main Package Code	1
2.1 Mfirstuc Documented Code	1
2.2 Mfirstuc-english Documented Code	14
Change History	14
Change History	15
Index	16

1 Introduction

This is the documented code for the mfirstuc package. See mfirstuc-manual.pdf for the main user guide.

2 Main Package Code

2.1 Mfirstuc Documented Code

```
1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{mfirstuc}[2021/10/15 v2.07 (NLCT)]
```

Requires etoolbox:

```
3 \RequirePackage{etoolbox}
```

`\makefirstuc` Syntax:

```
\makefirstuc{<text>}
```

Makes the first letter uppercase, but will skip initial control sequences if they are followed by a group and make the first thing in the group uppercase, unless the group is empty. Thus `\makefirstuc{abc}` will produce: `Abc`, `\makefirstuc{\ae bc}` will produce: `Æbc`, but `\makefirstuc{\emph{abc}}` will produce `Abc`. This is required by `\Gls` and `\Glspl`.

```
4 \newif\if@glscs
5 \newtoks\@glsmfirst
6 \newtoks\@glsmrest
7 \newrobustcmd*{\makefirstuc}[1]{%
```

Check if argument starts with `\protect` which needs to be skipped.

```
8 \MFU@p@makefirstuc#1\relax\relax\relax@end@MFU@protected{#1}%
9 }
```

```
\MFU@protect
```

```
10 \newcommand*{\MFU@protect}{\protect}
```

```
\MFU@skippunc
```

```
11 \newcommand*{\MFU@skippunc}{\MFU@skippunc}
```

```
\MFU@skippunc
```

```
12 \newrobustcmd*{\MFU@skippunc}[1]{#1}
```

```
U@p@makefirstuc
```

Internal command to do the required action.

```
13 \def\MFU@p@makefirstuc#1#2#3@end@MFU@protected#4{%
```

Test if the first argument is `\protect`.

```
14 \def\gls@argi{#1}%
15 \ifx\gls@argi\MFU@protect
16 \makefirstuc{#2#3}%
17 \else
```

v2.07+ also check for `\MFU@skippunc`

```
18 \ifx\gls@argi\MFU@skippunc
19 #2\makefirstuc{#3}%
20 \else
21 \makefirstuc{#4}%
22 \fi
23 \fi
24 }
```

```
\makefirstuc
```

Internal command to do the required action.

```
25 \newcommand*{\makefirstuc}[1]{%
26 \def\gls@argi{#1}%
27 \ifx\gls@argi@empty
```

If the argument is empty, do nothing.

```
28 \else
```

```

29 \def\@gls@tmp{\ #1}%
30 \@onelevel@sanitize\@gls@tmp
31 \expandafter\@gls@checkcs\@gls@tmp\relax\relax
32 \if@glscs

```

Check if we have two sequential control sequences.

```

33 \expandafter\def\expandafter\@gls@tmp\expandafter{\expandafter
34 \ \@gobble #1}%
35 \@onelevel@sanitize\@gls@tmp
36 \expandafter\@gls@checkcs\@gls@tmp\relax\relax
37 \if@glscs\@glscsfalse\else\@glscstrue\fi
38 \fi
39 \if@glscs
40 \@gls@getbody #1{\@nil
41 \ifx\@gls@rest\@empty
42 \glsmakefirstuc{#1}%
43 \else
44 \expandafter\@gls@split\@gls@rest\@nil
45 \ifx\@gls@first\@empty
46 \glsmakefirstuc{#1}%
47 \else

```

Does \@gls@first start with a control sequence?

```

48 \edef\@gls@tmp\expandafter{\noexpand\ \expandonce\@gls@first}%
49 \@onelevel@sanitize\@gls@tmp
50 \expandafter\@gls@checkcs\@gls@tmp\relax\relax
51 \if@glscs
52 \edef\@gls@domfirstuc{\expandonce\@gls@body
53 {\noexpand\makefirstuc{\expandonce\@gls@first
54 \expandonce\@gls@rest}}%
55 }%
56 }%
57 \else
58 \expandafter\@glsmfirst\expandafter{\@gls@first}%
59 \expandafter\@glsmrest\expandafter{\@gls@rest}%

60 \edef\@gls@domfirstuc{\noexpand\@gls@body
61 {\noexpand\glsmakefirstuc{\the\@glsmfirst}}%
62 \the\@glsmrest}%
63 \fi
64 \@gls@domfirstuc
65 \fi
66 \fi
67 \else
68 \glsmakefirstuc{#1}%
69 \fi
70 \fi
71 }

```

\@gls@split Put first argument in \@gls@first and second argument in \@gls@rest:

```

72 \ifdef\dtl@getfirst@UTFviii
73 {%
74   \def\@gls@split#1#2\@nil{%
75     \ifbool{\@dtl@utf8}
76       {%
77         \expandafter\dtl@if@two@octets#1#2\relax\dtl@end@if@two@octets
78         {%
79           \dtl@getfirst@UTFviii#1#2\@nil\end\dtl@getfirst@UTFviii
80           \let\@gls@first\dtl@first
81           \let\@gls@rest\dtl@rest
82         }%
83       }%
84       \def\@gls@first{#1}\def\@gls@rest{#2}%
85     }%
86   }%
87   {%

88     \def\@gls@first{#1}\def\@gls@rest{#2}%
89   }%
90 }
91 }%
92 {%
93   \def\@gls@split#1#2\@nil{%
94     \def\@gls@first{#1}\def\@gls@rest{#2}%
95   }
96 }

97 \def\@gls@checkcs#1 #2#3\relax{%
98   \def\@gls@argi{#1}\def\@gls@argii{#2}%
99   \ifx\@gls@argi\@gls@argii
100     \@glscstrue
101   \else
102     \@glscsfalse
103   \fi
104 }

```

Has datatool-base v2.24 (or above) been loaded? If so and we have UTF-8 enabled, we can grab the first octet.

fu@applytofirst

```

105 \ifdef\dtl@getfirst@UTFviii
106 {%
107   \def\@mfu@applytofirst#1#2\@mfu@end@applytofirst#3{%
108     \expandafter\dtl@if@two@octets#1#2\relax\dtl@end@if@two@octets
109     {%
110       \dtl@getfirst@UTFviii#1#2\@nil\end\dtl@getfirst@UTFviii
111       \expandafter#3\expandafter{\dtl@first}\dtl@rest
112     }%
113   }%

```

First argument needs to be grouped in case it consists of more than one token.

```

114     #3{#1}#2%
115   }%
116 }%
117 }%
118 {%

First argument needs to be grouped in case it consists of more than one token.
119 \def\@mfu@applytofirst#1#2\@mfu@end@applytofirst#3{#3{#1}#2}
120 }

```

MFUapplytofirst

```
\MFUapplytofirst{<cs>}{<text>}
```

Apply `<cs>` to first token of text, or first UTF-8 character if datatool-base v2.24 above has been loaded with the utf8 support enabled.

```

121 \ifdef\dtl@getfirst@UTFviii
122 {%
123   \newcommand*\MFUapplytofirst}[2]{%
124     \ifbool{@dtl@utf8}
125       {%
126         \ifblank{#2}%
127         {#1{#2}}%
128         {%
129           \ifblank{#1}%
130           {#1#2}%
131           {%
132             \@mfu@applytofirst#2\@mfu@end@applytofirst#1\relax
133             }%
134           }%
135         }%
136       }%
137     #1#2%
138   }%
139 }
140 }%
141 {%
142   \newcommand*\MFUapplytofirst}[2]{#1#2}
143 }

```

gls@makefirstuc

Make first thing upper case:

```
144 \def\@gls@makefirstuc#1{\MFUapplytofirst\mfirstucMakeUppercase{#1}}
```

ucMakeUppercase

Allow user to replace `\MakeUppercase` with another case changing command.

```
145 \newcommand*\mfirstucMakeUppercase{\MakeUppercase}
```

\glsmakefirstuc

Provide a user command to make it easier to customise.

```
146 \newcommand*\glsmakefirstuc}[1]{\@gls@makefirstuc{#1}}
```

Get the first grouped argument and store in `\@gls@body`.

```
147 \def\@gls@getbody#1#\def\@gls@body{#1}\@gls@gobbletonil}
```

Scoup up everything to \@nil and store in \@gls@rest:
148 \def\@gls@gobbletonil#1\@nil{\def\@gls@rest{#1}}

\xmakefirstuc Expand argument once before applying \makefirstuc (added v1.01).

```
149 \newcommand*\xmakefirstuc}[1]{%
150 \expandafter\makefirstuc\expandafter{#1}%
151 }
```

\emakefirstuc Fully expand argument before applying \makefirstuc

```
152 \newrobustcmd*\emakefirstuc}[1]{%
153 \protected@edef\@MFU@caparg{#1}%
154 \expandafter\makefirstuc\expandafter{\@MFU@caparg}%
155 }
```

capitalisewords Capitalise each word in the argument. Words are considered to be separated by plain spaces (i.e. non-breakable spaces won't be considered a word break).

```
156 \newrobustcmd*\capitalisewords}[1]{%
157 \let\@mfu@checkword\@gobble
158 \def\gls@add@space{}%
159 \let\@mfu@domakefirstuc\@MFU@dofirst@capword
160 \mfu@capitalisewords#1 \@nil\mfu@endcap
161 }
```

capitalisewords \@mfu@checkword needs to be set before use.

```
162 \newcommand*\@mfu@mid@capitalisewords}[1]{%
163 \def\gls@add@space{}%
164 \let\@mfu@domakefirstuc\@MFU@do@capword
165 \mfu@capitalisewords#1 \@nil\mfu@endcap
166 }
```

```
167 \def\mfu@capitalisewords#1 #2\mfu@endcap{%
168 \def\mfu@cap@first{#1}%
169 \def\mfu@cap@second{#2}%
170 \gls@add@space
171 \@mfu@checkword{#1}%
172 \@mfu@domakefirstuc{#1}%
173 \def\gls@add@space{ }%
174 \ifx\mfu@cap@second\@nnil
175 \let\next@mfu@cap\mfu@noop
176 \else
177 \let\next@mfu@cap\mfu@capitalisewords
178 \let\@mfu@checkword\mfu@checkword
179 \fi
180 \next@mfu@cap#2\mfu@endcap
181 }
182 \def\mfu@noop#1\mfu@endcap{ }
```

\ifMFUhyphen Split on hyphens.

```
183 \newif\ifMFUhyphen
184 \MFUhyphenfalse
```

@MFU@do@capword

```
185 \newcommand*{\@MFU@do@capword}[1]{%
186 \let\@MFU@cappingword\MFUcapword
187 \@MFUwordbreak#1\MFUwordbreak}\@nil\@endMFUwordbreak
188 }
```

dofirst@capword

```
189 \newcommand*{\@MFU@dofirst@capword}[1]{%
190 \let\@MFU@cappingword\MFUcapword
191 \@MFUfirstwordbreak#1\MFUwordbreak}\@nil\@endMFUfirstwordbreak
192 }
```

\MFUcapword

```
193 \newcommand*{\MFUcapword}[1]{%
194 \ifMFUhyphen
195 \MFUhyphencapword{#1}%
196 \else
197 \MFUcapwordfirstuc{#1}%
198 \fi
199 }
```

Ucapwordfirstuc

```
200 \newcommand*{\MFUcapwordfirstuc}[1]{%
201 \makefirstuc{#1}%
202 }
```

\MFUwordbreak Markup a character (or command) that should be treated as a word break.

```
203 \newrobustcmd*{\MFUwordbreak}[1]{#1}
```

FUhyphencapword

```
204 \newcommand*{\MFUhyphencapword}[1]{%
205 \@MFUcapword#1-\@nil\@endMFUcapword
206 }
```

\@MFUcapword

```
207 \def\@MFUcapword#1-#2\@endMFUcapword{%
208 \MFUcapwordfirstuc{#1}%
209 \def\@mfu@argii{#2}%
210 \ifx\@mfu@argii\@nnil
211 \else
212 -\@MFUcapword#2\@endMFUcapword
213 \fi
214 }
```

\@MFUwordbreak

```
215 \def\@MFUwordbreak#1\MFUwordbreak#2#3\@endMFUwordbreak{%
216 \def\@mfu@argiii{#3}%
217 \ifx\@mfu@argiii\@nnil
```

```

218 \MFU@cappingword{#1}%
219 \else
220 \let\MFU@cappingword\MFU@wordbreakcapword
221 \MFU@cappingword{#1}%
222 #2\MFU@wordbreak#3\endMFU@wordbreak
223 \fi
224 }

```

ordbreakcapword

```

225 \newcommand*\MFU@wordbreakcapword}[1]{%
226 \mfu@checkword{#1}%
227 \MFU@domakefirstuc{#1}%
228 }

```

Ufirstwordbreak

```

229 \def\MFU@firstwordbreak#1\MFU@wordbreak#2#3\endMFU@firstwordbreak{%
230 \def\mfu@argiii{#3}%
231 \ifx\mfu@argiii\@nnil
232 \MFU@cappingword{#1}%
233 \else
234 \let\MFU@cappingword\MFU@firstwordbreakcapword
235 \MFU@cappingword{#1}%
236 #2\MFU@firstwordbreak#3\endMFU@firstwordbreak
237 \fi
238 }

```

ordbreakcapword

```

239 \newcommand*\MFU@firstwordbreakcapword}[1]{%
240 \mfu@domakefirstuc{#1}%
241 }

```

\mfu@checkword Check if word should be capitalised. This originally used etoolbox's \ifinlist command but this doesn't work if the word contains grouping.

```

242 \newcommand*\mfu@checkword[1]{%
243 \def\mfu@checkword@arg{#1}%
244 \let\mfu@domakefirstuc\MFU@do@capword
245 \forlistloop\mfu@checkword@do\mfu@nocaplist
246 }

```

mfu@checkword@do Handler for loop used by \mfu@checkword

```

247 \newcommand*\mfu@checkword@do}[1]{%
248 \def\mfu@checkword@argii{#1}%
249 \ifdefequal\mfu@checkword@arg\mfu@checkword@argii
250 {%
251 \let\mfu@domakefirstuc\@firstofone
252 \listbreak
253 }%
254 {%

```


Need to check if \@mfu@terminator has been tacked onto the end.

```
255 \def\mfu@checkword@argii{#1{\@mfu@terminator}}%
256 \ifdefequal\mfu@checkword@arg\mfu@checkword@argii
257 {%
258 \let\@mfu@domakefirstuc\@firstofone
259 \listbreak
260 }%
261 {}%
262 }%
263 }
```

\@mfu@nocaplist List of words that shouldn't be capitalised.

```
264 \newcommand*\@mfu@nocaplist{}
```

\MFUnocap Provide the user with a means to add a word to the list. (No expansion performed on the argument.)

```
265 \newcommand*\MFUnocap[1]{\listadd{\@mfu@nocaplist}{#1}}
```

\gMFUnocap Global version.

```
266 \newcommand*\gMFUnocap[1]{\listgadd{\@mfu@nocaplist}{#1}}
```

\MFUclear Clear the list

```
267 \newcommand*\MFUclear{\renewcommand*\@mfu@nocaplist{}}
```

capitalisewords Short-cut command:

```
268 \newcommand*\xcapitalisewords[1]{%
269 \expandafter\capitalisewords\expandafter{#1}%
270 }
```

capitalisewords Fully expand argument before applying \capitalisewords

```
271 \newrobustcmd*\ecapitalisewords[1]{%
272 \protected@edef\@MFU@caparg{#1}%
273 \expandafter\capitalisewords\expandafter{\@MFU@caparg}%
274 }
```

\@mfu@nil

```
275 \newcommand\@mfu@nil{\unskip}
```

\@mfu@nnil

```
276 \newcommand*\@mfu@nnil{\@mfu@nil}
```

italisefmtwords Like \capitalisewords but assumes that the phrase contains formatting text-block commands.

```
277 \newcommand*\capitalisefmtwords{%
278 \@ifstar\@s@capitalisefmtwords\@ns@capitalisefmtwords
279 }
```

italisefmtwords Starred form.

```
280 \newcommand*{\@s@capitalisefmtwords}[1]{%
281 \@@s@checkgrp@@capitalisefmtwords#1\@nil
282 }
```

italisefmtwords

```
283 \newcommand*{\@@s@checkgrp@@capitalisefmtwords}{%
284 \@ifnextchar\bgroup{\@@s@noc@capitalisefmtwords}{\@@s@capitalisefmtwords}%
285 }
```

italisefmtwords Argument starts with a group.

```
286 \def\@@s@noc@capitalisefmtwords#1\@nil{%
287 \capitalisewords{#1}%
288 }
```

italisefmtwords

```
289 \newcommand*{\@@s@capitalisefmtwords}[1]{%
290 \def\gls@argi{#1}%
291 \ifx\gls@argi\@MFU@protect
Skip \protect
292 \let\@mfu@capfmtwordsnext\@@s@capitalisefmtwords
293 \else
294 \ifx\gls@argi\@nnil
Nothing to do.
295 \let\@mfu@capfmtwordsnext\relax
296 \else
297 \def\@gls@tmp{\ #1}%
298 \@onelevel@sanitize\@gls@tmp
299 \expandafter\@gls@checkcs\@gls@tmp\relax\relax
300 \if@glscs
301 \def\@mfu@capfmtwordsnext{\@@s@cs@capitalisefmtwords#1}%
302 \else
303 \def\@mfu@capfmtwordsnext{\@@s@noc@capitalisefmtwords#1}%
304 \fi
305 \fi
306 \fi
307 \@mfu@capfmtwordsnext
308 }
```

italisefmtwords

```
309 \newcommand*{\@@s@cs@capitalisefmtwords}[1]{%
310 \@ifnextchar\bgroup
311 {\@@s@cs@g@capitalisefmtwords#1}%
312 {\@@s@cs@ng@capitalisefmtwords#1}%
313 }
```

italisefmtwords

```
314 \def\@s@cs@g@capitalisefmtwords#1#2#3\@nil{%
315 #1{\capitalisewords{#2}}%
316 \ifblank{#3}%
317 {}%
318 {%
319 \def\gls@add@space{}%
320 \let\mfu@checkword\mfu@checkword
321 \mfu@capitalisewords#3 \@nil\mfu@endcap
322 }%
323 }
```

italisefmtwords

```
324 \def\@s@cs@g@capitalisefmtwords#1#2\@nil{%
325 \capitalisewords{#1#2}%
326 }
```

italisefmtwords Unstarred form.

```
327 \newcommand*\@ns@capitalisefmtwords}[1]{%
328 \let\mfu@checkword@gobble
329 \def\gls@add@space{}%
330 \@capitalisefmtwords#1 \@mfu@nil{} \unskip
331 }
```

italisefmtwords Take care of group at the start of the word.

```
332 \newcommand*\@capitalisefmtwords{%
333 \@ifnextchar\bgroup{\@capitalise@g@fmtwords}{\@capitalise@ng@fmtwords}%
334 }
335 \newcommand*\@capitalise@g@fmtwords}[1]{%
336 \@capitalise@ng@fmtwords{#1}}%
337 }
338 \def\mfu@ifnil#1#2\mfu@end@nil#3#4{%
339 \def\gls@argi{#1}%
340 \ifx\gls@argi\mfu@nnil
341 #3%
342 \else
343 #4%
344 \fi
345 }
346 \def\@capitalise@ng@fmtwords#1 {%
347 \mfu@ifnil#1\relax\relax\mfu@end@nil
348 {%
349 \def\mfu@capfmtwordsnext{#1}%
350 }%
351 {%
352 \def\mfu@capfmtwordsnext{%
353 \@capitalise@fmtwords#1\mfu@nil
354 \def\gls@add@space{ }%
355 \let\mfu@checkword\mfu@checkword
```

```

356     \@@cap@nofmtphrase}%
357 }%
358 \@mfu@capfmtwordsnext
359 }%

```

capitalise@fmtwords

```

360 \def\@capitalise@fmtwords#1{%
361 \@mfu@ifnil#1\relax\relax\@mfu@end@nil
362 {%
363     Do nothing.
364     \def\@mfu@capfmtwordsnext{#1}%
365     }%
366     \def\gls@argi{#1}%
367     \ifx\gls@argi\@MFU@protect
368     Skip \protect
369     \def\@mfu@capfmtwordsnext{\@capitalise@fmtwords}%
370     \else
371     \def\@gls@tmp{\ #1}%
372     \@onelevel@sanitize\@gls@tmp
373     \expandafter\@gls@checkcs\@gls@tmp\relax\relax
374     \if@glscs
375     \def\@mfu@capfmtwordsnext{\@capitalise@fmtwords#1}%
376     \else
377     \ifx\gls@argi\@empty
378     \def\@mfu@capfmtwordsnext{\@@capnofmtphrase{}}%
379     \else
380     \def\@mfu@capfmtwordsnext{\@@capnofmtphrase#1}%
381     \fi
382     \fi
383     }%
384 \@mfu@capfmtwordsnext
385 }

```

capitalise@fmtwords Check for a group following argument.

```

386 \def\@capitalise@fmtwords#1{%
387 \@ifnextchar\bgroup{\@capfmtsubphrase#1}{\@@capnofmtphrase#1}%
388 }

```

@capnofmtphrase

```

389 \def\@@capnofmtphrase{%
390 \@ifnextchar\bgroup{\@cap@g@nofmtphrase}{\@@cap@ng@nofmtphrase}%
391 }

```

@ng@nofmtphrase

```

392 \def\@@cap@ng@nofmtphrase#1\@mfu@nil{%
393 \gls@add@space

```

```

394 \mfu@mid@capitalisewords{#1}%
395 \def\gls@add@space{ }%
396 }

```

cap@nofmtphrase

```

397 \def\@cap@nofmtphrase#1{%
398 \@cap@ng@nofmtphrase{#1}}%
399 }

```

mfu@terminator

```

400 \newcommand*\mfu@terminator{\hbox{}}

```

capfmtsubphrase

```

401 \def\@capfmtsubphrase#1#2#3{%
    Protect any trailing space at the end of the second argument.
402 #1{\@capitalisefmtwords#2{\mfu@terminator} \mfu@nil{} \unskip}%
403 \def\gls@add@space{ }%
404 \let\mfu@checkword\mfu@checkword
405 \mfu@ifnil#1\relax\relax\mfu@end@nil
406 {\let\mfu@fmtsubphrasenext\relax}%
407 {\def\mfu@fmtsubphrasenext{#3}}%
408 \mfu@fmtsubphrasenext
409 }

```

cap@nofmtphrase

```

410 \def\@cap@nofmtphrase#1 {%
411 \mfu@ifnil#1\relax\relax\mfu@end@nil
412 {%
413 \def\mfu@capnofmtwordsnext{#1 }%
414 }%
415 {%
416 \def\mfu@capnofmtwordsnext{%
417 \@capitalisefmtwords#1 \mfu@nil{} \unskip
418 \def\gls@add@space{ }%
419 \let\mfu@checkword\mfu@checkword
420 \@capitalisefmtwords
421 }%
422 }%
423 \mfu@capnofmtwordsnext
424 }

```

italisefmtwords Short-cut command:

```

425 \newcommand*\xcapitalisefmtwords{%
426 \@ifstar\@s@xcapitalisefmtwords\@ns@xcapitalisefmtwords
427 }

```

italisefmtwords Starred version

```

428 \newcommand*\@s@xcapitalisefmtwords}[1]{%

```

```
429 \expandafter\@s@capitalisefmtwords\expandafter{#1}%
430 }
```

italisefmtwords Unstarred version

```
431 \newcommand*\@ns@xcapitalisefmtwords}[1]{%
432 \expandafter\@ns@capitalisefmtwords\expandafter{#1}%
433 }
```

italisefmtwords

```
434 \newrobustcmd*\@ecapitalisefmtwords}{%
435 \@ifstar\@s@ecapitalisefmtwords\@ns@ecapitalisefmtwords
436 }
```

italisefmtwords Starred version

```
437 \newcommand*\@s@ecapitalisefmtwords}[1]{%
438 \protected@edef\@MFU@caparg{#1}%
439 \expandafter\@s@capitalisefmtwords\expandafter{\@MFU@caparg}%
440 }
```

italisefmtwords Unstarred version

```
441 \newcommand*\@ns@ecapitalisefmtwords}[1]{%
442 \protected@edef\@MFU@caparg{#1}%
443 \expandafter\@ns@capitalisefmtwords\expandafter{\@MFU@caparg}%
444 }
```

2.2 Mfirstuc-english Documented Code

```
445 \NeedsTeXFormat{LaTeX2e}
446 \ProvidesPackage{mfirstuc-english}[2021/10/15 v2.07 (NLCT)]
```

Load mfirstuc if not already loaded:

```
447 \RequirePackage{mfirstuc}
```

Add no-cap words. This list isn't complete. I don't intend adding disputed words. Additional words can be added to the document in a similar manner.

```
448 \MFUnocap{a}
449 \MFUnocap{an}
450 \MFUnocap{and}
451 \MFUnocap{but}
452 \MFUnocap{for}
453 \MFUnocap{in}
454 \MFUnocap{of}
455 \MFUnocap{or}
456 \MFUnocap{no}
457 \MFUnocap{nor}
458 \MFUnocap{so}
459 \MFUnocap{some}
460 \MFUnocap{the}
461 \MFUnocap{with}
462 \MFUnocap{yet}
```

Change History

1.03 (2008/12/22)	\@mfu@mid@capitalisewords: new 6
\@makefirststuc: changed	\@capitalisefmtwords: new 9
\protected@edef to \def 2	\@ecapitalisefmtwords: new 13
1.05 (2011/04/02)	\@ifMFUhyphen: new 6
\glsmakefirststuc: new 5	\@MFUapplytofirst: new 4
1.06 (2012/05/21)	\@MFUcapword: new 7
General: now requires etoolbox 1	\@xcapitalisefmtwords: new 13
\@capitalisewords: new 6	2.04 (2016-07-31)
\@xcapitalisewords: new 9	\@@cap@g@nofmtphrase: new 12
1.08 (2013-11-04)	\@@cap@ng@nofmtphrase: new 12
\@capitalisewords: made robust 6	\@@s@capitalisefmtwords: new 10
\@makefirststuc: made robust 1	\@@s@checkgrp@capitalisefmtwords:
1.09 (2017-07-30)	new 9
\@mfu@nocaplist: new 9	\@@s@cs@capitalisefmtwords: new .. 10
\@capitalisewords: added check for	\@@s@cs@g@capitalisefmtwords: new 10
words that shouldn't be capitalised ... 6	\@@s@cs@ng@capitalisefmtwords:
\@gMFUocap: new 9	new 11
\@mfu@checkword: new 8	\@@s@noc@capitalisefmtwords: new 10
\@MFUclear: new 9	\@gls@split: added missing false part .. 4
\@MFUocap: new 9	\@mfu@terminator: new 12
1.10 (2015/05/03)	\@ns@capitalisefmtwords: new 11
\@ecapitalisewords: new 9	\@ns@ecapitalisefmtwords: new 14
\@emakefirststuc: new 6	\@ns@xcapitalisefmtwords: new 13
2.0 (2015/09/09)	\@s@capitalisefmtwords: new 9
\@makefirststuc: added check for two	\@s@ecapitalisefmtwords: new 14
control sequences 2	\@s@xcapitalisefmtwords: new 13
General: package split from glossaries 1	2.05 (2017-11-10)
2.01 (2015-12-08)	\@mfu@applytofirst: grouped first
\@makefirststuc: added check for	argument 4
\@protect 2	2.06 (2017-11-14)
2.02 (2015-12-17)	\@makefirststuc: added braces to
\@mfu@checkword: modified to use loop	\@glsmakefirststuc argument 3
instead of \@ifinlist 8	check for inner control sequence 3
\@mfu@checkword@do: new 8	2.07 (2021-10-15)
2.03 (2016-07-29)	\@MFU@do@capword: new 6
\@@cap@nofmtphrase: new 13	\@MFU@dofirst@capword: new 7
\@@capfntsubphrase: new 13	\@MFU@firstwordbreakcapword: new .. 8
\@@capitalisefmtwords: new 12	\@MFU@p@makefirststuc: check for
\@@capnofmtphrase: new 12	\@MFU@skippunc 2
\@MFU@p@makefirststuc: changed test 2	\@MFU@skippunc: new 2
\@MFU@protect: new 2	\@MFU@wordbreakcapword: new 8
\@MFU@capword: new 7	\@MFU@firstwordbreak: new 8
\@capitalise@fmtwords: new 11	\@MFU@wordbreak: new 7
\@capitalisefmtwords: new 11	\@MFU@capwordfirststuc: new 7
\@gls@split: added check for	\@MFU@hyphencapword: new 7
datatool-base UTF-8 support 3	\@MFU@skippunc: new 2
\@mfu@applytofirst: new 4	\@MFU@wordbreak: new 7

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

Symbols		
\@cap@g@nofmtphrase	12	\@gls@makefirstuc 5
\@cap@ng@nofmtphrase	12	\@gls@rest 3-5
\@cap@nofmtphrase	11	\@gls@split 3
\@capfmtsubphrase	12	\@gls@tmp 2, 3, 10, 12
\@capitalisefmtwords	12	\@gls@false 2, 4
\@capnofmtphrase	12	\@gls@true 2, 4
\@s@capitalisefmtwords	9	\@glsmfirst 1, 3
\@s@checkgrp@capitalisefmtwords	9	\@glsmrest 1, 3
\@s@cs@capitalisefmtwords	10	\@gobble 2, 6, 11
\@s@cs@g@capitalisefmtwords	10	\@ifnextchar 9-12
\@s@cs@ng@capitalisefmtwords	10	\@ifstar 9, 13
\@s@noc@capitalisefmtwords	9, 10	\@makefirstuc 2
\MFU@caparg	6, 9, 14	\@mfu@applytofirst 5
\MFU@cappingword	6-8	\@mfu@argii 7
\MFU@do@capword	6, 8	\@mfu@argiii 7, 8
\MFU@dofirst@capword	6	\@mfu@capfmtwordsnext 10-12
\MFU@firstwordbreakcapword	8	\@mfu@capnofmtwordsnext 13
\MFU@p@makefirstuc	2	\@mfu@checkword 6, 10, 11, 13
\MFU@protect	2, 10, 12	\@mfu@domakefirstuc 6, 8
\MFU@skippunc	2	\@mfu@end@applytofirst 4, 5
\MFU@wordbreakcapword	7	\@mfu@end@nil 11, 13
\MFUcapword	7	\@mfu@fntsubphrasenext 13
\MFUfirstwordbreak	7	\@mfu@ifnil 11, 13
\MFUwordbreak	6	\@mfu@mid@capitalisewords 12
\@capitalise@fntwords	11	\@mfu@nil 9, 11-13
\@capitalise@g@fntwords	11	\@mfu@nnil 11
\@capitalise@ng@fntwords	11	\@mfu@nocaplist 8, 9
\@capitalisefmtwords	11, 13	\@mfu@terminator 8, 13
\@empty	2, 3, 12	\@nll 3-7, 9-11
\@endMFU@protected	2	\@nll 6-8, 10
\@endMFUcapword	7	\@ns@capitalisefmtwords 9, 13, 14
\@endMFUfirstwordbreak	7, 8	\@ns@ecapitalisefmtwords 13
\@endMFUwordbreak	6, 7	\@ns@xcapitalisefmtwords 13
\@firstofone	8	\@onelevel@sanitize 2, 3, 10, 12
\@gls@argi	4	\@s@capitalisefmtwords 9, 13, 14
\@gls@argii	4	\@s@ecapitalisefmtwords 13
\@gls@body	3, 5	\@s@xcapitalisefmtwords 13
\@gls@checkcs	2-4, 10, 12	
\@gls@domfirstuc	3	
\@gls@first	3, 4	
\@gls@getbody	3, 5	
\@gls@gobbletonil	5	
		B
		\bgroup 9-12

C		M	
<code>\capitalisewords</code>	9-11	<code>\makefirstuc</code>	3, 5-7
D		<code>\MakeUppercase</code>	5
<code>datatool-base package</code>	4, 5	<code>mfirstuc package</code>	14
<code>\def</code>	2-8, 10-13	<code>\mfirstucMakeUppercase</code>	5
<code>\dtl@end@if@two@octets</code>	3, 4	<code>\mfu@cap@first</code>	6
<code>\dtl@first</code>	3, 4	<code>\mfu@cap@second</code>	6
<code>\dtl@getfirst@UTFviii</code>	3-5	<code>\mfu@capitalisewords</code>	6, 10
<code>\dtl@if@two@octets</code>	3, 4	<code>\mfu@checkword</code>	6, 8, 10, 11, 13
<code>\dtl@rest</code>	3, 4	<code>\mfu@checkword@arg</code>	8
E		<code>\mfu@checkword@argii</code>	8
<code>\edef</code>	3	<code>\mfu@checkword@do</code>	8
<code>\else</code>	2-4, 6-8, 10-12	<code>\mfu@endcap</code>	6, 10
<code>\end@dtl@getfirst@UTFviii</code>	3, 4	<code>\mfu@noop</code>	6
<code>etoolbox package</code>	1, 8	<code>\MFUapplytofirst</code>	5
<code>\expandafter</code>	2-6, 9, 10, 12-14	<code>\MFUcapword</code>	6, 7
<code>\expandonce</code>	3	<code>\MFUcapwordfirstuc</code>	7
F		<code>\MFUhyphencapword</code>	7
<code>\fi</code>	2-4, 6-8, 10-12	<code>\MFUhyphenfalse</code>	6
<code>\forlistloop</code>	8	<code>\MFUuncap</code>	14
G		<code>\MFUskipunc</code>	2
<code>\gls@add@space</code>	6, 10-13	<code>\MFUwordbreak</code>	6-8
<code>\gls@argi</code>	2, 10-12	N	
<code>\glsmakefirstuc</code>	3	<code>\NeedsTeXFormat</code>	1, 14
H		<code>\newcommand</code>	2, 5-14
<code>\hbox</code>	12	<code>\newif</code>	1, 6
I		<code>\newrobustcmd</code>	1, 2, 6, 7, 9, 13
<code>\if@glscs</code>	1-3, 10, 12	<code>\newtoks</code>	1
<code>\ifblank</code>	5, 10	<code>\next@mfu@cap</code>	6
<code>\ifbool</code>	3, 5	<code>\noexpand</code>	3
<code>\ifdef</code>	3-5	P	
<code>\ifdefequal</code>	8	<code>\protect</code>	2
<code>\ifMFUhyphen</code>	7	<code>\protected@edef</code>	6, 9, 14
<code>\ifx</code>	2-4, 6-8, 10-12	<code>\ProvidesPackage</code>	1, 14
L		R	
<code>\let</code>	3, 6-8, 10, 11, 13	<code>\relax</code>	2-5, 10-13
<code>\listadd</code>	9	<code>\renewcommand</code>	9
<code>\listbreak</code>	8	<code>\RequirePackage</code>	1, 14
<code>\listgadd</code>	9	T	
U		<code>\the</code>	3
U		U	
<code>\unskip</code>	9, 11, 13		