The magicnum package

Heiko Oberdiek

<heiko.oberdiek at googlemail.com>

2009/04/10 v1.1

Abstract

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

Contents

1 Documentation

1.1 Introduction

Especially since ε-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: ε-TeX's `\currentgrouplevel` reports 2 for a group caused by `\hbox`. This package has choosen 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}` $\Rightarrow$ 2
> `\magicnum{tex.catcode.14}` $\Rightarrow$ comment
> `\magicnum{tex.catcode.undefined}` $\Rightarrow$ $\emptyset$

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`

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

1.3.2 Category `etex.grouptype`

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

1.3.3 Category `etex.iftype`

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

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

Package identification:

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

```
58    }%
59  \fi
60 \expandafter\x\csname ver@magicnum.sty\endcsname
61 \ProvidesPackage{magicnum}%
62   [2009/04/10 v1.1 Magic numbers (HO)]
```

2.2   Catcodes

```
63 \begingroup
64   \catcode123 1 % {
65   \catcode125 2 % }
66   \def\x{\endgroup
67     \expandafter\edef\csname magicnum@AtEnd\endcsname{%
68       \catcode35 \the\catcode35\relax
69       \catcode64 \the\catcode64\relax
70       \catcode123 \the\catcode123\relax
71       \catcode125 \the\catcode125\relax
72     }%
73   }%
74 \x
75 \catcode35 6 % #
76 \catcode64 11 % @
77 \catcode123 1 % {
78 \catcode125 2 % }
79 \def\TMP@EnsureCode#1#2{%
80   \edef\magicnum@AtEnd{%
81     \magicnum@AtEnd
82     \catcode#1 \the\catcode#1\relax
83   }%
84   \catcode#1 #2\relax
85 }
86 \TMP@EnsureCode{34}{12}% "
87 \TMP@EnsureCode{40}{12}% (
88 \TMP@EnsureCode{41}{12}% )
89 \TMP@EnsureCode{42}{12}% *
90 \TMP@EnsureCode{44}{12}% ,
91 \TMP@EnsureCode{45}{12}% -
92 \TMP@EnsureCode{46}{12}% .
93 \TMP@EnsureCode{47}{12}% /
94 \TMP@EnsureCode{60}{12}% <
95 \TMP@EnsureCode{62}{12}% >
```

2.3   Check for previous definition

```
96 \begingroup\expandafter\expandafter\expandafter\endgroup
97 \expandafter\ifx\csname newcommand\endcsname\relax
98   \expandafter\ifx\csname magicnum\endcsname\relax
99   \else
100    \input infwarerr.sty\relax
101    \@PackageErrorNoLine{magicnum}{%
102      \string\magicnum\space is already defined%
103    }\@ehc
104  \fi
105 \else
106   \newcommand*{\magicnum}{}%
107 \fi
```

2.4   Without LuaTEX

```
108 \begingroup\expandafter\expandafter\expandafter\endgroup
109 \expandafter\ifx\csname directlua\endcsname\relax
```

\magicnum

```
110   \begingroup\expandafter\expandafter\expandafter\endgroup
```

6

```
111  \expandafter\ifx\csname ifcsname\endcsname\relax
112    \def\magicnum#1{%
113      \expandafter\ifx\csname MG@#1\endcsname\relax
114      \else
115        \csname MG@#1\endcsname
116      \fi
117    }%
118  \else
119    \begingroup
120      \edef\x{\endgroup
121        \def\noexpand\magicnum##1{%
122          \expandafter\noexpand\csname
123          ifcsname\endcsname MG@##1\noexpand\endcsname
124            \noexpand\csname MG@##1%
125                \noexpand\expandafter\noexpand\endcsname
126          \expandafter\noexpand\csname fi\endcsname
127        }%
128      }%
129    \x
130  \fi
131 \else
```

2.5 With LuaTeX

```
132  \begingroup\expandafter\expandafter\expandafter\endgroup
133  \expandafter\ifx\csname RequirePackage\endcsname\relax
134    \input luatex.sty\relax
135  \else
136    \RequirePackage{luatex}[2009/04/10]%
137  \fi
```

\magicnum@directlua

```
138  \ifnum\luatexversion<36 %
139    \def\magicnum@directlua{\directlua0 }%
140  \else
141    \let\magicnum@directlua\directlua
142  \fi
```

\magicnum

```
143  \def\magicnum#1{%
144    \magicnum@directlua{%
145      require("oberdiek.magicnum")%
146      oberdiek.magicnum.get(%
147        "\luaescapestring{#1}",\number\CatcodeTableString
148      )%
149    }%
150  }%
```

```
151  \magicnum@AtEnd
152  \expandafter\endinput
153 \fi
154 ⟨/package⟩
```

2.6 Data

2.6.1 Plain data

```
155 ⟨*data⟩
156 tex.catcode
157   escape = 0
158   begingroup = 1
159   endgroup = 2
```

```
160    math = 3
161    align = 4
162    eol = 5
163    parameter = 6
164    superscript = 7
165    subscript = 8
166    ignore = 9
167    space = 10
168    letter = 11
169    other = 12
170    active = 13
171    comment = 14
172    invalid = 15
173 etex.grouptype
174    bottomlevel = 0
175    simple = 1
176    hbox = 2
177    adjustedhbox = 3
178    vbox = 4
179    align = 5
180    noalign = 6
181    output = 8
182    math = 9
183    disc = 10
184    insert = 11
185    vcenter = 12
186    mathchoice = 13
187    semisimple = 14
188    mathshift = 15
189    mathleft = 16
190 etex.iftype
191    none = 0
192    char = 1
193    cat = 2
194    num = 3
195    dim = 4
196    odd = 5
197    vmode = 6
198    hmode = 7
199    mmode = 8
200    inner = 9
201    void = 10
202    hbox = 11
203    vbox = 12
204    x = 13
205    eof = 14
206    true = 15
207    false = 16
208    case = 17
209    defined = 18
210    csname = 19
211    fontchar = 20
212 etex.nodetype
213    none = -1
214    char = 0
215    hlist = 1
216    vlist = 2
217    rule = 3
218    ins = 4
219    mark = 5
220    adjust = 6
221    ligature = 7
```

```
222    disc = 8
223    whatsit = 9
224    math = 10
225    glue = 11
226    kern = 12
227    penalty = 13
228    unset = 14
229    maths = 15
230 etex.interactionmode
231    batch = 0
232    nonstop = 1
233    scroll = 2
234    errorstop = 3
235 luatex.pdfliteral.mode
236    setorigin = 0
237    page = 1
238    direct = 2
239 ⟨/data⟩
```

2.6.2 Data for TEX

```
240 ⟨*package⟩
```

\magicnum@add

```
241 \begingroup\expandafter\expandafter\expandafter\endgroup
242 \expandafter\ifx\csname detokenize\endcsname\relax
243   \def\magicnum@add#1#2#3{%
244     \expandafter\magicnum@@add
245         \csname MG@#1.#2\expandafter\endcsname
246         \csname MG@#1.#3\endcsname
247       {#3}{#2}%
248   }%
249   \def\magicnum@@add#1#2#3#4{%
250     \def#1{#3}%
251     \def#2{#4}%
252     \edef#1{%
253       \expandafter\strip@prefix\meaning#1%
254     }%
255     \edef#2{%
256       \expandafter\strip@prefix\meaning#2%
257     }%
258   }%
259   \expandafter\ifx\csname strip@prefix\endcsname\relax
260     \def\strip@prefix#1->{}%
261   \fi
262 \else
263   \def\magicnum@add#1#2#3{%
264     \expandafter\edef\csname MG@#1.#2\endcsname{%
265       \detokenize{#3}%
266     }%
267     \expandafter\edef\csname MG@#1.#3\endcsname{%
268       \detokenize{#2}%
269     }%
270   }%
271 \fi

272 \magicnum@add{tex.catcode}{escape}{0}
273 \magicnum@add{tex.catcode}{begingroup}{1}
274 \magicnum@add{tex.catcode}{endgroup}{2}
275 \magicnum@add{tex.catcode}{math}{3}
276 \magicnum@add{tex.catcode}{align}{4}
277 \magicnum@add{tex.catcode}{eol}{5}
278 \magicnum@add{tex.catcode}{parameter}{6}
279 \magicnum@add{tex.catcode}{superscript}{7}
```

9

```
280 \magicnum@add{tex.catcode}{subscript}{8}
281 \magicnum@add{tex.catcode}{ignore}{9}
282 \magicnum@add{tex.catcode}{space}{10}
283 \magicnum@add{tex.catcode}{letter}{11}
284 \magicnum@add{tex.catcode}{other}{12}
285 \magicnum@add{tex.catcode}{active}{13}
286 \magicnum@add{tex.catcode}{comment}{14}
287 \magicnum@add{tex.catcode}{invalid}{15}
288 \magicnum@add{etex.grouptype}{bottomlevel}{0}
289 \magicnum@add{etex.grouptype}{simple}{1}
290 \magicnum@add{etex.grouptype}{hbox}{2}
291 \magicnum@add{etex.grouptype}{adjustedhbox}{3}
292 \magicnum@add{etex.grouptype}{vbox}{4}
293 \magicnum@add{etex.grouptype}{align}{5}
294 \magicnum@add{etex.grouptype}{noalign}{6}
295 \magicnum@add{etex.grouptype}{output}{8}
296 \magicnum@add{etex.grouptype}{math}{9}
297 \magicnum@add{etex.grouptype}{disc}{10}
298 \magicnum@add{etex.grouptype}{insert}{11}
299 \magicnum@add{etex.grouptype}{vcenter}{12}
300 \magicnum@add{etex.grouptype}{mathchoice}{13}
301 \magicnum@add{etex.grouptype}{semisimple}{14}
302 \magicnum@add{etex.grouptype}{mathshift}{15}
303 \magicnum@add{etex.grouptype}{mathleft}{16}
304 \magicnum@add{etex.iftype}{none}{0}
305 \magicnum@add{etex.iftype}{char}{1}
306 \magicnum@add{etex.iftype}{cat}{2}
307 \magicnum@add{etex.iftype}{num}{3}
308 \magicnum@add{etex.iftype}{dim}{4}
309 \magicnum@add{etex.iftype}{odd}{5}
310 \magicnum@add{etex.iftype}{vmode}{6}
311 \magicnum@add{etex.iftype}{hmode}{7}
312 \magicnum@add{etex.iftype}{mmode}{8}
313 \magicnum@add{etex.iftype}{inner}{9}
314 \magicnum@add{etex.iftype}{void}{10}
315 \magicnum@add{etex.iftype}{hbox}{11}
316 \magicnum@add{etex.iftype}{vbox}{12}
317 \magicnum@add{etex.iftype}{x}{13}
318 \magicnum@add{etex.iftype}{eof}{14}
319 \magicnum@add{etex.iftype}{true}{15}
320 \magicnum@add{etex.iftype}{false}{16}
321 \magicnum@add{etex.iftype}{case}{17}
322 \magicnum@add{etex.iftype}{defined}{18}
323 \magicnum@add{etex.iftype}{csname}{19}
324 \magicnum@add{etex.iftype}{fontchar}{20}
325 \magicnum@add{etex.nodetype}{none}{-1}
326 \magicnum@add{etex.nodetype}{char}{0}
327 \magicnum@add{etex.nodetype}{hlist}{1}
328 \magicnum@add{etex.nodetype}{vlist}{2}
329 \magicnum@add{etex.nodetype}{rule}{3}
330 \magicnum@add{etex.nodetype}{ins}{4}
331 \magicnum@add{etex.nodetype}{mark}{5}
332 \magicnum@add{etex.nodetype}{adjust}{6}
333 \magicnum@add{etex.nodetype}{ligature}{7}
334 \magicnum@add{etex.nodetype}{disc}{8}
335 \magicnum@add{etex.nodetype}{whatsit}{9}
336 \magicnum@add{etex.nodetype}{math}{10}
337 \magicnum@add{etex.nodetype}{glue}{11}
338 \magicnum@add{etex.nodetype}{kern}{12}
339 \magicnum@add{etex.nodetype}{penalty}{13}
340 \magicnum@add{etex.nodetype}{unset}{14}
341 \magicnum@add{etex.nodetype}{maths}{15}
```

```
342 \magicnum@add{etex.interactionmode}{batch}{0}
343 \magicnum@add{etex.interactionmode}{nonstop}{1}
344 \magicnum@add{etex.interactionmode}{scroll}{2}
345 \magicnum@add{etex.interactionmode}{errorstop}{3}
346 \magicnum@add{luatex.pdfliteral.mode}{setorigin}{0}
347 \magicnum@add{luatex.pdfliteral.mode}{page}{1}
348 \magicnum@add{luatex.pdfliteral.mode}{direct}{2}
349 \magicnum@AtEnd
350 ⟨/package⟩
```

2.6.3   Lua module

```
351 ⟨*lua⟩
```

```
352 module("oberdiek.magicnum", package.seeall)
```

```
353 local data = {
354   ["tex.catcode"] = {
355     [0] = "escape",
356     [1] = "begingroup",
357     [2] = "endgroup",
358     [3] = "math",
359     [4] = "align",
360     [5] = "eol",
361     [6] = "parameter",
362     [7] = "superscript",
363     [8] = "subscript",
364     [9] = "ignore",
365     [10] = "space",
366     [11] = "letter",
367     [12] = "other",
368     [13] = "active",
369     [14] = "comment",
370     [15] = "invalid",
371     ["active"] = 13,
372     ["align"] = 4,
373     ["begingroup"] = 1,
374     ["comment"] = 14,
375     ["endgroup"] = 2,
376     ["eol"] = 5,
377     ["escape"] = 0,
378     ["ignore"] = 9,
379     ["invalid"] = 15,
380     ["letter"] = 11,
381     ["math"] = 3,
382     ["other"] = 12,
383     ["parameter"] = 6,
384     ["space"] = 10,
385     ["subscript"] = 8,
386     ["superscript"] = 7
387   },
388   ["etex.grouptype"] = {
389     [0] = "bottomlevel",
390     [1] = "simple",
391     [2] = "hbox",
392     [3] = "adjustedhbox",
393     [4] = "vbox",
394     [5] = "align",
395     [6] = "noalign",
396     [8] = "output",
397     [9] = "math",
398     [10] = "disc",
399     [11] = "insert",
400     [12] = "vcenter",
```

```
401    [13] = "mathchoice",
402    [14] = "semisimple",
403    [15] = "mathshift",
404    [16] = "mathleft",
405    ["adjustedhbox"] = 3,
406    ["align"] = 5,
407    ["bottomlevel"] = 0,
408    ["disc"] = 10,
409    ["hbox"] = 2,
410    ["insert"] = 11,
411    ["math"] = 9,
412    ["mathchoice"] = 13,
413    ["mathleft"] = 16,
414    ["mathshift"] = 15,
415    ["noalign"] = 6,
416    ["output"] = 8,
417    ["semisimple"] = 14,
418    ["simple"] = 1,
419    ["vbox"] = 4,
420    ["vcenter"] = 12
421    },
422    ["etex.iftype"] = {
423    [0] = "none",
424    [1] = "char",
425    [2] = "cat",
426    [3] = "num",
427    [4] = "dim",
428    [5] = "odd",
429    [6] = "vmode",
430    [7] = "hmode",
431    [8] = "mmode",
432    [9] = "inner",
433    [10] = "void",
434    [11] = "hbox",
435    [12] = "vbox",
436    [13] = "x",
437    [14] = "eof",
438    [15] = "true",
439    [16] = "false",
440    [17] = "case",
441    [18] = "defined",
442    [19] = "csname",
443    [20] = "fontchar",
444    ["case"] = 17,
445    ["cat"] = 2,
446    ["char"] = 1,
447    ["csname"] = 19,
448    ["defined"] = 18,
449    ["dim"] = 4,
450    ["eof"] = 14,
451    ["false"] = 16,
452    ["fontchar"] = 20,
453    ["hbox"] = 11,
454    ["hmode"] = 7,
455    ["inner"] = 9,
456    ["mmode"] = 8,
457    ["none"] = 0,
458    ["num"] = 3,
459    ["odd"] = 5,
460    ["true"] = 15,
461    ["vbox"] = 12,
462    ["vmode"] = 6,
```

```
463        ["void"] = 10,
464        ["x"] = 13
465      },
466      ["etex.nodetype"] = {
467        [-1] = "none",
468        [0] = "char",
469        [1] = "hlist",
470        [2] = "vlist",
471        [3] = "rule",
472        [4] = "ins",
473        [5] = "mark",
474        [6] = "adjust",
475        [7] = "ligature",
476        [8] = "disc",
477        [9] = "whatsit",
478        [10] = "math",
479        [11] = "glue",
480        [12] = "kern",
481        [13] = "penalty",
482        [14] = "unset",
483        [15] = "maths",
484        ["adjust"] = 6,
485        ["char"] = 0,
486        ["disc"] = 8,
487        ["glue"] = 11,
488        ["hlist"] = 1,
489        ["ins"] = 4,
490        ["kern"] = 12,
491        ["ligature"] = 7,
492        ["mark"] = 5,
493        ["math"] = 10,
494        ["maths"] = 15,
495        ["none"] = -1,
496        ["penalty"] = 13,
497        ["rule"] = 3,
498        ["unset"] = 14,
499        ["vlist"] = 2,
500        ["whatsit"] = 9
501      },
502      ["etex.interactionmode"] = {
503        [0] = "batch",
504        [1] = "nonstop",
505        [2] = "scroll",
506        [3] = "errorstop",
507        ["batch"] = 0,
508        ["errorstop"] = 3,
509        ["nonstop"] = 1,
510        ["scroll"] = 2
511      },
512      ["luatex.pdfliteral.mode"] = {
513        [0] = "setorigin",
514        [1] = "page",
515        [2] = "direct",
516        ["direct"] = 2,
517        ["page"] = 1,
518        ["setorigin"] = 0
519      }
520 }

521 function get(name, catcodetable)
522   local startpos, endpos, category, entry =
523       string.find(name, "^(%a[%a%d%.]*)%.(-?[%a%d]+)$")
524   if not entry then
```

```
525     return
526   end
527   local node = data[category]
528   if not node then
529     return
530   end
531   local num = tonumber(entry)
532   local value
533   if num then
534     value = node[num]
535     if not value then
536       return
537     end
538   else
539     value = node[entry]
540     if not value then
541       return
542     end
543     value = "" .. value
544   end
545   tex.print(catcodetable, value)
546 end
```

547 ⟨/lua⟩

3   Test

3.1   Catcode checks for loading

548 ⟨*test1⟩

```
549 \catcode`\{=1 %
550 \catcode`\}=2 %
551 \catcode`\#=6 %
552 \catcode`\@=11 %
553 \expandafter\ifx\csname count@\endcsname\relax
554   \countdef\count@=255 %
555 \fi
556 \expandafter\ifx\csname @gobble\endcsname\relax
557   \long\def\@gobble#1{}%
558 \fi
559 \expandafter\ifx\csname @firstofone\endcsname\relax
560   \long\def\@firstofone#1{#1}%
561 \fi
562 \expandafter\ifx\csname loop\endcsname\relax
563   \expandafter\@firstofone
564 \else
565   \expandafter\@gobble
566 \fi
567 {%
568   \def\loop#1\repeat{%
569     \def\body{#1}%
570     \iterate
571   }%
572   \def\iterate{%
573     \body
574       \let\next\iterate
575     \else
576       \let\next\relax
577     \fi
578     \next
579   }%
580   \let\repeat=\fi
581 }%
```

14

```
582 \def\RestoreCatcodes{}
583 \count@=0 %
584 \loop
585   \edef\RestoreCatcodes{%
586     \RestoreCatcodes
587     \catcode\the\count@=\the\catcode\count@\relax
588   }%
589 \ifnum\count@<255 %
590   \advance\count@ 1 %
591 \repeat
592
593 \def\RangeCatcodeInvalid#1#2{%
594   \count@=#1\relax
595   \loop
596     \catcode\count@=15 %
597   \ifnum\count@<#2\relax
598     \advance\count@ 1 %
599   \repeat
600 }
601 \expandafter\ifx\csname LoadCommand\endcsname\relax
602   \def\LoadCommand{\input magicnum.sty\relax}%
603 \fi
604 \def\Test{%
605   \RangeCatcodeInvalid{0}{47}%
606   \RangeCatcodeInvalid{58}{64}%
607   \RangeCatcodeInvalid{91}{96}%
608   \RangeCatcodeInvalid{123}{255}%
609   \catcode`\@=12 %
610   \catcode`\\=0 %
611   \catcode`\{=1 %
612   \catcode`\}=2 %
613   \catcode`\#=6 %
614   \catcode`\[=12 %
615   \catcode`\]=12 %
616   \catcode`\%=14 %
617   \catcode`\ =10 %
618   \catcode13=5 %
619   \LoadCommand
620   \RestoreCatcodes
621 }
622 \Test
623 \csname @@end\endcsname
624 \end
625 ⟨/test1⟩
```

3.2   Test data

```
626 ⟨*testplain⟩
627 \input magicnum.sty\relax
628 \def\Test#1#2{%
629   \edef\result{\magicnum{#1}}%
630   \edef\expect{#2}%
631   \edef\expect{\expandafter\stripprefix\meaning\expect}%
632   \ifx\result\expect
633   \else
634     \errmessage{%
635       Failed: [#1] % hash-ok
636       returns [\result] instead of [\expect]%
637     }%
638   \fi
639 }
640 \def\stripprefix#1->{}
641 ⟨/testplain⟩
```

```
642 ⟨*testlatex⟩
643 \NeedsTeXFormat{LaTeX2e}
644 \documentclass{minimal}
645 \usepackage{magicnum}[2009/04/10]
646 \usepackage{qstest}
647 \IncludeTests{*}
648 \LogTests{log}{*}{*}
649 \newcommand*{\Test}[2]{%
650   \Expect*{\magicnum{#1}}{#2}%
651 }
652 \begin{qstest}{magicnum}{magicnum}
653 ⟨/testlatex⟩

654 ⟨*testdata⟩
655 \Test{tex.catcode.escape}{0}
656 \Test{tex.catcode.invalid}{15}
657 \Test{tex.catcode.unknown}{}
658 \Test{tex.catcode.0}{escape}
659 \Test{tex.catcode.15}{invalid}
660 \Test{etex.iftype.true}{15}
661 \Test{etex.iftype.false}{16}
662 \Test{etex.iftype.15}{true}
663 \Test{etex.iftype.16}{false}
664 \Test{etex.nodetype.none}{-1}
665 \Test{etex.nodetype.-1}{none}
666 \Test{luatex.pdfliteral.mode.direct}{2}
667 \Test{luatex.pdfliteral.mode.1}{page}
668 \Test{}{}
669 \Test{unknown}{}
670 \Test{unknown.foo.bar}{}
671 \Test{unknown.foo.4}{}
672 ⟨/testdata⟩

673 ⟨*testplain⟩
674 \csname @@end\endcsname
675 \end
676 ⟨/testplain⟩
677 ⟨*testlatex⟩
678 \end{qstest}
679 \csname @@end\endcsname
680 ⟨/testlatex⟩
```

4 Installation

4.1 Download

Package. This package is available on CTAN[1]:

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 TeX Files" (CTAN:tds/tds.pdf). Directories with texmf in their name are usually organized this way.

---

[1] ftp://ftp.ctan.org/tex-archive/

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-TEX:

```
tex magicnum.dtx
```

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
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 TEX distribution (teTEX, mikTEX, ...) relies on file name databases, you must refresh these. For example, teTEX users run `texhash` or `mktexlsr`.

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 LATEX.   The `.dtx` chooses its action depending on the format:

plain-TEX: Run `docstrip` and extract the files.

LATEX: Generate the documentation.

If you insist on using LATEX for `docstrip` (really, `docstrip` does not need 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 pdfLATEX:

```
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 LUATEX 0.40.

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; numbers in roman refer to the code lines where the entry is used.

Symbols

\# 551, 613
\% 616
\@ 552, 609
\@PackageErrorNoLine 101
\@ehc 103
\@firstofone 560, 563
\@gobble 557, 565
\@undefined 52
\[. . . . 614
\\ 610
\{ 549, 611
\} 550, 612
\] 615

\␣ 617

A

\advance 590, 598
\aftergroup 26

B

\begin 652
\body 569, 573

C

\catcode 3, 4, 5, 6, 7, 8, 9, 17, 31, 32,
33, 34, 35, 36, 37, 38, 39, 40, 41,
42, 43, 64, 65, 68, 69, 70, 71, 75,
76, 77, 78, 82, 84, 549, 550, 551,
552, 587, 596, 609, 610, 611,
612, 613, 614, 615, 616, 617, 618
\CatcodeTableString 147
\count@ 554, 583,
587, 589, 590, 594, 596, 597, 598
\countdef 554
\csname 10, 18,
44, 60, 67, 97, 98, 109, 111, 113,
115, 122, 124, 126, 133, 242,
245, 246, 259, 264, 267, 553,
556, 559, 562, 601, 623, 674, 679

D

\detokenize 265, 268
\directlua 139, 141
\documentclass 644

E

\empty 13, 14
\end 624, 675, 678
\endcsname 10, 18,
44, 60, 67, 97, 98, 109, 111, 113,
115, 123, 125, 126, 133, 242,
245, 246, 259, 264, 267, 553,
556, 559, 562, 601, 623, 674, 679
\endinput 26, 152
\errmessage 634
\Expect 650
\expect 630, 631, 632, 636

19