

# The `magicnum` package

Heiko Oberdiek  
<heiko.oberdiek at gmail.com>

2011/04/10 v1.4

## Abstract

This package allows to access magic numbers by a hierarchical name system.

## Contents

<b>1</b>	<b>Documentation</b>	<b>2</b>
1.1	Introduction	2
1.2	User interface	2
1.2.1	<code>\magicnum</code>	2
1.2.2	Properties	3
1.3	Data	3
1.3.1	Category <code>tex.catcode</code>	3
1.3.2	Category <code>etex.grouptype</code>	3
1.3.3	Category <code>etex.iftyp</code>	4
1.3.4	Category <code>etex.nodetype</code>	4
1.3.5	Category <code>etex.interactionmode</code>	4
1.3.6	Category <code>luatex.pdfliteral.mode</code>	4
<b>2</b>	<b>Implementation</b>	<b>4</b>
2.1	Reload check and package identification	5
2.2	Catcodes	6
2.3	Check for previous definition	7
2.4	Without LuaTeX	7
2.5	With LuaTeX	7
2.6	Data	8
2.6.1	Plain data	8
2.6.2	Data for TeX	10
2.6.3	Lua module	12
<b>3</b>	<b>Test</b>	<b>15</b>
3.1	Catcode checks for loading	15
3.2	Test data	17
3.3	Small test for iniTeX	18
<b>4</b>	<b>Installation</b>	<b>18</b>
4.1	Download	18
4.2	Bundle installation	18
4.3	Package installation	18
4.4	Refresh file name databases	19
4.5	Some details for the interested	19

<b>5 History</b>	<b>19</b>
[2007/12/12 v1.0]	19
[2009/04/10 v1.1]	20
[2010/03/09 v1.2]	20
[2011/03/24 v1.3]	20
[2011/04/10 v1.4]	20
<b>6 Index</b>	<b>20</b>

# 1 Documentation

## 1.1 Introduction

Especially since  $\varepsilon$ -TeX there are many integer values with special meanings, such as catcodes, group types, ... Package `etex`, enabled by options, defines macros in the user namespace for these values.

This package goes another approach for storing the names and values.

- If LuaTeX is available, they are stored in Lua tables.
- Without LuaTeX they are remembered using internal macros.

## 1.2 User interface

The integer values and names are organized in a hierarchical scheme of categories with the property names as leaves. Example:  $\varepsilon$ -TeX's `\currentgrouplevel` reports 2 for a group caused by `\hbox`. This package has chosen to organize the group types in a main category `etex` and its subcategory `grouptype`:

```
etex.grouptype.hbox = 2
```

The property name `hbox` in category `etex.grouptype` has value 2. Dots are used to separate components.

If you want to have the value, the access key is constructed by the category with all its components and the property name. For the opposite the value is used instead of the property name.

Values are always integers (including negative numbers).

### 1.2.1 \magicnum

`\magicnum {⟨access key⟩}`

Macro `\magicnum` expects an access key as argument and expands to the requested data. The macro is always expandable. In case of errors the expansion result is empty.

The same macro is also used for getting a property name. In this case the property name part in the access key is replaced by the value.

The catcodes of the resulting numbers and strings follow TeX's tradition of `\string`, `\meaning`, ...: The space has catcode 10 (`tex.catcode.space`) and the other characters have catcode 12 (`tex.catcode.other`).

Examples:

```
\magicnum{etex.grouptype.hbox} ⇒ 2
\magicnum{tex.catcode.14} ⇒ comment
\magicnum{tex.catcode.undefined} ⇒ ∅
```

### 1.2.2 Properties

- The components of a category are either subcategories or key value pairs, but not both.
- The full specified property names are unique and thus has one integer value exactly.
- Also the values inside a category are unique. This condition is a prerequisite for the reverse mapping of `\magicnum`.
- All names start with a letter. Only letters or digits may follow.

## 1.3 Data

### 1.3.1 Category `tex.catcode`

<code>tex.catcode.escape</code>	0
<code>tex.catcode.begingroup</code>	1
<code>tex.catcode.endgroup</code>	2
<code>tex.catcode.math</code>	3
<code>tex.catcode.align</code>	4
<code>tex.catcode.eol</code>	5
<code>tex.catcode.parameter</code>	6
<code>tex.catcode.superscript</code>	7
<code>tex.catcode.subscript</code>	8
<code>tex.catcode.ignore</code>	9
<code>tex.catcode.space</code>	10
<code>tex.catcode.letter</code>	11
<code>tex.catcode.other</code>	12
<code>tex.catcode.active</code>	13
<code>tex.catcode.comment</code>	14
<code>tex.catcode.invalid</code>	15

### 1.3.2 Category `etex.grouptype`

<code>etex.grouptype.bottomlevel</code>	0
<code>etex.grouptype.simple</code>	1
<code>etex.grouptype.hbox</code>	2
<code>etex.grouptype.adjustedhbox</code>	3
<code>etex.grouptype.vbox</code>	4
<code>etex.grouptype.align</code>	5
<code>etex.grouptype.noalign</code>	6
<code>etex.grouptype.output</code>	8
<code>etex.grouptype.math</code>	9
<code>etex.grouptype.disc</code>	10
<code>etex.grouptype.insert</code>	11
<code>etex.grouptype.vcenter</code>	12
<code>etex.grouptype.mathchoice</code>	13
<code>etex.grouptype.semisimple</code>	14
<code>etex.grouptype.mathshift</code>	15
<code>etex.grouptype.mathleft</code>	16

### 1.3.3 Category etex.ifttype

etex.ifttype.none	0
etex.ifttype.char	1
etex.ifttype.cat	2
etex.ifttype.num	3
etex.ifttype.dim	4
etex.ifttype.odd	5
etex.ifttype.vmode	6
etex.ifttype.hmode	7
etex.ifttype.mmode	8
etex.ifttype.inner	9
etex.ifttype.void	10
etex.ifttype.hbox	11
etex.ifttype.vbox	12
etex.ifttype.x	13
etex.ifttype.eof	14
etex.ifttype.true	15
etex.ifttype.false	16
etex.ifttype.case	17
etex.ifttype.defined	18
etex.ifttype.csname	19
etex.ifttype.fontchar	20

### 1.3.4 Category etex.nodetype

etex.nodetype.none	-1
etex.nodetype.char	0
etex.nodetype.hlist	1
etex.nodetype.vlist	2
etex.nodetype.rule	3
etex.nodetype.ins	4
etex.nodetype.mark	5
etex.nodetype.adjust	6
etex.nodetype.ligature	7
etex.nodetype.disc	8
etex.nodetype.whatsit	9
etex.nodetype.math	10
etex.nodetype.glue	11
etex.nodetype.kern	12
etex.nodetype.penalty	13
etex.nodetype.unset	14
etex.nodetype.maths	15

### 1.3.5 Category etex.interactionmode

etex.interactionmode.batch	0
etex.interactionmode.nonstop	1
etex.interactionmode.scroll	2
etex.interactionmode.errorstop	3

### 1.3.6 Category luatex.pdfliteral.mode

luatex.pdfliteral.mode.setorigin	0
luatex.pdfliteral.mode.page	1
luatex.pdfliteral.mode.direct	2

## 2 Implementation

```
1 (*package\
```

## 2.1 Reload check and package identification

Reload check, especially if the package is not used with L<sup>A</sup>T<sub>E</sub>X.

```
2 \begingroup\catcode61\catcode48\catcode32=10\relax%
3 \catcode13=5 % ^^M
4 \endlinechar=13 %
5 \catcode35=6 % #
6 \catcode39=12 % '
7 \catcode44=12 % ,
8 \catcode45=12 % -
9 \catcode46=12 % .
10 \catcode58=12 % :
11 \catcode64=11 % @
12 \catcode123=1 % {
13 \catcode125=2 % }
14 \expandafter\let\expandafter\x\csname ver@magicnum.sty\endcsname
15 \ifx\x\relax % plain-TeX, first loading
16 \else
17 \def\empty{}%
18 \ifx\x\empty % LaTeX, first loading,
19 % variable is initialized, but \ProvidesPackage not yet seen
20 \else
21 \expandafter\ifx\csname PackageInfo\endcsname\relax
22 \def\x#1#2{%
23 \immediate\write-1{Package #1 Info: #2.}%
24 }%
25 \else
26 \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
27 \fi
28 \x{magicnum}{The package is already loaded}%
29 \aftergroup\endinput
30 \fi
31 \fi
32 \endgroup%
```

Package identification:

```
33 \begingroup\catcode61\catcode48\catcode32=10\relax%
34 \catcode13=5 % ^^M
35 \endlinechar=13 %
36 \catcode35=6 % #
37 \catcode39=12 % '
38 \catcode40=12 % (
39 \catcode41=12 % )
40 \catcode44=12 % ,
41 \catcode45=12 % -
42 \catcode46=12 % .
43 \catcode47=12 % /
44 \catcode58=12 % :
45 \catcode64=11 % @
46 \catcode91=12 % [
47 \catcode93=12 % ]
48 \catcode123=1 % {
49 \catcode125=2 % }
50 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
51 \def\x#1#2#3[#4]{\endgroup
52 \immediate\write-1{Package: #3 #4}%
53 \xdef#1{#4}%
54 }%
55 \else
56 \def\x#1#2[#3]{\endgroup
57 #2[#3]}%
```

```

58     \ifx#1\@undefined
59     \xdef#1{#3}%
60     \fi
61     \ifx#1\relax
62     \xdef#1{#3}%
63     \fi
64 }%
65 \fi
66 \expandafter\x\csname ver@magicnum.sty\endcsname
67 \ProvidesPackage{magicnum}%
68 [2011/04/10 v1.4 Magic numbers (H0)]%

```

## 2.2 Catcodes

```

69 \begingroup\catcode61\catcode48\catcode32=10\relax%
70 \catcode13=5 % ^^M
71 \endlinechar=13 %
72 \catcode123=1 % {
73 \catcode125=2 % }
74 \catcode64=11 % @
75 \def\x{\endgroup
76   \expandafter\edef\csname magicnum@AtEnd\endcsname{%
77     \endlinechar=\the\endlinechar\relax
78     \catcode13=\the\catcode13\relax
79     \catcode32=\the\catcode32\relax
80     \catcode35=\the\catcode35\relax
81     \catcode61=\the\catcode61\relax
82     \catcode64=\the\catcode64\relax
83     \catcode123=\the\catcode123\relax
84     \catcode125=\the\catcode125\relax
85   }%
86 }%
87 \x\catcode61\catcode48\catcode32=10\relax%
88 \catcode13=5 % ^^M
89 \endlinechar=13 %
90 \catcode35=6 % #
91 \catcode64=11 % @
92 \catcode123=1 % {
93 \catcode125=2 % }
94 \def\TMP@EnsureCode#1#2{%
95   \edef\magicnum@AtEnd{%
96     \magicnum@AtEnd
97     \catcode#1=\the\catcode#1\relax
98   }%
99   \catcode#1=#2\relax
100 }
101 \TMP@EnsureCode{34}{12}% "
102 \TMP@EnsureCode{39}{12}% '
103 \TMP@EnsureCode{40}{12}% (
104 \TMP@EnsureCode{41}{12}% )
105 \TMP@EnsureCode{42}{12}% *
106 \TMP@EnsureCode{44}{12}% ,
107 \TMP@EnsureCode{45}{12}% -
108 \TMP@EnsureCode{46}{12}% .
109 \TMP@EnsureCode{47}{12}% /
110 \TMP@EnsureCode{58}{12}% :
111 \TMP@EnsureCode{60}{12}% <
112 \TMP@EnsureCode{62}{12}% >
113 \TMP@EnsureCode{91}{12}% [
114 \TMP@EnsureCode{93}{12}% ]
115 \edef\magicnum@AtEnd{\magicnum@AtEnd\noexpand\endinput}

```

## 2.3 Check for previous definition

```

116 \begingroup\expandafter\expandafter\expandafter\endgroup
117 \expandafter\ifx\csname newcommand\endcsname\relax
118 \expandafter\ifx\csname magicnum\endcsname\relax
119 \else
120 \input infwarerr.sty\relax
121 \@PackageError{magicnum}{%
122 \string\magicnum\space is already defined%
123 }\@ehc
124 \fi
125 \else
126 \newcommand*{\magicnum}{}%
127 \fi

```

## 2.4 Without LuaTeX

```

128 \begingroup\expandafter\expandafter\expandafter\endgroup
129 \expandafter\ifx\csname directlua\endcsname\relax

\magicnum

130 \begingroup\expandafter\expandafter\expandafter\endgroup
131 \expandafter\ifx\csname ifcsname\endcsname\relax
132 \def\magicnum#1{%
133 \expandafter\ifx\csname MG@#1\endcsname\relax
134 \else
135 \csname MG@#1\endcsname
136 \fi
137 }%
138 \else
139 \begingroup
140 \edef\x{\endgroup
141 \def\noexpand\magicnum##1{%
142 \expandafter\noexpand\csname
143 ifcsname\endcsname MG@##1\noexpand\endcsname
144 \noexpand\csname MG@##1%
145 \noexpand\expandafter\noexpand\endcsname
146 \expandafter\noexpand\csname fi\endcsname
147 }%
148 }%
149 \x
150 \fi

151 \else

```

## 2.5 With LuaTeX

```

152 \begingroup\expandafter\expandafter\expandafter\endgroup
153 \expandafter\ifx\csname RequirePackage\endcsname\relax
154 \input ifluatex.sty\relax
155 \input infwarerr.sty\relax
156 \else
157 \RequirePackage{ifluatex}[2010/03/01]%
158 \RequirePackage{infwarerr}[2010/04/08]%
159 \fi

\magicnum@directlua

160 \ifnum\luatexversion<36 %
161 \def\magicnum@directlua{\directlua0 }%
162 \else
163 \let\magicnum@directlua\directlua
164 \fi

165 \magicnum@directlua{%

```

```

166     require("oberdiek.magicnum")%
167 }%
168 \begingroup
169   \def\x{2011/04/10 v1.4}%
170   \def\StripPrefix#1>{}%
171   \edef\x{\expandafter\StripPrefix\meaning\x}%
172   \edef\y{%
173     \magicnum@directlua{%
174       if oberdiek.magicnum.getversion then %
175         oberdiek.magicnum.getversion()%
176       end%
177     }%
178   }%
179   \ifx\x\y
180   \else
181     \@PackageError{magicnum}{%
182       Wrong version of lua module.\MessageBreak
183       Package version: \x\MessageBreak
184       Lua module: \y
185     }\@ehc
186   \fi
187 \endgroup

\luaescapestring
188 \begingroup
189   \expandafter\ifx\csname luaescapestring\endcsname\relax
190     \directlua{%
191       if tex.enableprimitives then %
192         tex.enableprimitives('magicnum@', {'luaescapestring'})%
193       end%
194     }%
195     \global\let\luaescapestring\magicnum@luaescapestring
196   \fi
197   \expandafter\ifx\csname luaescapestring\endcsname\relax
198     \escapechar=92 %
199     \@PackageError{magicnum}{%
200       Missing \string\luaescapestring
201     }\@ehc
202   \fi
203 \endgroup

\magicnum
204 \def\magicnum#1{%
205   \magicnum@directlua{%
206     oberdiek.magicnum.get("\luaescapestring{#1}")%
207   }%
208 }%

209 \expandafter\magicnum@AtEnd
210 \fi%
211 </package>

```

## 2.6 Data

### 2.6.1 Plain data

```

212 <*/data>
213 tex.catcode
214   escape = 0
215   begingroup = 1
216   endgroup = 2
217   math = 3
218   align = 4

```



```

219 eol = 5
220 parameter = 6
221 superscript = 7
222 subscript = 8
223 ignore = 9
224 space = 10
225 letter = 11
226 other = 12
227 active = 13
228 comment = 14
229 invalid = 15
230 etex.grouptype
231 bottomlevel = 0
232 simple = 1
233 hbox = 2
234 adjustedhbox = 3
235 vbox = 4
236 align = 5
237 noalign = 6
238 output = 8
239 math = 9
240 disc = 10
241 insert = 11
242 vcenter = 12
243 mathchoice = 13
244 semisimple = 14
245 mathshift = 15
246 mathleft = 16
247 etex.iftype
248 none = 0
249 char = 1
250 cat = 2
251 num = 3
252 dim = 4
253 odd = 5
254 vmode = 6
255 hmode = 7
256 mmode = 8
257 inner = 9
258 void = 10
259 hbox = 11
260 vbox = 12
261 x = 13
262 eof = 14
263 true = 15
264 false = 16
265 case = 17
266 defined = 18
267 csname = 19
268 fontchar = 20
269 etex.nodetype
270 none = -1
271 char = 0
272 hlist = 1
273 vlist = 2
274 rule = 3
275 ins = 4
276 mark = 5
277 adjust = 6
278 ligature = 7
279 disc = 8
280 whatsit = 9

```

```

281 math = 10
282 glue = 11
283 kern = 12
284 penalty = 13
285 unset = 14
286 maths = 15
287 etex.interactionmode
288 batch = 0
289 nonstop = 1
290 scroll = 2
291 errorstop = 3
292 luatex.pdfliteral.mode
293 setorigin = 0
294 page = 1
295 direct = 2
296 </data>

```

## 2.6.2 Data for T<sub>E</sub>X

```

297 (*package)

\magicnum@add

298 \begingroup\expandafter\expandafter\expandafter\endgroup
299 \expandafter\ifx\csname detokenize\endcsname\relax
300   \def\magicnum@add#1#2#3{%
301     \expandafter\magicnum@@add
302       \csname MG@#1.#2\expandafter\endcsname
303       \csname MG@#1.#3\endcsname
304       {#3}{#2}%
305   }%
306   \def\magicnum@@add#1#2#3#4{%
307     \def#1{#3}%
308     \def#2{#4}%
309     \edef#1{%
310       \expandafter\strip@prefix\meaning#1%
311     }%
312     \edef#2{%
313       \expandafter\strip@prefix\meaning#2%
314     }%
315   }%
316   \expandafter\ifx\csname strip@prefix\endcsname\relax
317     \def\strip@prefix#1->{%}%
318   \fi
319 \else
320   \def\magicnum@add#1#2#3{%
321     \expandafter\edef\csname MG@#1.#2\endcsname{%
322       \detokenize{#3}%
323     }%
324     \expandafter\edef\csname MG@#1.#3\endcsname{%
325       \detokenize{#2}%
326     }%
327   }%
328 \fi

329 \magicnum@add{tex.catcode}{escape}{0}
330 \magicnum@add{tex.catcode}{begingroup}{1}
331 \magicnum@add{tex.catcode}{endgroup}{2}
332 \magicnum@add{tex.catcode}{math}{3}
333 \magicnum@add{tex.catcode}{align}{4}
334 \magicnum@add{tex.catcode}{eol}{5}
335 \magicnum@add{tex.catcode}{parameter}{6}
336 \magicnum@add{tex.catcode}{superscript}{7}
337 \magicnum@add{tex.catcode}{subscript}{8}
338 \magicnum@add{tex.catcode}{ignore}{9}

```

```

339 \magicnum@add{tex.catcode}{space}{10}
340 \magicnum@add{tex.catcode}{letter}{11}
341 \magicnum@add{tex.catcode}{other}{12}
342 \magicnum@add{tex.catcode}{active}{13}
343 \magicnum@add{tex.catcode}{comment}{14}
344 \magicnum@add{tex.catcode}{invalid}{15}
345 \magicnum@add{etex.grouptype}{bottomlevel}{0}
346 \magicnum@add{etex.grouptype}{simple}{1}
347 \magicnum@add{etex.grouptype}{hbox}{2}
348 \magicnum@add{etex.grouptype}{adjustedhbox}{3}
349 \magicnum@add{etex.grouptype}{vbox}{4}
350 \magicnum@add{etex.grouptype}{align}{5}
351 \magicnum@add{etex.grouptype}{noalign}{6}
352 \magicnum@add{etex.grouptype}{output}{8}
353 \magicnum@add{etex.grouptype}{math}{9}
354 \magicnum@add{etex.grouptype}{disc}{10}
355 \magicnum@add{etex.grouptype}{insert}{11}
356 \magicnum@add{etex.grouptype}{vcenter}{12}
357 \magicnum@add{etex.grouptype}{mathchoice}{13}
358 \magicnum@add{etex.grouptype}{semisimple}{14}
359 \magicnum@add{etex.grouptype}{mathshift}{15}
360 \magicnum@add{etex.grouptype}{mathleft}{16}
361 \magicnum@add{etex.ifttype}{none}{0}
362 \magicnum@add{etex.ifttype}{char}{1}
363 \magicnum@add{etex.ifttype}{cat}{2}
364 \magicnum@add{etex.ifttype}{num}{3}
365 \magicnum@add{etex.ifttype}{dim}{4}
366 \magicnum@add{etex.ifttype}{odd}{5}
367 \magicnum@add{etex.ifttype}{vmode}{6}
368 \magicnum@add{etex.ifttype}{hmode}{7}
369 \magicnum@add{etex.ifttype}{mmode}{8}
370 \magicnum@add{etex.ifttype}{inner}{9}
371 \magicnum@add{etex.ifttype}{void}{10}
372 \magicnum@add{etex.ifttype}{hbox}{11}
373 \magicnum@add{etex.ifttype}{vbox}{12}
374 \magicnum@add{etex.ifttype}{x}{13}
375 \magicnum@add{etex.ifttype}{eof}{14}
376 \magicnum@add{etex.ifttype}{true}{15}
377 \magicnum@add{etex.ifttype}{false}{16}
378 \magicnum@add{etex.ifttype}{case}{17}
379 \magicnum@add{etex.ifttype}{defined}{18}
380 \magicnum@add{etex.ifttype}{csname}{19}
381 \magicnum@add{etex.ifttype}{fontchar}{20}
382 \magicnum@add{etex.nodetype}{none}{-1}
383 \magicnum@add{etex.nodetype}{char}{0}
384 \magicnum@add{etex.nodetype}{hlist}{1}
385 \magicnum@add{etex.nodetype}{vlist}{2}
386 \magicnum@add{etex.nodetype}{rule}{3}
387 \magicnum@add{etex.nodetype}{ins}{4}
388 \magicnum@add{etex.nodetype}{mark}{5}
389 \magicnum@add{etex.nodetype}{adjust}{6}
390 \magicnum@add{etex.nodetype}{ligature}{7}
391 \magicnum@add{etex.nodetype}{disc}{8}
392 \magicnum@add{etex.nodetype}{whatsit}{9}
393 \magicnum@add{etex.nodetype}{math}{10}
394 \magicnum@add{etex.nodetype}{glue}{11}
395 \magicnum@add{etex.nodetype}{kern}{12}
396 \magicnum@add{etex.nodetype}{penalty}{13}
397 \magicnum@add{etex.nodetype}{unset}{14}
398 \magicnum@add{etex.nodetype}{maths}{15}
399 \magicnum@add{etex.interactionmode}{batch}{0}
400 \magicnum@add{etex.interactionmode}{nonstop}{1}

```

```

401 \magicnum@add{etex.interactionmode}{scroll}{2}
402 \magicnum@add{etex.interactionmode}{errorstop}{3}
403 \magicnum@add{luatex.pdfliteral.mode}{setorigin}{0}
404 \magicnum@add{luatex.pdfliteral.mode}{page}{1}
405 \magicnum@add{luatex.pdfliteral.mode}{direct}{2}

406 \magicnum@AtEnd%
407 </package>

```

### 2.6.3 Lua module

```

408 <*lua>

409 module("oberdiek.magicnum", package.seeall)

410 function getversion()
411   tex.write("2011/04/10 v1.4")
412 end

413 local data = {
414   ["tex.catcode"] = {
415     [0] = "escape",
416     [1] = "begingroup",
417     [2] = "endgroup",
418     [3] = "math",
419     [4] = "align",
420     [5] = "eol",
421     [6] = "parameter",
422     [7] = "superscript",
423     [8] = "subscript",
424     [9] = "ignore",
425     [10] = "space",
426     [11] = "letter",
427     [12] = "other",
428     [13] = "active",
429     [14] = "comment",
430     [15] = "invalid",
431     ["active"] = 13,
432     ["align"] = 4,
433     ["begingroup"] = 1,
434     ["comment"] = 14,
435     ["endgroup"] = 2,
436     ["eol"] = 5,
437     ["escape"] = 0,
438     ["ignore"] = 9,
439     ["invalid"] = 15,
440     ["letter"] = 11,
441     ["math"] = 3,
442     ["other"] = 12,
443     ["parameter"] = 6,
444     ["space"] = 10,
445     ["subscript"] = 8,
446     ["superscript"] = 7
447   },
448   ["etex.grouptype"] = {
449     [0] = "bottomlevel",
450     [1] = "simple",
451     [2] = "hbox",
452     [3] = "adjustedhbox",
453     [4] = "vbox",
454     [5] = "align",
455     [6] = "noalign",
456     [8] = "output",
457     [9] = "math",
458     [10] = "disc",

```

```

459     [11] = "insert",
460     [12] = "vcenter",
461     [13] = "mathchoice",
462     [14] = "semisimple",
463     [15] = "mathshift",
464     [16] = "mathleft",
465     ["adjustedhbox"] = 3,
466     ["align"] = 5,
467     ["bottomlevel"] = 0,
468     ["disc"] = 10,
469     ["hbox"] = 2,
470     ["insert"] = 11,
471     ["math"] = 9,
472     ["mathchoice"] = 13,
473     ["mathleft"] = 16,
474     ["mathshift"] = 15,
475     ["noalign"] = 6,
476     ["output"] = 8,
477     ["semisimple"] = 14,
478     ["simple"] = 1,
479     ["vbox"] = 4,
480     ["vcenter"] = 12
481 },
482 ["etex.ifttype"] = {
483     [0] = "none",
484     [1] = "char",
485     [2] = "cat",
486     [3] = "num",
487     [4] = "dim",
488     [5] = "odd",
489     [6] = "vmode",
490     [7] = "hmode",
491     [8] = "mmode",
492     [9] = "inner",
493     [10] = "void",
494     [11] = "hbox",
495     [12] = "vbox",
496     [13] = "x",
497     [14] = "eof",
498     [15] = "true",
499     [16] = "false",
500     [17] = "case",
501     [18] = "defined",
502     [19] = "csname",
503     [20] = "fontchar",
504     ["case"] = 17,
505     ["cat"] = 2,
506     ["char"] = 1,
507     ["csname"] = 19,
508     ["defined"] = 18,
509     ["dim"] = 4,
510     ["eof"] = 14,
511     ["false"] = 16,
512     ["fontchar"] = 20,
513     ["hbox"] = 11,
514     ["hmode"] = 7,
515     ["inner"] = 9,
516     ["mmode"] = 8,
517     ["none"] = 0,
518     ["num"] = 3,
519     ["odd"] = 5,
520     ["true"] = 15,

```

```

521     ["vbox"] = 12,
522     ["vmode"] = 6,
523     ["void"] = 10,
524     ["x"] = 13
525 },
526 ["etex.nodetype"] = {
527     [-1] = "none",
528     [0] = "char",
529     [1] = "hlist",
530     [2] = "vlist",
531     [3] = "rule",
532     [4] = "ins",
533     [5] = "mark",
534     [6] = "adjust",
535     [7] = "ligature",
536     [8] = "disc",
537     [9] = "whatsit",
538     [10] = "math",
539     [11] = "glue",
540     [12] = "kern",
541     [13] = "penalty",
542     [14] = "unset",
543     [15] = "maths",
544     ["adjust"] = 6,
545     ["char"] = 0,
546     ["disc"] = 8,
547     ["glue"] = 11,
548     ["hlist"] = 1,
549     ["ins"] = 4,
550     ["kern"] = 12,
551     ["ligature"] = 7,
552     ["mark"] = 5,
553     ["math"] = 10,
554     ["maths"] = 15,
555     ["none"] = -1,
556     ["penalty"] = 13,
557     ["rule"] = 3,
558     ["unset"] = 14,
559     ["vlist"] = 2,
560     ["whatsit"] = 9
561 },
562 ["etex.interactionmode"] = {
563     [0] = "batch",
564     [1] = "nonstop",
565     [2] = "scroll",
566     [3] = "errorstop",
567     ["batch"] = 0,
568     ["errorstop"] = 3,
569     ["nonstop"] = 1,
570     ["scroll"] = 2
571 },
572 ["luatex.pdfliteral.mode"] = {
573     [0] = "setorigin",
574     [1] = "page",
575     [2] = "direct",
576     ["direct"] = 2,
577     ["page"] = 1,
578     ["setorigin"] = 0
579 }
580 }

581 function get(name)
582     local startpos, endpos, category, entry =

```

```

583     string.find(name, "^(%a[%a%d%.]*)%.(-?[%a%d]+)$")
584 if not entry then
585     return
586 end
587 local node = data[category]
588 if not node then
589     return
590 end
591 local num = tonumber(entry)
592 local value
593 if num then
594     value = node[num]
595     if not value then
596         return
597     end
598 else
599     value = node[entry]
600     if not value then
601         return
602     end
603     value = "" .. value
604 end
605 tex.write(value)
606 end
607 </lua>

```

## 3 Test

### 3.1 Catcode checks for loading

```

608 <*test1>
609 \catcode'\{=1 %
610 \catcode'\}=2 %
611 \catcode'\#=6 %
612 \catcode'\@=11 %
613 \expandafter\ifx\csname count@\endcsname\relax
614   \countdef\count@=255 %
615 \fi
616 \expandafter\ifx\csname @gobble\endcsname\relax
617   \long\def\@gobble#1{}%
618 \fi
619 \expandafter\ifx\csname @firstofone\endcsname\relax
620   \long\def\@firstofone#1{#1}%
621 \fi
622 \expandafter\ifx\csname loop\endcsname\relax
623   \expandafter\@firstofone
624 \else
625   \expandafter\@gobble
626 \fi
627 {%
628   \def\loop#1\repeat{%
629     \def\body{#1}%
630     \iterate
631   }%
632   \def\iterate{%
633     \body
634     \let\next\iterate
635   \else
636     \let\next\relax
637   \fi
638   \next
639 }%

```

```

640 \let\repeat=\fi
641 }%
642 \def\RestoreCatcodes{}
643 \count@=0 %
644 \loop
645   \edef\RestoreCatcodes{%
646     \RestoreCatcodes
647     \catcode\the\count@=\the\catcode\count@\relax
648   }%
649 \ifnum\count@<255 %
650   \advance\count@ 1 %
651 \repeat
652
653 \def\RangeCatcodeInvalid#1#2{%
654   \count@=#1\relax
655   \loop
656     \catcode\count@=15 %
657   \ifnum\count@<#2\relax
658     \advance\count@ 1 %
659   \repeat
660 }
661 \def\RangeCatcodeCheck#1#2#3{%
662   \count@=#1\relax
663   \loop
664     \ifnum#3=\catcode\count@
665     \else
666       \errmessage{%
667         Character \the\count@\space
668         with wrong catcode \the\catcode\count@\space
669         instead of \number#3%
670       }%
671     \fi
672   \ifnum\count@<#2\relax
673     \advance\count@ 1 %
674   \repeat
675 }
676 \def\space{ }
677 \expandafter\ifx\csname LoadCommand\endcsname\relax
678   \def\LoadCommand{\input magicnum.sty\relax}%
679 \fi
680 \def\Test{%
681   \RangeCatcodeInvalid{0}{47}%
682   \RangeCatcodeInvalid{58}{64}%
683   \RangeCatcodeInvalid{91}{96}%
684   \RangeCatcodeInvalid{123}{255}%
685   \catcode'\@=12 %
686   \catcode'\=0 %
687   \catcode'\%=14 %
688   \LoadCommand
689   \RangeCatcodeCheck{0}{36}{15}%
690   \RangeCatcodeCheck{37}{37}{14}%
691   \RangeCatcodeCheck{38}{47}{15}%
692   \RangeCatcodeCheck{48}{57}{12}%
693   \RangeCatcodeCheck{58}{63}{15}%
694   \RangeCatcodeCheck{64}{64}{12}%
695   \RangeCatcodeCheck{65}{90}{11}%
696   \RangeCatcodeCheck{91}{91}{15}%
697   \RangeCatcodeCheck{92}{92}{0}%
698   \RangeCatcodeCheck{93}{96}{15}%
699   \RangeCatcodeCheck{97}{122}{11}%
700   \RangeCatcodeCheck{123}{255}{15}%
701   \RestoreCatcodes

```



```

702 }
703 \Test
704 \csname @@end\endcsname
705 \end
706 </test1>

3.2 Test data

707 (*testplain)
708 \input magicnum.sty\relax
709 \def\Test#1#2{%
710   \edef\result{\magicnum{#1}}%
711   \edef\expect{#2}%
712   \edef\expect{\expandafter\stripprefix\meaning\expect}%
713   \ifx\result\expect
714   \else
715     \errmessage{%
716       Failed: [#1] % hash-ok
717       returns [\result] instead of [\expect]%
718     }%
719   \fi
720 }
721 \def\stripprefix#1->{}
722 </testplain>

723 (*testlatex)
724 \NeedsTeXFormat{LaTeX2e}
725 \documentclass{minimal}
726 \usepackage{magicnum}[2011/04/10]
727 \usepackage{qstest}
728 \IncludeTests{*}
729 \LogTests{log}{*}{*}
730 \newcommand*{\Test}[2]{%
731   \Expect*{\magicnum{#1}}{#2}%
732 }
733 \begin{qstest}{magicnum}{magicnum}
734 </testlatex>

735 (*testdata)
736 \Test{tex.catcode.escape}{0}
737 \Test{tex.catcode.invalid}{15}
738 \Test{tex.catcode.unknown}{}
739 \Test{tex.catcode.0}{escape}
740 \Test{tex.catcode.15}{invalid}
741 \Test{etex.iftypes.true}{15}
742 \Test{etex.iftypes.false}{16}
743 \Test{etex.iftypes.15}{true}
744 \Test{etex.iftypes.16}{false}
745 \Test{etex.nodetype.none}{-1}
746 \Test{etex.nodetype.-1}{none}
747 \Test{luatex.pdfliteral.mode.direct}{2}
748 \Test{luatex.pdfliteral.mode.1}{page}
749 \Test{}{}
750 \Test{unknown}{}
751 \Test{unknown.foo.bar}{}
752 \Test{unknown.foo.4}{}
753 </testdata>

754 (*testplain)
755 \csname @@end\endcsname
756 \end
757 </testplain>
758 (*testlatex)
759 \end{qstest}
760 \csname @@end\endcsname

```

761  $\langle$ /testlatex $\rangle$

### 3.3 Small test for $\text{\texttt{iniT\TeX}}$

```
762  $\langle$ *test4 $\rangle$ 
763 \catcode'\{=1
764 \catcode'\}=2
765 \catcode'\#=6
766 \input magicnum.sty\relax
767 \edef\x{\magicnum{tex.catcode.15}}
768 \edef\y{invalid}
769 \def\Strip#1>{}
770 \edef\y{\expandafter\Strip\meaning\y}
771 \ifx\x\y
772   \immediate\write16{0k}%
773 \else
774   \errmessage{\x<>\y}%
775 \fi
776 \csname @@end\endcsname\end
777  $\langle$ /test4 $\rangle$ 
```

## 4 Installation

### 4.1 Download

**Package.** This package is available on CTAN<sup>1</sup>:

[CTAN:macros/latex/contrib/oberdiek/magicnum.dtx](#) The source file.

[CTAN:macros/latex/contrib/oberdiek/magicnum.pdf](#) Documentation.

**Bundle.** All the packages of the bundle ‘oberdiek’ are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

[CTAN:install/macros/latex/contrib/oberdiek.tds.zip](#)

TDS refers to the standard “A Directory Structure for  $\text{\TeX}$  Files” ([CTAN:tds/tds.pdf](#)). Directories with `texmf` in their name are usually organized this way.

### 4.2 Bundle installation

**Unpacking.** Unpack the `oberdiek.tds.zip` in the TDS tree (also known as `texmf` tree) of your choice. Example (linux):

```
unzip oberdiek.tds.zip -d ~/texmf
```

**Script installation.** Check the directory `TDS:scripts/oberdiek/` for scripts that need further installation steps. Package `attachfile2` comes with the Perl script `pdfatfi.pl` that should be installed in such a way that it can be called as `pdfatfi`. Example (linux):

```
chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/
```

### 4.3 Package installation

**Unpacking.** The `.dtx` file is a self-extracting docstrip archive. The files are extracted by running the `.dtx` through plain  $\text{\TeX}$ :

```
tex magicnum.dtx
```

---

<sup>1</sup><http://ftp.ctan.org/tex-archive/>

**TDS.** Now the different files must be moved into the different directories in your installation TDS tree (also known as `texmf` tree):

```

magicnum.sty          → tex/generic/oberdiek/magicnum.sty
magicnum.lua          → scripts/oberdiek/magicnum.lua
oberdiek.magicnum.lua → scripts/oberdiek/oberdiek.magicnum.lua
magicnum.pdf          → doc/latex/oberdiek/magicnum.pdf
magicnum.txt          → doc/latex/oberdiek/magicnum.txt
test/magicnum-test1.tex → doc/latex/oberdiek/test/magicnum-test1.tex
test/magicnum-test2.tex → doc/latex/oberdiek/test/magicnum-test2.tex
test/magicnum-test3.tex → doc/latex/oberdiek/test/magicnum-test3.tex
test/magicnum-test4.tex → doc/latex/oberdiek/test/magicnum-test4.tex
magicnum.dtx          → source/latex/oberdiek/magicnum.dtx

```

If you have a `docstrip.cfg` that configures and enables `docstrip`'s TDS installing feature, then some files can already be in the right place, see the documentation of `docstrip`.

## 4.4 Refresh file name databases

If your  $\text{\TeX}$  distribution (`te $\text{\TeX}$` , `mik $\text{\TeX}$` , ...) relies on file name databases, you must refresh these. For example, `te $\text{\TeX}$`  users run `texhash` or `mktextlsr`.

## 4.5 Some details for the interested

**Attached source.** The PDF documentation on CTAN also includes the `.dtx` source file. It can be extracted by AcrobatReader 6 or higher. Another option is `pdftk`, e.g. unpack the file into the current directory:

```
pdftk magicnum.pdf unpack_files output .
```

**Unpacking with  $\text{\LaTeX}$ .** The `.dtx` chooses its action depending on the format:

**plain  $\text{\TeX}$ :** Run `docstrip` and extract the files.

**$\text{\LaTeX}$ :** Generate the documentation.

If you insist on using  $\text{\LaTeX}$  for `docstrip` (really, `docstrip` does not need  $\text{\LaTeX}$ ), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{magicnum.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

**Generating the documentation.** You can use both the `.dtx` or the `.drv` to generate the documentation. The process can be configured by the configuration file `ltxdoc.cfg`. For instance, put this line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with `pdf $\text{\LaTeX}$` :

```

pdflatex magicnum.dtx
makeindex -s gind.ist magicnum.idx
pdflatex magicnum.dtx
makeindex -s gind.ist magicnum.idx
pdflatex magicnum.dtx

```

# 5 History

[2007/12/12 v1.0]

- First public version.

[2009/04/10 v1.1]

- Adaptation to LuaT<sub>E</sub>X 0.40.

[2010/03/09 v1.2]

- Adaptation to package luatex 0.4.

[2011/03/24 v1.3]

- Catcode fixes.

[2011/04/10 v1.4]

- Compatibility for iniT<sub>E</sub>X.
- Dependency from package luatex removed.
- Version check for lua module.

## 6 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; plain numbers refer to the code lines where the entry is used.

Symbols		146, 153, 189, 197, 299, 302, 303, 316, 321, 324, 613, 616, 619, 622, 677, 704, 755, 760, 776
<code>\#</code>	611, 765	
<code>\%</code>	687	
<code>\@</code>	612, 685	
<code>\@PackageError</code>	121, 181, 199	
<code>\@ehc</code>	123, 185, 201	
<code>\@firstofone</code>	620, 623	
<code>\@gobble</code>	617, 625	
<code>\@undefined</code>	58	
<code>\%</code>	686	
<code>\{</code>	609, 763	
<code>\}</code>	610, 764	
A		
<code>\advance</code>	650, 658, 673	
<code>\aftergroup</code>	29	
B		
<code>\begin</code>	733	
<code>\body</code>	629, 633	
C		
<code>\catcode</code>	2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 33, 34, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 69, 70, 72, 73, 74, 78, 79, 80, 81, 82, 83, 84, 87, 88, 90, 91, 92, 93, 97, 99, 609, 610, 611, 612, 647, 656, 664, 668, 685, 686, 687, 763, 764, 765	
<code>\count@</code>	614, 643, 647, 649, 650, 654, 656, 657, 658, 662, 664, 667, 668, 672, 673	
<code>\countdef</code>	614	
<code>\csname</code>	14, 21, 50, 66, 76, 117, 118, 129, 131, 133, 135, 142, 144,	
D		
<code>\detokenize</code>	322, 325	
<code>\directlua</code>	161, 163, 190	
<code>\documentclass</code>	725	
E		
<code>\empty</code>	17, 18	
<code>\end</code>	705, 756, 759, 776	
<code>\endcsname</code>	14, 21, 50, 66, 76, 117, 118, 129, 131, 133, 135, 143, 145, 146, 153, 189, 197, 299, 302, 303, 316, 321, 324, 613, 616, 619, 622, 677, 704, 755, 760, 776	
<code>\endinput</code>	29, 115	
<code>\endlinechar</code>	4, 35, 71, 77, 89	
<code>\errmessage</code>	666, 715, 774	
<code>\escapechar</code>	198	
<code>\Expect</code>	731	
<code>\expect</code>	711, 712, 713, 717	
I		
<code>\ifnum</code>	160, 649, 657, 664, 672	
<code>\ifx</code>	15, 18, 21, 50, 58, 61, 117, 118, 129, 131, 133, 153, 179, 189, 197, 299, 316, 613, 616, 619, 622, 677, 713, 771	
<code>\immediate</code>	23, 52, 772	
<code>\IncludeTests</code>	728	
<code>\input</code>	120, 154, 155, 678, 708, 766	
<code>\iterate</code>	630, 632, 634	

<b>L</b>		<b>R</b>	
<code>\LoadCommand</code>	678, 688	<code>\RangeCatcodeCheck</code>	661, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700
<code>\LogTests</code>	729	<code>\RangeCatcodeInvalid</code>	653, 681, 682, 683, 684
<code>\loop</code>	628, 644, 655, 663	<code>\repeat</code>	628, 640, 651, 659, 674
<code>\luaescapestring</code>	188, 206	<code>\RequirePackage</code>	157, 158
<code>\luatexversion</code>	160	<code>\RestoreCatcodes</code>	642, 645, 646, 701
<b>M</b>		<code>\result</code>	710, 713, 717
<code>\magicnum</code>	2, 122, 126, 130, 204, 710, 731, 767	<b>S</b>	
<code>\magicnum@@add</code>	301, 306	<code>\space</code>	122, 667, 668, 676
<code>\magicnum@add</code>	298, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405	<code>\Strip</code>	769, 770
<code>\magicnum@AtEnd</code>	95, 96, 115, 209, 406	<code>\strip@prefix</code>	310, 313, 317
<code>\magicnum@directlua</code>	160, 165, 173, 205	<code>\StripPrefix</code>	170, 171
<code>\magicnum@luaescapestring</code>	195	<code>\stripprefix</code>	712, 721
<code>\meaning</code>	171, 310, 313, 712, 770	<b>T</b>	
<code>\MessageBreak</code>	182, 183	<code>\Test</code>	680, 703, 709, 730, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752
<b>N</b>		<code>\the</code>	77, 78, 79, 80, 81, 82, 83, 84, 97, 647, 667, 668
<code>\NeedsTeXFormat</code>	724	<code>\TMP@EnsureCode</code>	94, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114
<code>\newcommand</code>	126, 730	<b>U</b>	
<code>\next</code>	634, 636, 638	<code>\usepackage</code>	726, 727
<code>\number</code>	669	<b>W</b>	
<b>P</b>		<code>\write</code>	23, 52, 772
<code>\PackageInfo</code>	26	<b>X</b>	
<code>\ProvidesPackage</code>	19, 67	<code>\x</code>	14, 15, 18, 22, 26, 28, 51, 56, 66, 75, 87, 140, 149, 169, 171, 179, 183, 767, 771, 774
		<b>Y</b>	
		<code>\y</code>	172, 179, 184, 768, 770, 771, 774