

Documented Code For `mfirstuc` v2.06

Nicola L.C. Talbot

Dickimaw Books

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

2017-11-14

Contents

1	Introduction	1
2	Main Package Code	1
2.1	Mfirstuc Documented Code	1
2.2	Mfirstuc-english Documented Code	13
	Change History	13
	Change History	13
	Index	14

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}[2017/11/14 v2.06 (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   \OMFU@p@makefirstuc#1\relax\relax\relax\end@MFU@protected{#1}%
9 }

```

`\OMFU@protect`

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

`U@p@makefirstuc`

Internal command to do the required action.

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

Test if the first argument is `\protect`.

```

12 \def\gls@argi{#1}%
13 \ifx\gls@argi\OMFU@protect
14   \OMakefirstuc{#2#3}%
15 \else
16   \OMakefirstuc{#4}%
17 \fi
18 }

```

`\OMakefirstuc`

Internal command to do the required action.

```

19 \newcommand*\OMakefirstuc}[1]{%
20   \def\gls@argi{#1}%
21   \ifx\gls@argi\empty

```

If the argument is empty, do nothing.

```

22 \else
23   \def\@glscs@tmp{\ #1}%
24   \Onelevel@sanitize\@glscs@tmp
25   \expandafter\gls@checkcs\@glscs@tmp\relax\relax
26   \if@glscs

```

Check if we have two sequential control sequences.

```

27   \expandafter\def\expandafter\gls@tmp\expandafter{\expandafter
28     \ \ @gobble #1}%
29   \Onelevel@sanitize\@glscs@tmp
30   \expandafter\gls@checkcs\@glscs@tmp\relax\relax
31   \if@glscs\@glscsfalse\else\@glscstrue\fi
32   \fi
33   \if@glscs
34     \gls@getbody #1{}\@nil

```

```

35      \ifx\@gls@rest\empty
36          \glsmakefirstuc{\#1}%
37      \else
38          \expandafter\@gls@split\@gls@rest\@nil
39          \ifx\@gls@first\empty
40              \glsmakefirstuc{\#1}%
41          \else
42              \edef\@gls@tmp\expandafter{\noexpand\ \expandonce\@gls@first}%
43                  \onelevel@sanitize\@gls@tmp
44                  \expandafter\@gls@checkcs\@gls@tmp\relax\relax
45                  \if@gls@scs
46                      \edef\@gls@domfirstuc{\expandonce\@gls@body
47                          {\noexpand\makefirstuc{\expandonce\@gls@first
48                              \expandonce\@gls@rest}}%
49                      }%
50                  }%
51              \else
52                  \expandafter\@gls@first\expandafter{\@gls@first}%
53                  \expandafter\@gls@rest\expandafter{\@gls@rest}%
54
55                  \edef\@gls@domfirstuc{\noexpand\@gls@body
56                      {\noexpand\glsmakefirstuc{\the\@gls@first}}%
57                      \the\@gls@rest}%
58                  \fi
59                  \@gls@domfirstuc
60              \fi
61          \else
62              \glsmakefirstuc{\#1}%
63          \fi
64      \fi
65 }

```

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

```

66 \ifdef\dtl@getfirst@UTFviii
67 {%
68     \def\@gls@split{\#1\#2\@nil}%
69     \ifbool{\dtl@utf8}{%
70         {%
71             \expandafter\dtl@if@two@octets{\#1\#2}\relax\dtl@end@if@two@octets
72         }%
73             \dtl@getfirst@UTFviii{\#1\#2\@nil}\end@dtl@getfirst@UTFviii
74             \let\@gls@first\dtl@first
75             \let\@gls@rest\dtl@rest
76         }%
77         {%
78             \def\@gls@first{\#1}\def\@gls@rest{\#2}%
79         }%

```

```

80      }%
81      {%
82          \def\@gls@first{\#1}\def\@gls@rest{\#2}%
83      }%
84  }
85 }%
86 {%
87  \def\@gls@split#1#2\@nil{%
88      \def\@gls@first{\#1}\def\@gls@rest{\#2}%
89  }
90 }

91 \def\@gls@checkcs#1 #2#3\relax{%
92  \def\@gls@argi{\#1}\def\@gls@argii{\#2}%
93  \ifx\@gls@argi\@gls@argii
94      \@glscstrue
95  \else
96      \@glscsfals
97  \fi
98 }

```

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

```

99 \ifdef\dtl@getfirst@UTFviii
100 {%
101  \def\@mfu@applytofirst#1#2\@mfu@end@applytofirst#3{%
102      \expandafter\dtl@if@two@octets#1#2\relax\dtl@end@two@octets
103      {%
104          \dtl@getfirst@UTFviii#1#2\@nil\end@dtl@getfirst@UTFviii
105          \expandafter#3\expandafter{\dtl@first}\dtl@rest
106      }%
107      {%

```

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

```

108      #3{\#1}#2%
109      }%
110  }%
111 }%
112 {%

```

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

```

113  \def\@mfu@applytofirst#1#2\@mfu@end@applytofirst#3{\#3{\#1}#2}
114 }

```

\MFUapplytofirst \MFUapplytofirst{\langle cs \rangle}{\langle text \rangle}

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.

```
115 \ifdef\dtl@getfirst@UTFviii
116 {%
117   \newcommand*\MFUapplytofirst}[2]{%
118     \ifbool{@dtl@utf8}%
119     {%
120       \ifblank{#2}%
121       {#1{#2}}%
122     {%
123       \ifblank{#1}%
124       {#1#2}%
125     {%
126       \cmfu@applytofirst#2\cmfu@end@applytofirst#1\relax
127     }%
128   }%
129 }%
130 {%
131   #1#2%
132 }%
133 }%
134 }%
135 {%
136   \newcommand*\MFUapplytofirst}[2]{#1#2}
137 }
```

`\gls@makefirsttuc` Make first thing upper case:

```
138 \def\@gls@makefirsttuc#1{\MFUapplytofirst\mfirsttucMakeUppercase{#1}}
```

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

```
139 \newcommand*\mfisttucMakeUppercase}{\MakeUppercase}
```

`\glsmakefirsttuc` Provide a user command to make it easier to customise.

```
140 \newcommand*\glsmakefirsttuc}[1]{\@gls@makefirsttuc{#1}}
```

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

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

Scoup up everything to `\@nil` and store in `\@gls@rest`:

```
142 \def\@gls@gobbletonil#1\@nil{\def\@gls@rest{#1}}
```

`\xmakefirsttuc` Expand argument once before applying `\makefirsttuc` (added v1.01).

```
143 \newcommand*\xmakefirsttuc}[1]{%
144   \expandafter\makefirsttuc\expandafter{#1}%
145 }
```

`\emakefirsttuc` Fully expand argument before applying `\makefirsttuc`

```
146 \newrobustcmd*\emakefirsttuc}[1]{%
```

```

147 \protected@edef\@MFU@caparg{#1}%
148 \expandafter\makefirstuc\expandafter{\@MFU@caparg}%
149 }

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).
150 \newrobustcmd*\capitalisewords}[1]{%
151   \let\@mfu@checkword\gobble
152   \def\gls@add@space{}%
153   \let\@mfu@domakefirstuc\MFUcapword
154   \mfu@capitalisewords#1 \@nil\mfu@endcap
155 }

capitalisewords \@mfu@checkword needs to be set before use.
156 \newcommand*\@mfu@mid@capitalisewords}[1]{%
157   \def\gls@add@space{}%
158   \let\@mfu@domakefirstuc\MFUcapword
159   \mfu@capitalisewords#1 \@nil\mfu@endcap
160 }

161 \def\mfu@capitalisewords#1 #2\mfu@endcap{%
162   \def\mfu@cap@first{#1}%
163   \def\mfu@cap@second{#2}%
164   \gls@add@space
165   \@mfu@checkword{#1}%
166   \@mfu@domakefirstuc{#1}%
167   \def\gls@add@space{}%
168   \ifx\mfu@cap@second\@nil
169     \let\next@\mfu@cap\mfu@noop
170   \else
171     \let\next@\mfu@cap\mfu@capitalisewords
172     \let\@mfu@checkword\mfu@checkword
173   \fi
174   \next@\cap#2\mfu@endcap
175 }
176 \def\mfu@noop#1\mfu@endcap{}

\ifMFUhyphen Split on hyphens.
177 \newif\ifMFUhyphen
178 \MFUhyphenfalse

\MFUcapword
179 \newcommand*\MFUcapword}[1]{%
180   \ifMFUhyphen
181     \@MFUcapword#1-\@nil\@endMFUcapword
182   \else
183     \makefirstuc{#1}%
184   \fi
185 }

```

```

\@MFUcapword
186 \def\@MFUcapword#1-#2\@endMFUcapword{%
187   \makefirststuc{#1}%
188   \def\@mfu@argii{#2}%
189   \ifx\@mfu@argii\@nnil
190   \else
191     -\@MFUcapword#2\@endMFUcapword
192   \fi
193 }

\@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.
194 \newcommand*\@mfu@checkword[1]{%
195   \def\@mfu@checkword@arg{#1}%
196   \let\@mfu@domakefirststuc\@MFUcapword
197   \forlistloop\@mfu@checkword@do\@mfu@nocaplist
198 }

\@mfu@checkword@do Handler for loop used by \@mfu@checkword
199 \newcommand*\@mfu@checkword@do}[1]{%
200   \def\@mfu@checkword@argii{#1}%
201   \ifdefequal\@mfu@checkword@arg\@mfu@checkword@argii
202   {%
203     \let\@mfu@domakefirststuc\@firstofone
204     \listbreak
205   }%
206   {%
207     Need to check if \@mfu@terminator has been tacked onto the end.
208     \def\@mfu@checkword@argii{#1{\@mfu@terminator}}%
209     \ifdefequal\@mfu@checkword@arg\@mfu@checkword@argii
210     {%
211       \let\@mfu@domakefirststuc\@firstofone
212       \listbreak
213     }%
214   }%
215 }

\@mfu@nocaplist List of words that shouldn't be capitalised.
216 \newcommand*\@mfu@nocaplist{}}

\@MFUnocap Provide the user with a means to add a word to the list. (No expansion performed on the
argument.)
217 \newcommand*\@MFUnocap[1]{\listadd{\@mfu@nocaplist}{#1}}

\g@MFUnocap Global version.
218 \newcommand*\g@MFUnocap[1]{\listgadd{\@mfu@nocaplist}{#1}}

```

```

\MFUclear Clear the list
219 \newcommand*\{\MFUclear\}{\renewcommand*\{@mfu@nocaplist\}\{}}

capitalisewords Short-cut command:
220 \newcommand*\xcapitalisewords[1]{%
221   \expandafter\capitalisewords\expandafter{#1}%
222 }

capitalisewords Fully expand argument before applying \capitalisewords
223 \newrobustcmd*\ecapitalisewords[1]{%
224   \protected@edef\@MFU@caparg{#1}%
225   \expandafter\capitalisewords\expandafter{\@MFU@caparg}%
226 }

\@mfu@nil
227 \newcommand\@mfu@nil{\unskip}

\@mfu@nnil
228 \newcommand*\@mfu@nnil{\@mfu@nil}

italisefmtwords Like \capitalisewords but assumes that the phrase contains formatting text-block commands.
229 \newcommand*\capitalisefmtwords{%
230   \ifstar@s@capitalisefmtwords@ns@capitalisefmtwords
231 }

italisefmtwords Starred form.
232 \newcommand*\@s@capitalisefmtwords[1]{%
233   \@@s@checkgrp@capitalisefmtwords#1\@nil
234 }

italisefmtwords
235 \newcommand*\@@s@checkgrp@capitalisefmtwords{%
236   \ifnextchar\bgroup{\@@s@no@cs@capitalisefmtwords}{\@@s@capitalisefmtwords}%
237 }

italisefmtwords Argument starts with a group.
238 \def\@@s@no@cs@capitalisefmtwords#1\@nil{%
239   \capitalisewords{#1}%
240 }

italisefmtwords
241 \newcommand*\@@s@capitalisefmtwords[1]{%
242   \def\gls@argi{#1}%
243   \ifx\gls@argi\@MFU@protect

```

```

Skip \protect
244   \let\@mfu@capfmtwordsnext\@s@capitalisefmtwords
245   \else
246     \ifx\gls@argi\@nnil
Nothing to do.
247   \let\@mfu@capfmtwordsnext\relax
248   \else
249     \def\gls@tmp{\ #1}%
250     \onelevel@sanitize\gls@tmp
251     \expandafter\gls@checkcs\gls@tmp\relax\relax
252     \if@gls@scs
253       \def\@mfu@capfmtwordsnext{\@s@cs@capitalisefmtwords#1}%
254     \else
255       \def\@mfu@capfmtwordsnext{\@s@nocs@capitalisefmtwords#1}%
256     \fi
257   \fi
258 \fi
259 \@mfu@capfmtwordsnext
260 }

italisefmtwords
261 \newcommand*{\@s@cs@capitalisefmtwords}[1]{%
262   \ifnextchar\bgroup
263     {\@s@cs@g@capitalisefmtwords#1}%
264     {\@s@cs@ng@capitalisefmtwords#1}%
265 }

italisefmtwords
266 \def\@s@cs@g@capitalisefmtwords#1#2#3\@nil{%
267   #1{\capitalisewords{#2}}%
268   \ifblank{#3}%
269   {}%
270   {}%
271   \def\gls@add@space{}%
272   \let\@mfu@checkword\mfu@checkword
273   \mfu@capitalisewords#3 \@nil\mfu@endcap
274 }%
275 }

italisefmtwords
276 \def\@s@cs@ng@capitalisefmtwords#1#2\@nil{%
277   \capitalisewords{#1#2}%
278 }

italisefmtwords Unstarred form.
279 \newcommand*{\@ns@capitalisefmtwords}[1]{%
280   \let\@mfu@checkword\@gobble
281   \def\gls@add@space{}%

```

```

282  \@capitalisefmtwords#1 \@mfu@nil{} \unskip
283 }

italisefmtwords Take care of group at the start of the word.
284 \newcommand*{\@capitalisefmtwords}{%
285   \@ifnextchar\bgroup{\@capitalise@g@fmtwords}{\@capitalise@ng@fmtwords}%
286 }
287 \newcommand*{\@capitalise@g@fmtwords}[1]{%
288   \@capitalise@ng@fmtwords{{#1}}%
289 }
290 \def\@mfu@ifnil#1#2\@mfu@end@nil#3#4{%
291   \def\gls@argi{{#1}}%
292   \ifx\gls@argi\@mfu@nnil
293     #3%
294   \else
295     #4%
296   \fi
297 }
298 \def\@capitalise@ng@fmtwords#1 {%
299   \@mfu@ifnil#1\relax\relax\@mfu@end@nil
300   {%
301     \def\@mfu@capfmtwordsnext{{#1}}%
302   }%
303   {%
304     \def\@mfu@capfmtwordsnext{%
305       \@capitalise@fmtwords#1\@mfu@nil
306       \def\gls@add@space{ }%
307       \let\@mfu@checkword\mfu@checkword
308       \@@cap@nofmtphrase}%
309   }%
310   \@mfu@capfmtwordsnext
311 }%

```

talise@fmtwords

```

312 \def\@capitalise@fmtwords#1{%
313   \@mfu@ifnil#1\relax\relax\@mfu@end@nil
314   {%

```

Do nothing.

```

315   \def\@mfu@capfmtwordsnext{{#1}}%
316 }%
317 {%
318   \def\gls@argi{{#1}}%
319   \ifx\gls@argi\@MFU@protect

```

Skip \protect

```

320   \def\@mfu@capfmtwordsnext{\@capitalise@fmtwords}%
321   \else
322     \def\@gls@tmp{\ #1}%
323     \onelevel@sanitize@\gls@tmp

```

```

324     \expandafter\@gls@checkcs\@gls@tmp\relax\relax
325     \if@glscs
326         \def\@mfu@capfmtwordsnext{\@capitalisefmtwords#1}%
327     \else
328         \ifx\gls@argi\@empty
329             \def\@mfu@capfmtwordsnext{\@capnofmtphrase{} }%
330         \else
331             \def\@mfu@capfmtwordsnext{\@capnofmtphrase#1}%
332         \fi
333     \fi
334 \fi
335 }%
336 \@mfu@capfmtwordsnext
337 }

```

`italisefmtwords` Check for a group following argument.

```

338 \def\@capitalisefmtwords#1{%
339   \@ifnextchar\bgroup{\@capfmtsubphrase#1}{\@capnofmtphrase#1}%
340 }

```

`@capnofmtphrase`

```

341 \def\@capnofmtphrase{%
342   \ifnextchar\bgroup{\@cap@g@nofmtphrase}{\@cap@ng@nofmtphrase}%
343 }

```

`@ng@nofmtphrase`

```

344 \def\@cap@ng@nofmtphrase#1\@mfu@nil{%
345   \gls@add@space
346   \@mfu@mid@capitalisewords{#1}%
347   \def\gls@add@space{ }%
348 }

```

`p@g@nofmtphrase`

```

349 \def\@cap@g@nofmtphrase#1{%
350   \@cap@ng@nofmtphrase{{#1}}%
351 }

```

`@mfu@terminator`

```

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

```

`capfmtsubphrase`

```

353 \def\@capfmtsubphrase#1#2#3{%

```

Protect any trailing space at the end of the second argument.

```

354 #1{\@capitalisefmtwords#2{\@mfu@terminator} \@mfu@nil{} \unskip}%
355 \def\gls@add@space{ }%
356 \let\@mfu@checkword\@mfu@checkword
357 \@mfu@ifnil#1\relax\relax\@mfu@end@nil
358 {\let\@mfu@fmtsubphrasenext\relax}%

```

```

359  {\def\@mfu@fmtsubphrasenext{#3}}%
360  \@mfu@fmtsubphrasenext
361 }

cap@nofmtphrase
362 \def\@@cap@nofmtphrase#1 {%
363   \@mfu@ifnil#1\relax\relax\@mfu@end@nil
364   {%
365     \def\@mfu@capnofmtwordsnext{#1 }%
366   }%
367   {%
368     \def\@mfu@capnofmtwordsnext{%
369       @capitalisefmtwords#1 \@mfu@nil{} \unskip
370       \def\gls@add@space{ }%
371       \let\@mfu@checkword\@mfu@checkword
372       @capitalisefmtwords
373     }%
374   }%
375   \@mfu@capnofmtwordsnext
376 }

italisefmtwords Short-cut command:
377 \newcommand*{\xcapitalisefmtwords}{%
378   \@ifstar@s\xcapitalisefmtwords@ns\xcapitalisefmtwords
379 }

italisefmtwords Starred version
380 \newcommand*{\@s\xcapitalisefmtwords}[1]{%
381   \expandafter@s\xcapitalisefmtwords\expandafter{#1}%
382 }

italisefmtwords Unstarred version
383 \newcommand*{\@ns\xcapitalisefmtwords}[1]{%
384   \expandafter@ns\xcapitalisefmtwords\expandafter{#1}%
385 }

italisefmtwords
386 \newrobustcmd*{\ecapitalisefmtwords}{%
387   \@ifstar@s\ecapitalisefmtwords@ns\ecapitalisefmtwords
388 }

italisefmtwords Starred version
389 \newcommand*{\@s\ecapitalisefmtwords}[1]{%
390   \protected@edef\@MFU@caparg{#1}%
391   \expandafter@s\xcapitalisefmtwords\expandafter{\@MFU@caparg}%
392 }

```

italisefmtwords

Unstarred version

```
393 \newcommand*{\@ns@ecapitalisefmtwords}[1]{%
394   \protected@edef\@MFU@caparg{#1}%
395   \expandafter\@ns@capitalisefmtwords\expandafter{\@MFU@caparg}%
396 }
```

2.2 Mfirstuc-english Documented Code

```
397 \NeedsTeXFormat{LaTeX2e}
398 \ProvidesPackage{mfirstuc-english}[2017/11/14 v2.06 (NLCT)]
```

Load mfirstuc if not already loaded:

```
399 \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.

```
400 \MFUnocap{a}
401 \MFUnocap{an}
402 \MFUnocap{and}
403 \MFUnocap{but}
404 \MFUnocap{for}
405 \MFUnocap{in}
406 \MFUnocap{of}
407 \MFUnocap{or}
408 \MFUnocap{no}
409 \MFUnocap{nor}
410 \MFUnocap{so}
411 \MFUnocap{some}
412 \MFUnocap{the}
413 \MFUnocap{with}
414 \MFUnocap{yet}
```

Change History

1.03 (2008/12/22)	\capitalisewords: added check for words that shouldn't be capitalised 6
	\protected@edef to \def 2
1.05 (2011/04/02)	\glsmakefirststuc: new 7
	\mfu@checkword: new 7
1.06 (2012/05/21)	\MFUclear: new 8
	1.10 (2015/05/03)
	\ecapitalisewords: new 8
	\emakefirststuc: new 5
1.08 (2013-11-04)	2.0 (2015/09/09)
\capitalisewords: made robust 6	\@makefirststuc: added check for two control sequences 2
\makefirststuc: made robust 2	General: package split from glossaries 1
1.09 (2017-07-30)	2.01 (2015-12-08)
\mfu@nocaplist: new 7	\makefirststuc: added check for

\protect	2	2.04 (2016-07-31)
2.02 (2015-12-17)		
\mfu@checkword: modified to use loop		\@cap@g@nofmtphrase: new
instead of \ifinlist	7	11
\mfu@checkword@do: new	7	\@cap@ng@nofmtphrase: new
2.03 (2016-07-29)		\@cap@capitalisefmtwords: new
\@cap@nofmtphrase: new	12	8
\@capfmtsubphrase: new	11	\@cap@cs@capitalisefmtwords: new
\@cap@capitalisefmtwords: new	11	9
\@cap@nofmtphrase: new	11	\@cap@ng@capitalisefmtwords: new
\@MFU@p@makefirststuc: changed test	2	9
\@MFU@protect: new	2	\@gls@split: added missing false part ..
\@MFU@capword: new	7	4
\@capitalise@fmtwords: new	10	\@mfu@terminator: new
\@capitalisefmtwords: new	10	11
\@gls@split: added check for		\@ns@capitalisefmtwords: new
datatool-base UTF-8 support	3	9
\@mfu@applytofirst: new	4	\@ns@ecapitalisefmtwords: new
\@mfu@mid@capitalisewords: new	6	13
\@capitalisefmtwords: new	8	\@ns@xcapitalisefmtwords: new
\@capitalisefmtwords: new	12	12
\@capfmtsubphrase: new	6	\@s@capitalisefmtwords: new
\@ifMFUhyphen: new	6	8
\@MFUapplytofirst: new	4	\@s@ecapitalisefmtwords: new
\@MFUcapword: new	6	12
\@xcapitalisefmtwords: new	12	\@s@xcapitalisefmtwords: new
		12
		2.05 (2017-11-10)
		\@mfu@applytofirst: grouped first
		argument
		2.06 (2017-11-14)
		\@makefirststuc: added braces to
		\gls@makefirststuc argument
		3
		check for inner control sequence
		3

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	11	\@MFU@protect
\@cap@ng@nofmtphrase	11	2, 8, 10
\@cap@nofmtphrase	10	\@MFUcapword
\@capfmtsubphrase	11	6
\@capitalisefmtwords	11	\@capitalise@fmtwords
\@cap@nofmtphrase	11	10
\@s@capitalisefmtwords	8	\@capitalise@g@fmtwords
\@s@checkgrp@@capitalisefmtwords ...	8	10
\@s@cs@capitalisefmtwords	9	\@capitalise@ng@fmtwords
\@s@cs@g@capitalisefmtwords	9	10-12
\@s@cs@ng@capitalisefmtwords	9	\@empty
\@s@no@cs@capitalisefmtwords	8, 9	2, 3, 11
\@MFU@caparg	6, 8, 12, 13	\@end@MFU@protected
\@MFU@p@makefirststuc	2	2
		\@endMFUcapword
		6, 7
		\@firstofone
		7
		\@gls@argi
		4
		\@gls@argii
		4
		\@gls@body
		3, 5
		\@gls@checkcs
		2-4, 9, 11
		\@gls@domfirststuc
		3

\@gls@first	3, 4	D	
\@gls@getbody	2, 5	datatool-base package	4, 5
\@gls@gobbletonil	5	\def	2–12
\@gls@makefirststuc	5	\dtl@end@if@two@octets	3, 4
\@gls@rest	3–5	\dtl@first	3, 4
\@gls@split	3	\dtl@getfirst@UTFviii	3–5
\@gls@tmp	2, 3, 9–11	\dtl@if@two@octets	3, 4
\@glscsfalse	2, 4	\dtl@rest	3, 4
\@glscstrue	2, 4		
\@glsmfirst	2, 3	E	
\@glsmrest	2, 3	\edef	3
\@gobble	2, 6, 9	\else	2–4, 6, 7, 9–11
\@ifnextchar	8–11	\end@dtl@getfirst@UTFviii	3, 4
\@ifstar	8, 12	etoolbox package	1, 7
\@makefirststuc	2	\expandafter	2–6, 8, 9, 11–13
\@mfu@applytofirst	5	\expandonce	3
\@mfu@argii	7		
\@mfu@capfmtwordsnext	9–11	\fi	2–4, 6, 7, 9–11
\@mfu@capnofmtwordsnext	12	\forlistloop	7
\@mfu@checkword	6, 9–12		
\@mfu@domakefirststuc	6, 7	G	
\@mfu@end@applytofirst	4, 5	\gls@add@space	6, 9–12
\@mfu@end@nil	10–12	\gls@argi	2, 8–11
\@mfu@fmtsubphrasenext	11, 12	\glsmakefirststuc	3
\@mfu@ifnil	10–12		
\@mfu@mid@capitalisewords	11	H	
\@mfu@nil	8, 10–12	\hbox	11
\@mfu@nnil	10		
\@mfu@nocaplist	7, 8	I	
\@mfu@terminator	7, 11	\if@glscts	2, 3, 9, 11
\@nil	2–6, 8, 9	\ifblank	5, 9
\@nnil	6, 7, 9	\ifbool	3, 5
\@ns@capitalisefmtwords	8, 12, 13	\ifdef	3–5
\@ns@ecapitalisefmtwords	12	\ifdefequal	7
\@ns@xcapitalisefmtwords	12	\ifMFUhyphen	6
\@onelevel@sanitize	2, 3, 9, 10	\ifx	2–4, 6–11
\@s@capitalisefmtwords	8, 12		
\@s@ecapitalisefmtwords	12	L	
\@s@xcapitalisefmtwords	12	\let	3, 6, 7, 9–12
		\listadd	7
		\listbreak	7
		\listgadd	7
_	2, 3, 9, 10		
		M	
B		\makefirststuc	3, 5–7
\bgroup	8–11	\MakeUppercase	5
		mfirrstuc package	13
C		\mfirrstucMakeUppercase	5
\capitalisewords	8, 9	\mfu@cap@first	6
		\mfu@cap@second	6

\mfu@capitalisewords	6, 9	\next@mfu@cap	6
\mfu@checkword	6, 9–12	\noexpand	3
\mfu@checkword@arg	7		
\mfu@checkword@argii	7	P	
\mfu@checkword@do	7	\protect	2
\mfu@endcap	6, 9	\protected@edef	6, 8, 12, 13
\mfu@noop	6	\ProvidesPackage	1, 13
\MFUapplytofirst	5		
\MFUcapword	6, 7	R	
\MFUhyphenfalse	6	\relax	2–5, 9–12
\MFUnocap	13	\renewcommand	8
		\RequirePackage	1, 13
N			
\NeedsTeXFormat	1, 13		
\newcommand	2, 5–13	T	
\newif	2, 6	\the	3
\newrobustcmd	2, 5, 6, 8, 12		
\newtoks	2	U	
		\unskip	8, 10–12