The **magicum** package

Heiko Oberdiek*

<heiko.oberdiek at googlemail.com>

2016/05/16 v1.5

Abstract

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

Contents

1 Documentation  2
  1.1 Introduction ............................................. 2
  1.2 User interface .......................................... 2
    1.2.1 \magicnum .................................. 2
    1.2.2 Properties .................................. 3
  1.3 Data .................................................. 3
    1.3.1 Category tex.catcode .......................... 3
    1.3.2 Category etex.grouptype ....................... 3
    1.3.3 Category etex.iftype ......................... 4
    1.3.4 Category etex.nodetype ....................... 4
    1.3.5 Category etex.interactionmode ............... 4
    1.3.6 Category luatex.pdfliteral.mode .......... 4

2 Implementation  4
  2.1 Reload check and package identification ............. 5
  2.2 Catcodes ............................................. 6
  2.3 Check for previous definition ....................... 7
  2.4 Without Lua\TeX{} .................................. 7
  2.5 With Lua\TeX{} .................................... 7
  2.6 Data ................................................ 8
    2.6.1 Plain data .................................. 8
    2.6.2 Data for \TeX{} ................................ 10
    2.6.3 Lua module .................................. 12

3 Test  15
  3.1 Catcode checks for loading ........................... 15
  3.2 Test data ........................................... 17
  3.3 Small test for \init\TeX{} ......................... 18

4 Installation  18
  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

5 Catalogue  19

*Please report any issues at https://github.com/ho-tex/oberdiek/issues
1 Documentation

1.1 Introduction

Especially since \texttt{\LaTeX} there are many integer values with special meanings, such as catcodes, group types, ...Package \texttt{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 Lua\TeX{} is available, they are stored in Lua tables.
- Without Lua\TeX{} 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: \texttt{\LaTeX}’s \texttt{\currentgrouplevel} reports 2 for a group caused by \texttt{\hbox}. This package has chosen to organize the group types in a main category \texttt{etex} and its subcategory \texttt{grouptype}:

\begin{verbatim}
etex.grouptype.hbox = 2\end{verbatim}

The property name \texttt{hbox} in category \texttt{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 \texttt{\magicnum}

\begin{verbatim}
\magicnum \{\langle access key\rangle\}\end{verbatim}

Macro \texttt{\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 \texttt{\string}, \texttt{\meaning}, ...: The space has catcode 10 (\texttt{\tex.catcode.space}) and the other characters have catcode 12 (\texttt{\tex.catcode.other}).

Examples:

\begin{verbatim}
\magicnum{etex.grouptype.hbox} ⇒ 2
\magicnum{tex.catcode.14} ⇒ comment
\magicnum{tex.catcode.undefined} ⇒ ∅\end{verbatim}
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 \texttt{\textbackslash magicnum}.
- All names start with a letter. Only letters or digits may follow.

1.3 Data

1.3.1 Category \texttt{tex.catcode}

\begin{verbatim}
  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
\end{verbatim}

1.3.2 Category \texttt{etex.grouptype}

\begin{verbatim}
  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
\end{verbatim}
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
2.1 Reload check and package identification

Reload check, especially if the package is not used with \LaTeX{}.
2.2 Catcodes

\begin{group}
code{61}\catcode{48}\catcode{32}=10\relax\% 
\catcode{13}=5 \^^M 
\endlinechar=13 \% 
\catcode{123}=1 \{ 
\catcode{125}=2 \} 
\def{x}{\endgroup 
\expandafter\edef\csname magicnum@AtEnd\endcsname{\endlinechar=\the\endlinechar\relax 
\catcode{13}=\the\catcode{13}\relax 
\catcode{32}=\the\catcode{32}\relax 
\catcode{35}=\the\catcode{35}\relax 
\catcode{61}=\the\catcode{61}\relax 
\catcode{64}=\the\catcode{64}\relax 
\catcode{123}=\the\catcode{123}\relax 
\catcode{125}=\the\catcode{125}\relax 
} \% 
\x\catcode{61}\catcode{48}\catcode{32}=10\relax\% 
\catcode{13}=5 \^^M 
\endlinechar=13 \% 
\catcode{35}=6 \# 
\catcode{64}=11 \@ 
\catcode{123}=1 \{ 
\catcode{125}=2 \} 
\def\TMP@EnsureCode#1#2{\edef\magicnum@AtEnd{\magicnum@AtEnd
\catcode{#1}=\the\catcode{#1}\relax 
\catcode{#1}=\#2\relax 
\endinput} 
\TMP@EnsureCode{34}{12}\% * 
\TMP@EnsureCode{39}{12}\% ' 
\TMP@EnsureCode{40}{12}\% ( 
\TMP@EnsureCode{41}{12}\% ) 
\TMP@EnsureCode{42}{12}\% * 
\TMP@EnsureCode{44}{12}\% , 
\TMP@EnsureCode{45}{12}\% - 
\TMP@EnsureCode{46}{12}\% . 
\TMP@EnsureCode{47}{12}\% / 
\TMP@EnsureCode{58}{12}\% : 
\TMP@EnsureCode{60}{12}\% < 
\TMP@EnsureCode{62}{12}\% > 
\TMP@EnsureCode{91}{12}\% [ 
\TMP@EnsureCode{93}{12}\% ] 
\edef\magicnum@AtEnd{\magicnum@AtEnd\noexpand\endinput}
2.3 Check for previous definition

\begingroup\expandafter\expandafter\expandafter\endgroup
\ifx\csname newcommand\endcsname\relax
\else
\input infwarerr.sty\relax
\PackageError{magicnum}{string\space magicnum\space space is already defined\space}{}\@ehc
\fi
\else
\newcommand*{\magicnum}{}\fi

2.4 Without \LaTeXe

\begingroup\expandafter\expandafter\expandafter\endgroup
\ifx\csname directlua\endcsname\relax
\magicnum
\begingroup\expandafter\expandafter\expandafter\endgroup
\ifx\csname ifcsname\endcsname\relax
\def\magicnum#1{%
\expandafter\ifx\csname MG@#1\endcsname\relax
\else
\csname MG@#1\endcsname
\fi
}%
\else
\begingroup
\edef\x{\endgroup
\def\noexpand\magicnum##1{%
\expandafter\noexpand\csname
ifcsname\endcsname MG@##1\noexpand\endcsname
\noexpand\csname MG@##1%
\noexpand\expandafter\noexpand\endcsname
\expandafter\noexpand\csname fi\endcsname
}%
\x
\fi
\else
\begingroup
\def\x{%
\endgroup
}\x
\fi
\else
\begingroup\expandafter\expandafter\expandafter\endgroup
\edef\x{%
\expandafter\ifx\csname MG@#1\endcsname\relax
\else
\csname MG@#1\endcsname
\fi
}%
\x
\fi
\else
\endgroup
\fi

2.5 With \LaTeXe

\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname RequirePackage\endcsname\relax
\input ifluatex.sty\relax
\input infwarerr.sty\relax
\else
\RequirePackage{ifluatex}[2010/03/01]%
\RequirePackage{infwarerr}[2010/04/08]%
\fi
\magicnum@directlua

\ifnum\luatexversion<36 \%
\def\magicnum@directlua{%
\directlua0 \}%
\else
\let\magicnum@directlua\directlua
\fi
\magicnum@directlua%
require("oberdiek.magicnum")%
\begin{verbatim}
\def\x{2016/05/16 v1.5}
\def\StripPrefix#1>{}
\edef\x{\expandafter\StripPrefix\meaning\x}
\edef\y{\magicnum@directlua{
  if oberdiek.magicnum.getversion then 
    oberdiek.magicnum.getversion()
  end
}}
\ifx\x\y
\else
\@PackageError{magicnum}{Wrong version of lua module.\MessageBreak
Package version: \x\MessageBreak
Lua module: \y}
\fi
\endgroup
\luaescapestring
\begin{verbatim}
\expandafter\ifx\csname luaescapestring\endcsname\relax
\directlua{
  if tex.enableprimitives then 
    tex.enableprimitives{\magicnum@, \{\luaescapestring\}}
  end
}\fi
\expandafter\ifx\csname luaescapestring\endcsname\relax
\escapechar=92 \@PackageError{magicnum}{Missing \string\luaescapestring\MessageBreak
}\fi
\endgroup
\magicnum
\def\magicnum#1{\magicnum@directlua{
  oberdiek.magicnum.get("\luaescapestring{#1}")}
}\expandafter\magicnum@AtEnd\fi

\section{Data}
\subsection{Plain data}
\begin{verbatim}
tex.catcode
\endgroup
\end{verbatim}
\end{verbatim}
\end{verbatim}
\end{verbatim}
\end{verbatim}
\end{verbatim}
\end{verbatim}
\end{verbatim}

8
2.6.2 Data for TeX

\magicnum@add{tex.catcode}{escape}{0}
\magicnum@add{tex.catcode}{begingroup}{1}
\magicnum@add{tex.catcode}{endgroup}{2}
\magicnum@add{tex.catcode}{math}{3}
\magicnum@add{tex.catcode}{align}{4}
\magicnum@add{tex.catcode}{eol}{5}
\magicnum@add{tex.catcode}{parameter}{6}
\magicnum@add{tex.catcode}{superscript}{7}
\magicnum@add{tex.catcode}{subscript}{8}
\magicnum@add{tex.catcode}{ignore}{9}
\magicnum@add{tex.catcode}{space}{10}
2.6.3 Lua module

```lua
(*lua)
module(“oberdiek.magicnum”, package.seeall)
function getversion()
  tex.write(“2016/05/16 v1.5”)
end

local data = {
  [”tex.catcode”] = {
    [0] = “escape”,
    [1] = “begingroup”,
    [7] = “superscript”,
    [8] = “subscript”,
    [9] = “ignore”,
    [10] = “space”,
    [12] = “other”,
    [13] = “active”,
    [14] = “comment”,
    [”active”] = 13,
    [”align”] = 4,
    [”begingroup”] = 1,
    [”comment”] = 14,
    [”endgroup”] = 2,
    [”eol”] = 5,
    [”escape”] = 0,
    [”ignore”] = 9,
    [”invalid”] = 15,
    [”letter”] = 11,
    [”math”] = 3,
    [”other”] = 12,
    [”parameter”] = 6,
    [”space”] = 10,
    [”subscript”] = 8,
    [”superscript”] = 7
  },
  [”etex.grouptype”] = {
    [0] = “bottomlevel”,
    [1] = “simple”,
    [8] = “output”,
    [9] = “math”,
    [10] = “disc”,
}```
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.iftype"] = {
483  [0] = "none",
484  [1] = "char",
486  [3] = "num",
488  [5] = "odd",
489  [6] = "vmode",
490  [7] = "hmode",
491  [8] = "mmode",
492  [9] = "inner",
493  [10] = "void",
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,
function get(name)
  local startpos, endpos, category, entry =
    string.find(name, "^(%a[%a%d%.%]+)%(-?[%a%d]+)$")
  return "etex.nodetype"[startpos .. endpos],
    "etex.interactionmode"[startpos .. endpos],
    "luatex.pdfliteral.mode"[startpos .. endpos]
end
if not entry then
    return
end
local node = data[category]
if not node then
    return
end
local num = tonumber(entry)
local value
if num then
    value = node[num]
    if not value then
        return
    end
else
    value = node[entry]
    if not value then
        return
    end
    value = "" .. value
end
tex.write(value)
end

3 Test
3.1 Catcode checks for loading

\catcode`\{=1 %
\catcode`\}=2 %
\catcode`#=6 %
\catcode`@=11 %
\expandafter\ifx\csname count@\endcsname\relax
\countdef\count@=255 %
\fi
\expandafter\ifx\csname @gobble\endcsname\relax
\long\def\@gobble#1{}%
\fi
\expandafter\ifx\csname @firstofone\endcsname\relax
\long\def\@firstofone#1{#1}%
\fi
\expandafter\ifx\csname loop\endcsname\relax
\else
\expandafter\@gobble
\fi
{%
\def\loop#1\repeat{%
\def\body{#1}%
\iterate
}%
\def\iterate{%
\body
\let\next\iterate
\else
\let\next\relax
\fi
\next
}%
\let\repeat=\fi

15
\def\RestoreCatcodes{}
\count@=0 %
\loop
  \edef\RestoreCatcodes{\RestoreCatcodes\catcode\the\count@=\the\catcode\count@\relax}
  \ifnum\count@<255 %
    \advance\count@ 1 %
  \repeat
\def\RangeCatcodeInvalid#1#2{\count@=#1\relax
  \loop
    \catcode\count@=15 %
    \ifnum\count@<#2\relax
      \advance\count@ 1 %
    \repeat
  \ifnum\count@<#2\repeat
\def\RangeCatcodeCheck#1#2#3{\count@=#1\relax
  \loop
    \ifnum#3=\catcode\count@
    \else
      \errmessage{Character \the\count@ with wrong catcode \the\catcode\count@ instead of \number#3}%
    \fi
    \ifnum\count@<#2\relax
      \advance\count@ 1 %
    \repeat
\def\space{ }
\expandafter\ifx\csname LoadCommand\endcsname\relax
  \def\LoadCommand{\input magicnum.sty\relax}
\fi
\def\Test{%
  \RangeCatcodeInvalid{0}{47}%
  \RangeCatcodeInvalid{58}{64}%
  \RangeCatcodeInvalid{91}{96}%
  \RangeCatcodeInvalid{123}{255}%
  \catcode\@=12 %
  \catcode\\=0 %
}
3.2 Test data

\NeedsTeXFormat{LaTeX2e}
\documentclass{minimal}
\usepackage{magicnum}[2016/05/16]
\usepackage{qstest}
\IncludeTests{*}
\LogTests{log}{*}{*}
\newcommand*{\Test}{#1#2}{\Expect{\magicnum{#1}}{#2}}
\begin{qstest}{magicnum}{magicnum}
\Test{tex.catcode.escape}{0}
\Test{tex.catcode.invalid}{15}
\Test{tex.catcode.unknown}{}
\Test{tex.catcode.0}{escape}
\Test{tex.catcode.15}{invalid}
\Test{etex.iftype.true}{15}
\Test{etex.iftype.false}{16}
\Test{etex.iftype.15}{true}
\Test{etex.iftype.16}{false}
\Test{etex.nodetype.none}{-1}
\Test{etex.nodetype.-1}{none}
\Test{luatex.pdfliteral.mode.direct}{2}
\Test{luatex.pdfliteral.mode.1}{page}
\end{qstest}
3.3 Small test for init\TeX

\begin{verbatim}
\edef\x{\magicnum{tex.catcode.15}}
\edef\y{invalid}
\def\Strip#1>{}
\edef\y{\expandafter\Strip\meaning\y}
\ifx\x\y
\immediate\write16{Ok}\
\else
\errmessage{\x<\y}\
\fi
\end{verbatim}

4 Installation

4.1 Download

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


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.

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

\footnote{http://ctan.org/pkg/magicnum}
TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as \texttt{texmf} tree):

- \texttt{magicnum.sty} \rightarrow \texttt{tex/generic/oberdiek/magicnum.sty}
- \texttt{magicnum.lua} \rightarrow \texttt{scripts/oberdiek/magicnum.lua}
- \texttt{oberdiek.magicnum.lua} \rightarrow \texttt{scripts/oberdiek/oberdiek.magicnum.lua}
- \texttt{magicnum.pdf} \rightarrow \texttt{doc/latex/oberdiek/magicnum.pdf}
- \texttt{magicnum.txt} \rightarrow \texttt{doc/latex/oberdiek/magicnum.txt}
- \texttt{test/magicnum-test1.tex} \rightarrow \texttt{doc/latex/oberdiek/test/magicnum-test1.tex}
- \texttt{test/magicnum-test2.tex} \rightarrow \texttt{doc/latex/oberdiek/test/magicnum-test2.tex}
- \texttt{test/magicnum-test3.tex} \rightarrow \texttt{doc/latex/oberdiek/test/magicnum-test3.tex}
- \texttt{test/magicnum-test4.tex} \rightarrow \texttt{doc/latex/oberdiek/test/magicnum-test4.tex}
- \texttt{magicnum.dtx} \rightarrow \texttt{source/latex/oberdiek/magicnum.dtx}

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

4.4 Refresh file name databases

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

4.5 Some details for the interested

Unpacking with \LaTeX{}. The \texttt{.dtx} chooses its action depending on the format:

- \texttt{plain \TeX{}}: Run \texttt{docstrip} and extract the files.
- \texttt{\LaTeX{}}: Generate the documentation.

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

\begin{verbatim}
latex \let\install=y\input{magicnum.dtx}
\end{verbatim}

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

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

\begin{verbatim}
\PassOptionsToClass{a4paper}{article}
\end{verbatim}

An example follows how to generate the documentation with \texttt{pdflatex}:

\begin{verbatim}
pdflatex magicnum.dtx
makeindex -s gind.ist magicnum.idx
pdflatex magicnum.dtx
makeindex -s gind.ist magicnum.idx
pdflatex magicnum.dtx
\end{verbatim}

5 Catalogue

The following XML file can be used as source for the \TeX{} Catalogue. The elements \texttt{caption} and \texttt{description} are imported from the original XML file from the Catalogue. The name of the XML file in the Catalogue is \texttt{magicnum.xml}.

778 (*catalogue)
779 <!DOCTYPE entry SYSTEM ‘catalogue.dtd’>
This package allows access to the various parameter values in TeX (catcode values), eTeX (group, if and node types, and interaction mode), and LuaTeX (pdfliteral mode) by a hierarchical name system.

The package is part of the oberdiek bundle.

6 History

[2007/12/12 v1.0]

• First public version.

[2009/04/10 v1.1]

• Adaptation to LuaTeX 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 iniTeX.
• Dependency from package luatex removed.
• Version check for lua module.

[2016/05/16 v1.5]

• Documentation updates.

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

Symbols

\#...............................611, 765
<table>
<thead>
<tr>
<th>T</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>\Test</td>
<td>\write</td>
</tr>
<tr>
<td>680, 703, 709, 730, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752</td>
<td>23, 52, 772</td>
</tr>
<tr>
<td>\the</td>
<td>\x</td>
</tr>
<tr>
<td>77, 78, 79, 80, 81, 82, 83, 84, 97, 647, 667, 668</td>
<td>14, 15, 18, 22, 26, 28, 51, 56, 66, 75, 87, 140, 149, 169, 171, 179, 183, 767, 771, 774</td>
</tr>
<tr>
<td>\TMP@EnsureCode</td>
<td>\y</td>
</tr>
<tr>
<td>94, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114</td>
<td>172, 179, 184, 768, 770, 771, 774</td>
</tr>
<tr>
<td>\usepackage</td>
<td></td>
</tr>
<tr>
<td>726, 727</td>
<td></td>
</tr>
</tbody>
</table>