The **luatex** package

Heiko Oberdiek*

<*heiko.oberdiek at googlemail.com*>

2016/05/16 v0.6

Abstract

This package manages the new and extended features and resources that LuaTeX provides. Examples are attributes and catcode tables.

Contents

---

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

1 Documentation

1.1 Introduction

TeX provides global resources such as registers. But it does not provide an interface for managing these resources. For example, two packages want to use a counter register. If they take the same register number, then the use of both packages will conflict and they cannot be used together. Therefore formats such as plain TeX or LaTeX implement an allocation scheme for registers. A package reserves with `\newcount` an unused register number for its own exclusive use.

Nowadays TeX is not alone anymore: ε-TeX, pdfTeX and other compilers for TeX are developed that extend and add new features and resources.

Now LuaTeX has reached beta state. It inherits most of pdfTeX's features including ε-TeX. Also it implements new concepts such as attributes or catcode tables.

1.1.1 LaTeX

Since 2015 LuaLaTeX includes support for luatex by default and so this package is essentially obsolete now, however it is kept for backwards compatibility.

1.1.2 plain TeX

LaTeX has inherited its resource handling from plain TeX. The interface is basically the same: `\newcount`, …Therefore this package tries to follow this tradition by providing compatibility to plain TeX. It can be loaded with plain TeX and defines at least some of the features that this packages provides for LaTeX.

1.2 Register allocation

1.2.1 Register with 16 bit

Because LuaTeX is a super set of ε-TeX regarding registers, the register allocation scheme should not conflict with package `etex`. Therefore this package is loaded to inherit its allocation scheme. The only change is currently that the limit is increased to 65536 registers for the following register classes:

- count
- dimen
- skip
- muskip
- marks
- toks
- box

This affects the number of global and local registers. Because it is done in a package and not in the kernel, it is possible that someone loads package `etex` before uses the local allocation variants. This will prevent the extension for this register class. If more registers are needed, just load package `luatex` earlier.

1.2.2 Insertions

Insertions need four registers `\count`, `\dimen`, `\skip`, and `\box` with the same number. Usually they are allocated downwards from 254, 253, ...Also `\newcount`, `\newdimen`, ...fill up these register numbers from below before switching to higher register numbers by package `etex`. When this occurs, no insertions can be allocated anymore.

Therefore `\newcount`, `\newdimen`, `\newskip`, and `\newbox` are replaced by their global variants (`\globcount`, ...) that use the higher numbers immediately, leaving the room for insertions. There should not be an efficiency penalty because LuaTeX stores the registers of a class in the same Lua table unlike ε-TeX, where registers below 256 are stored in an array and higher numbers are put in a tree structure.

1.3 Attributes

Nodes can have custom attributes in LuaTeX. These attributes are organized by a new register class. As the other registers up to 2^{16} attributes are supported. An attribute value can be negative that means the attribute is not set. Otherwise TeX's range of non-negative integers up to 2^31 are available.

`\newattribute {⟨cmd⟩}`

Macro `\newattribute` defines command ⟨cmd⟩ using `\attributedef` using an new attribute number. The new attribute is initially unset.

`\setattribute {⟨cmd⟩} {⟨value⟩}`

Macro `\setattribute` locally sets attribute command ⟨cmd⟩ to the number ⟨value⟩. Valid values range from −1 until 2^31 (the upper limit is the same as for other TeX integer numbers).

`\unsetattribute {⟨cmd⟩}`

Macro `\unsettattribute` clears the attribute command ⟨cmd⟩.

1.4 Catcode tables

LuaTeX introduces catcode tables as new feature, see documentation. There is need for discussion, how to deal best:

- \initcatcodetable and \setcatcodetable act globally.

- \catcodetable causes an error if used with an uninitialized catcode table.

- Large catcode table numbers should be avoided because of performance breakdown.

- Use case LaTeX package: The package must not be surprised by changed catcodes and must not surprise by changing catcodes accidently. Catcode tables could offer a solution. At the begin a catcode regime with standard catcodes is established and the old one is restored afterwards.

- Use case: LuaTeX's tex.print might be used with a catcode table number, for example a table where all entries have catcode "other".

- Readonly catcode tables.

- Is there is a need for local allocations? (Package etex's \loc variants are not used in TeX Live 2007.)

1.4.1 Interface proposal

The idea: \newcatcodetable allocates odd numbered catcode tables. Even numbered tables are managed as stack. Also some catcode tables are defined. These must not be changed.

\newcatcodetable {⟨cmd⟩}

Macro \newcatcodetable reserves a new catcode table and remembers its number in ⟨cmd⟩. The catcode table is initialized with ini-TeX's catcodes.

\CatcodeTableIniTeX
\CatcodeTableString
\CatcodeTableOther
\CatcodeTableLaTeX

These are catcode tables and must not be changed. \CatcodeTableIniTeX contains the catcode settings of ini-TeX. \CatcodeTableString follows TeX's convention of \string, \meaning and friends. The space gets catcode 10 (space), the other characters have catcode 12 (other). In \CatcodeTableOther all entries have catcode 12 (other). \CatcodeTableLaTeX contains the setting of a pure LaTeX format ('at' is other).

\CatcodeTableStack
\IncCatcodeTableStack
\DecCatcodeTableStack

\CatcodeTableStack is the stack pointer. Initially it is catcode table zero. \IncCatcodeTableStack and \DecCatcodeTableStack increments and decrements the stack pointer. Currently \IncCatcodeTableStack does not initialize a new catcode table. Both increment and decrement operations do not set a catcode table.

```
\PushCatcodeTableNumStack
\PopCatcodeTableNumStack
```

It can be handy to have a global stack for catcode table numbers to deal with the global assignment property of \initcatcodetable and \savecatcodetable. \Push-CatcodeTableNumStack pushes the current catcode table on the stack. \Pop-CatcodeTableNumStack pops the topmost number off the number stack to set the current catcode table. Catcode table zero is used in case of an empty stack.

```
\BeginCatcodeRegime {⟨catcodetable⟩}
\EndCatcodeRegime
```

\BeginCatcodeRegime remembers the current catcode table number. Then it creates and uses a fresh catcode table on the stack that is initialized by ⟨*catcodetable*⟩:

```
\PushCatcodeTableNumStack
\catcodetable⟨catcodetable⟩ \IncCatcodeTableStack
\savecatcodetable\CatcodeTableStack
\catcodetable\CatcodeTableStack
```

\EndCatcodeRegime drops the catcode table, created by \BeginCatcodeRegime and sets the catcode table that was active before:

```
\DecCatcodeTableStack
\PopCatcodeTableNumStack
```

These macros solve the use case, described earlier for a LaTeX package:

```
% package foobar.sty
\BeginCatcodeRegime\CatcodeTableLaTeX
\makeatletter
% ... package contents ...
\EndCatcodeRegime
% end of package
```

If the package wants to change catcodes after its loading, \AtBeginDocument or \AtEndOfPackage can be used.

```
\SetCatcodeRange {⟨from⟩} {⟨to⟩} {⟨catcode⟩}
```

The catcodes of characters in range from ⟨*from*⟩ to inclusive ⟨*to*⟩ are set to ⟨*catcode*⟩.

1.5 Lua module loading

Currently LuaTeX (version 0.20) does not support Lua script files inside `TDS:scripts//`, because Lua's mechanism for module loading does not use the `kpathsea` library. Therefore this packages appends a kpse loader to the list of Lua's module loaders. It finds the module ⟨*module*⟩ by

```
kpse.find_file("⟨module⟩.lua", "texmfscripts")
```

Unhappily `kpathsea` does not support directory components in a file name. Therefore the Lua convention is not followed to replace dots in the module name by the directory separator.

Example: A Lua script of a package foobar wants the following modules:

```
require("foobar.hello.world")
require("org.somewhere.xyz")
```

Then they can be find in:

> TDS:scripts/foobar/foobar.hello.world.lua
> TDS:scripts/foobar/org.somewhere.xyz.lua

I would have preferred the following locations, following lua conventions, e. g.:

> TDS:scripts/foobar/hello/world.lua
> TDS:scripts/foobar/org/somewhere/xyz.lua

But I do not know, how to achieve this in a reliable way using kpathsea.

1.5.1 Package luatex-loader

If someone do not need or want package luatex but it's extension for module loading, then he can use package luatex-loader. Both plain TeX and LaTeX are supported.

2 Implementation

1 ⟨*package⟩

2.1 Reload check and package identification

Reload check, especially if the package is not used with LaTeX.

```
 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@luatex.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{luatex}{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@luatex.sty\endcsname
67 \ProvidesPackage{luatex}%
68   [2016/05/16 v0.6 LuaTeX basic definition package (HO)]%
```

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 LuT@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\LuT@AtEnd{%
96     \LuT@AtEnd
97     \catcode#1=\the\catcode#1\relax
98 }%
```

```
 99   \catcode#1=#2\relax
100 }
101 \TMP@EnsureCode{10}{12}% ^^J
102 \TMP@EnsureCode{34}{12}% "
103 \TMP@EnsureCode{36}{3}% $
104 \TMP@EnsureCode{39}{12}% '
105 \TMP@EnsureCode{40}{12}% (
106 \TMP@EnsureCode{41}{12}% )
107 \TMP@EnsureCode{42}{12}% *
108 \TMP@EnsureCode{43}{12}% +
109 \TMP@EnsureCode{44}{12}% ,
110 \TMP@EnsureCode{45}{12}% -
111 \TMP@EnsureCode{46}{12}% .
112 \TMP@EnsureCode{47}{12}% /
113 \TMP@EnsureCode{60}{12}% <
114 \TMP@EnsureCode{62}{12}% >
115 \TMP@EnsureCode{91}{12}% [
116 \TMP@EnsureCode{93}{12}% ]
117 \TMP@EnsureCode{95}{12}% _ (other!)
118 \TMP@EnsureCode{96}{12}% `
119 \edef\LuT@AtEnd{\LuT@AtEnd\noexpand\endinput}
```

2.3   Check for LuaTEX

Without LuaTEX there is no point in using this package.

```
120 \begingroup\expandafter\expandafter\expandafter\endgroup
121 \expandafter\ifx\csname RequirePackage\endcsname\relax
122   \input infwarerr.sty\relax
123   \input ifluatex.sty\relax
124 \else
125   \RequirePackage{infwarerr}[2007/09/09]%
126   \RequirePackage{ifluatex}[2009/04/10]%
127 \fi
128 \ifluatex
129 \else
130   \@PackageError{luatex}{%
131     This package may only be run using LuaTeX%
132   }\@ehc
133   \expandafter\LuT@AtEnd
134 \fi%
```

2.4   Provide LuaTEX primitives

```
135 \ifnum\luatexversion<36 %
136 \def\LuT@MakePrimitive#1{%
137   \expandafter\let\csname luatex#1\expandafter\endcsname
138   \csname #1\endcsname
139 }%
140 \else
141 \def\LuT@MakeLuatexPrimitive#1{%
142   \begingroup\expandafter\expandafter\expandafter\endgroup
143   \expandafter\ifx\csname luatex#1\endcsname\relax
144     \begingroup\expandafter\expandafter\expandafter\endgroup
145     \expandafter\ifx\csname #1\endcsname\relax
146     \else
147       \expandafter\let
148       \csname luatex#1\expandafter\endcsname
149       \csname #1\endcsname
150     \fi
151   \fi
152   \begingroup\expandafter\expandafter\expandafter\endgroup
153   \expandafter\ifx\csname luatex#1\endcsname\relax
```

8

```
154    \begingroup
155      \expandafter\let\csname luatex#1\endcsname\@undefined
156      \ifnum0%
157        \directlua{%
158          if tex.enableprimitives then %
159            tex.enableprimitives('luatex',{'#1'})%
160            tex.print('1')%
161          end%
162        }%
163        \expandafter\ifx\csname luatex#1\endcsname\relax\else1\fi
164      =11 %
165        \global\expandafter\let
166        \csname luatex#1\expandafter\endcsname
167        \csname luatex#1\endcsname
168      \else
169        \@PackageError{luatex}{%
170          tex.enableprimitives failed for `#1'%
171        }\@ehc
172      \fi
173      \endgroup
174    \fi
175  }%
176  \def\LuT@MakePrimitive#1{%
177    \begingroup\expandafter\expandafter\expandafter\endgroup
178    \expandafter\ifx\csname#1\endcsname\relax
179      \begingroup
180        \expandafter\let\csname#1\endcsname\@undefined
181        \ifnum0%
182          \directlua{%
183            if tex.enableprimitives then %
184              tex.enableprimitives('',{'#1'})%
185              tex.print('1')%
186            end%
187          }%
188          \expandafter\ifx\csname#1\endcsname\relax\else1\fi
189        =11 %
190          \global\expandafter\let
191          \csname#1\expandafter\endcsname
192          \csname#1\endcsname
193        \else
194          \@PackageError{luatex}{%
195            tex.enableprimitives failed for `#1'%
196          }\@ehc
197        \fi
198      \endgroup
199    \fi
200  }%
201  \fi
202  \LuT@MakeLuatexPrimitive{attribute}
203  \LuT@MakeLuatexPrimitive{attributedef}
204  \LuT@MakeLuatexPrimitive{catcodetable}
205  \LuT@MakeLuatexPrimitive{initcatcodetable}
206  \LuT@MakeLuatexPrimitive{luaescapestring}
207  \LuT@MakeLuatexPrimitive{savecatcodetable}
208  \LuT@MakePrimitive{numexpr}
```

2.5 Inherit support for ε-TeX

Package etex is not compatible for plain TeX. But it could be present if a format is used that is based on etex.src. Therefore we only load the package in case of LaTeX and tests its presence independently of the format by looking for \et@xins.

```
209 \begingroup\expandafter\expandafter\expandafter\endgroup
210 \expandafter\ifx\csname RequirePackage\endcsname\relax
```

```
211 \else
212   \RequirePackage{etex}[1998/03/26]%
213 \fi
```

2.6 Adaption of ε-TeX's register allocation

ε-TeX has increased the number of TeX registers from 2^8 (256) to 2^{15} (32768) for a register class. LuaTeX extends the limit further to 2^16 (65536). The allocation scheme of package etex is not changed. But this can be subject for discussion.

If a register class hasn't registered any local registers yet, then the limit can safely be pushed to 65536.

```
214 \begingroup\expandafter\expandafter\expandafter\endgroup
215 \expandafter\ifx\csname et@xins\endcsname\relax
216   \@PackageWarningNoLine{luatex}{%
217     Support for eTeX is not loaded (etex.src)%
218   }%
219 \else
220   \def\LuT@temp#1{%
221     \ifnum\count27#1=32768 %
222       \count27#1=65536 %
223     \fi
224   }%
225   \LuT@temp0%
226   \LuT@temp1%
227   \LuT@temp2%
228   \LuT@temp3%
229   \LuT@temp4%
230   \LuT@temp5%
231   \LuT@temp6%
```

ε-TeX uses an array for the first 256 registers and then a tree structure. LuaTeX stores all registers of a class in one Lua table. There shouldn't be large performance differences. This allows starting immediately in the extended area, leaving room for insertions.

```
232   \let\newcount\globcount
233   \let\newdimen\globdimen
234   \let\newskip\globskip
235   \let\newbox\globbox
236 \fi
```

2.7 plain TeX compatibility

\@empty

```
237 \expandafter\ifx\csname @empty\endcsname\relax
238   \def\@empty{}%
239 \fi
```

\@gobble

```
240 \expandafter\ifx\csname @gobble\endcsname\relax
241   \long\def\@gobble#1{}%
242 \fi
```

\@firstofone

```
243 \expandafter\ifx\csname @firstofone\endcsname\relax
244   \long\def\@firstofone#1{#1}%
245 \fi
```

\@firstoftwo

```
246 \expandafter\ifx\csname @firstoftwo\endcsname\relax
247   \long\def\@firstoftwo#1#2{#1}%
248 \fi
```

10

\@car

```
249 \expandafter\ifx\csname @car\endcsname\relax
250   \def\@car#1#2\@nil{#1}%
251 \fi
```

\@cdr

```
252 \expandafter\ifx\csname @cdr\endcsname\relax
253   \def\@cdr#1#2\@nil{#2}%
254 \fi
```

\@ifstar

```
255 \expandafter\ifx\csname @ifstar\endcsname\relax
256   \def\@ifstar#1{%
257     \@ifnextchar*{\@firstoftwo{#1}}%
258   }%
```

\@ifnextchar

```
259   \long\def\@ifnextchar#1#2#3{%
260     \let\reserved@d=#1%
261     \def\reserved@a{#2}%
262     \def\reserved@b{#3}%
263     \futurelet\@let@token\@ifnch
264   }%
```

\@ifnch

```
265   \def\@ifnch{%
266     \ifx\@let@token\@sptoken
267       \let\reserved@c\@xifnch
268     \else
269       \ifx\@let@token\reserved@d
270         \let\reserved@c\reserved@a
271       \else
272         \let\reserved@c\reserved@b
273       \fi
274     \fi
275     \reserved@c
276   }%
```

\@sptoken

```
277   \let\LuT@temp\:%
278   \def\:{\let\@sptoken= }%
279   \: % explicit space
```

\@xifnch

```
280   \def\:{\@xifnch}%
281   \expandafter\def\: {%
282     \futurelet\@let@token\@ifnch
283   }%
284   \let\:\LuT@temp
285 \fi
```

\@tempcnta

```
286 \expandafter\ifx\csname @tempcnta\endcsname\relax
287   \csname newcount\endcsname\@tempcnta
288 \fi
```

\@tempcntb

```
289 \expandafter\ifx\csname @tempcntb\endcsname\relax
290   \csname newcount\endcsname\@tempcntb
291 \fi
```

\LuT@newcommand

```
292 \begingroup\expandafter\expandafter\expandafter\endgroup
293 \expandafter\ifx\csname newcommand\endcsname\relax
294 \def\LuT@newcommand#1[#2]#3{%
295   \ifx#1\@undefined
296     \let#1\relax
297   \else
298     \ifx#1\relax
299     \else
300       \@PackageError{luatex}{%
301         \string#1 is already defined.\MessageBreak
302         Redefinition is skipped%
303       }\@ehc
304     \fi
305   \fi
306   \ifx#1\relax
307     \ifcase#2 %
308       \def#1{#3}%
309     \or
310       \def#1##1{#3}%
311     \or
312       \def#1##1##2{#3}%
313     \or
314       \def#1##1##2##3{#3}%
315     \or
316       \@INTERNAL@ERROR
317     \fi
318   \fi
319 }%
320 \else
321   \def\LuT@newcommand{\newcommand*}%
322 \fi
```

2.8 Attributes

2.8.1 Allocation

\LuT@AllocAttribute

```
323 \newcount\LuT@AllocAttribute
324 \LuT@AllocAttribute=\m@ne
```

\newattribute

```
325 \ifx\newattribute\@undefined
326 \LuT@newcommand\newattribute[1]{%
327   \ifnum\LuT@AllocAttribute<65535 %
328     \global\advance\LuT@AllocAttribute\@ne
329     \allocationnumber\LuT@AllocAttribute
330     \global\luatexattributedef#1=\allocationnumber
331     \unsetattribute{#1}%
332     \wlog{\string#1=\string\attribute\the\allocationnumber}%
333   \else
334     \errmessage{No room for a new \string\attribute}%
335   \fi
336 }
337 \fi
```

2.8.2 Interface

\setattribute

```
338 \ifx\setattribute\@undefined
339 \LuT@newcommand\setattribute[2]{%
340   #1=\numexpr#2\relax
```

```
341 }
342 \fi
```

\unsetattribute

```
343 \ifx\unsetattribute\@undefined
344 \ifnum\luatexversion<37
345   \LuT@newcommand\LuT@UnsetAttributeValue[0]{}%
346   \let\LuT@UnsetAttributeValue\m@ne
347 \else
348   \LuT@newcommand\LuT@UnsetAttributeValue[0]{-2147483647 }%
349 \fi
350 \LuT@newcommand\unsetattribute[1]{%
351   #1=\LuT@UnsetAttributeValue
352 }
353 \fi
```

2.9 Catcode tables

2.9.1 Allocation

\LuT@AllocCatcodeTable

```
354 \newcount\LuT@AllocCatcodeTable
355 \LuT@AllocCatcodeTable=\m@ne
356 \newcount\CatcodeTableStack
357 \CatcodeTableStack=\z@
```

\newcatcodetable

```
358 \ifx\newcatcodetable\@undefined
359 \LuT@newcommand\newcatcodetable[1]{%
360   \ifnum\LuT@AllocCatcodeTable<1114110 % 0x10FFFF is maximal \chardef
361     % or < 268435455 % 2^28 - 1
362     \global\advance\LuT@AllocCatcodeTable by\tw@
363     \allocationnumber=\LuT@AllocCatcodeTable
364     \global\chardef#1=\allocationnumber
365     \wlog{%
366       \string#1=\string\catcodetable\the\allocationnumber
367     }%
368   \else
369     \errmessage{No room for a new \string\catcodetable}%
370   \fi
371 }%
372 \fi
```

\IncCatcodeTableStack

```
373 \LuT@newcommand\IncCatcodeTableStack[0]{%
374   \ifnum\CatcodeTableStack<268435454 %
375     \global\advance\CatcodeTableStack by\tw@
376   \else
377     \@PackageError{luatex}{%
378       Catcode table stack overflow%
379     }\@ehd
380   \fi
381 }
```

\DecCatcodeTableStack

```
382 \LuT@newcommand\DecCatcodeTableStack[0]{%
383   \ifnum\CatcodeTableStack>\z@
384     \global\advance\CatcodeTableStack by-2 %
385   \else
386     \@PackageError{luatex}{%
387       Catcode table stack is empty%
388     }\@ehd
```

```
389   \fi
390 }
```

2.9.2 \SetCatcodeRange

```
391 \LuT@newcommand\SetCatcodeRange[3]{%
392 \edef\LuT@temp{%
393   \noexpand\@tempcnta=\the\@tempcnta
394   \noexpand\@tempcntb=\the\@tempcntb
395   \noexpand\count@=\the\count@
396   \relax
397 }%
398 \@tempcnta=\numexpr#1\relax
399 \@tempcntb=\numexpr#2\relax
400 \count@=\numexpr#3\relax
401 \loop
402   \unless\ifnum\@tempcnta>\@tempcntb
403   \catcode\@tempcnta=\count@
404   \advance\@tempcnta by \@ne
405 \repeat
406 \LuT@temp
407 }
```

2.9.3 Predefined catcode tables

```
408 \newcatcodetable\CatcodeTableIniTeX
409 \newcatcodetable\CatcodeTableString
410 \newcatcodetable\CatcodeTableOther
411 \newcatcodetable\CatcodeTableLaTeX

412 \luatexinitcatcodetable\CatcodeTableIniTeX
413 \begingroup
414   \def\@makeother#1{\catcode#1=12\relax}%
415   \@firstofone{%
416     \luatexcatcodetable\CatcodeTableIniTeX
417     \begingroup
418     \SetCatcodeRange{0}{8}{15}%
419     \catcode9=10 % tab
420     \catcode11=15 %
421     \catcode12=13 % form feed
422     \SetCatcodeRange{14}{31}{15}%
423     \catcode35=6 % hash
424     \catcode36=3 % dollar
425     \catcode38=4 % ampersand
426     \catcode94=7 % circumflex
427     \catcode95=8 % underscore
428     \catcode123=1 % brace left
429     \catcode125=2 % brace right
430     \catcode126=13 % tilde
431     \catcode127=15 %
432     \luatexsavecatcodetable\CatcodeTableLaTeX
433     \endgroup
434     \@makeother{0}% nul
435     \@makeother{13}% carriage return
436     \@makeother{37}% percent
437     \@makeother{92}% backslash
438     \@makeother{127}%
439     \SetCatcodeRange{65}{90}{12}% A-Z
440     \SetCatcodeRange{97}{122}{12}% a-z
441     \luatexsavecatcodetable\CatcodeTableString
442     \@makeother{32}% space
443     \luatexsavecatcodetable\CatcodeTableOther
```

```
444  \endgroup
445 }%
```

2.9.4 Number stack

\LuT@NumStackEmpty   A special empty stack value because of \@cdr's brace removal.

```
446 \def\LuT@NumStackEmpty{0}
```

\LuT@NumStack

```
447 \let\LuT@NumStack\LuT@NumStackEmpty
```

\PushCatcodeTableNumStack

```
448 \LuT@newcommand\PushCatcodeTableNumStack[0]{%
449  \xdef\LuT@NumStack{%
450    {\the\luatexcatcodetable}\LuT@NumStack
451  }%
452 }
```

\PopCatcodeTableNumStack

```
453 \LuT@newcommand\PopCatcodeTableNumStack[0]{%
454  \ifx\LuT@NumStack\LuT@NumStackEmpty
455    \@PackageWarning{luatex}{Empty catcode table number stack}%
456    \luatexcatcodetable\z@
457  \else
458    \luatexcatcodetable=\expandafter\@car\LuT@NumStack\@nil\relax
459    \xdef\LuT@NumStack{%
460      \expandafter\@cdr\LuT@NumStack\@nil
461    }%
462  \fi
463 }
```

2.9.5 Catcode regime macros

\BeginCatcodeRegime

```
464 \LuT@newcommand\BeginCatcodeRegime[1]{%
465  \PushCatcodeTableNumStack
466  \luatexcatcodetable=\numexpr#1\relax
467  \IncCatcodeTableStack
468  \luatexsavecatcodetable\CatcodeTableStack
469  \luatexcatcodetable\CatcodeTableStack
470 }
```

\EndCatcodeRegime

```
471 \LuT@newcommand\EndCatcodeRegime[0]{%
472  \DecCatcodeTableStack
473  \PopCatcodeTableNumStack
474 }
```

2.10 Lua module loader

```
475 \begingroup\expandafter\expandafter\expandafter\endgroup
476 \expandafter\ifx\csname RequirePackage\endcsname\relax
477  \input luatex-loader.sty\relax
478 \else
479  \RequirePackage{luatex-loader}[2010/03/09]%
480 \fi
481 \LuT@AtEnd%
482 ⟨/package⟩
483 ⟨*loader⟩
```

Reload check, especially if the package is not used with LaTeX.

```
484 \begingroup\catcode61\catcode48\catcode32=10\relax%
```

```
485 \catcode13=5 % ^^M
486 \endlinechar=13 %
487 \catcode35=6 % #
488 \catcode39=12 % '
489 \catcode44=12 % ,
490 \catcode45=12 % -
491 \catcode46=12 % .
492 \catcode58=12 % :
493 \catcode64=11 % @
494 \catcode123=1 % {
495 \catcode125=2 % }
496 \expandafter\let\expandafter\x\csname ver@luatex-loader.sty\endcsname
497 \ifx\x\relax % plain-TeX, first loading
498 \else
499   \def\empty{}%
500   \ifx\x\empty % LaTeX, first loading,
501     % variable is initialized, but \ProvidesPackage not yet seen
502   \else
503   \expandafter\ifx\csname PackageInfo\endcsname\relax
504     \def\x#1#2{%
505       \immediate\write-1{Package #1 Info: #2.}%
506     }%
507   \else
508     \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
509   \fi
510   \x{luatex-loader}{The package is already loaded}%
511   \aftergroup\endinput
512   \fi
513 \fi
514 \endgroup%
```

Package identification:
```
515 \begingroup\catcode61\catcode48\catcode32=10\relax%
516 \catcode13=5 % ^^M
517 \endlinechar=13 %
518 \catcode35=6 % #
519 \catcode39=12 % '
520 \catcode40=12 % (
521 \catcode41=12 % )
522 \catcode44=12 % ,
523 \catcode45=12 % -
524 \catcode46=12 % .
525 \catcode47=12 % /
526 \catcode58=12 % :
527 \catcode64=11 % @
528 \catcode91=12 % [
529 \catcode93=12 % ]
530 \catcode123=1 % {
531 \catcode125=2 % }
532 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
533   \def\x#1#2#3[#4]{\endgroup
534     \immediate\write-1{Package: #3 #4}%
535     \xdef#1{#4}%
536   }%
537 \else
538   \def\x#1#2[#3]{\endgroup
539     #2[{#3}]%
540     \ifx#1\@undefined
541       \xdef#1{#3}%
542     \fi
543     \ifx#1\relax
544       \xdef#1{#3}%
545     \fi
```

```
546    }%
547  \fi
548  \expandafter\x\csname ver@luatex-loader.sty\endcsname
549  \ProvidesPackage{luatex-loader}%
550    [2016/05/16 v0.6 Lua module loader (HO)]%
551  \begingroup\catcode61\catcode48\catcode32=10\relax%
552    \catcode13=5\endlinechar=13\relax%
553    \catcode10=12 % ^^J
554    \catcode34=12 % "
555    \catcode39=12 % '
556    \catcode40=12 % (
557    \catcode41=12 % )
558    \catcode44=12 % ,
559    \catcode46=12 % .
560    \catcode60=12 % <
561    \catcode61=12 % =
562    \catcode95=12 % _ (other!)
563    \catcode96=12 % `
564    \catcode123=1 % {
565    \catcode125=2 % }
566    \endlinechar=10 %
567    \ifnum\luatexversion<36 %
568      \directlua0%
569    \else %
570      \expandafter\directlua %
571    \fi %
572    {%
573      do
574      local script = "oberdiek.luatex.lua"
575      local file = kpse.find_file(script, "texmfscripts")
576      if file then
577        texio.write_nl("(" .. file .. ")")
578        dofile(file)
579      else
580        error("File `" .. script .. "' not found")
581      end
582    end
583    }%
584  \endgroup\endinput%
585  ⟨/loader⟩
```

2.11 Lua script

Currently LuaTeX does not use KPSE when searching for module files. The follow-
ing Lua script implements a workaround. It extends `package.loader` by another
search method. Modules are found by the module name with extension `.lua` similar
to

kpsewhich --format=texmfscripts ⟨*module*⟩.lua

Unhappily `kpsewhich` does not support directory components in the file name.
Therefore a module `a.b.c` cannot be installed as `a/b/c.lua`. The script must be
named `a.b.c.lua`.

```
586  ⟨*lua⟩

587  module("oberdiek.luatex", package.seeall)

588  function kpse_module_loader(module)
589    local script = module .. ".lua"
590    local file = kpse.find_file(script, "texmfscripts")
591    if file then
592      local loader, error = loadfile(file)
593      if loader then
594        texio.write_nl("(" .. file .. ")")
```

```
595       return loader
596     end
597   return "\n\t[oberdiek.luatex.kpse_module_loader] Loading error:\n\t"
598         .. error
599   end
600   return "\n\t[oberdiek.luatex.kpse_module_loader] Search failed"
601 end
```

following line changed to use package.searchers instead of package.loaders for current luatex; this is the only change in the code. eroux, 28apr13, (or package.loaders), 29mar14.

```
602 table.insert(package.searchers or package.loaders, kpse_module_loader)
```

```
603 ⟨/lua⟩
```

3   Test

```
604 ⟨*test2⟩
605 \documentclass{article}
606 \def\LoadCommand{%
607   \RequirePackage{luatex}[2010/03/09]%
608 }
609 ⟨/test2⟩
```

```
610 ⟨*test3⟩
611 \documentclass{article}
612 \def\LoadCommand{%
613   \RequirePackage{luatex-loader}[2010/03/09]%
614 }
615 ⟨/test3⟩
```

3.1   Catcode checks for loading

```
616 ⟨*test1⟩
```

```
617 \catcode`\{=1 %
618 \catcode`\}=2 %
619 \catcode`\#=6 %
620 \catcode`\@=11 %
621 \expandafter\ifx\csname count@\endcsname\relax
622   \countdef\count@=255 %
623 \fi
624 \expandafter\ifx\csname @gobble\endcsname\relax
625   \long\def\@gobble#1{}%
626 \fi
627 \expandafter\ifx\csname @firstofone\endcsname\relax
628   \long\def\@firstofone#1{#1}%
629 \fi
630 \expandafter\ifx\csname loop\endcsname\relax
631   \expandafter\@firstofone
632 \else
633   \expandafter\@gobble
634 \fi
635 {%
636   \def\loop#1\repeat{%
637     \def\body{#1}%
638     \iterate
639   }%
640   \def\iterate{%
641     \body
642       \let\next\iterate
643     \else
644       \let\next\relax
645     \fi
```

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

```
708   \RangeCatcodeCheck{123}{255}{15}%
709   \RestoreCatcodes
710 }
711 \Test
712 \csname @@end\endcsname
713 \end

714 ⟨/test1⟩
```

3.2 Catcode tables

3.2.1 Predefined catcode tables

```
715 ⟨*test4⟩
716 \NeedsTeXFormat{LaTeX2e}
```

Remember LaTeX's initial catcodes in count registers starting at \TestLaTeX.

```
717 \count0=0 %
718 \chardef\TestLaTeX=1000 %
719 \chardef\TestMax=300 %
720 \loop
721   \count\numexpr\TestLaTeX+\count0\relax=\catcode\count0 %
722 \ifnum\count0<\TestMax
723   \advance\count0 by 1 %
724 \repeat
725 \documentclass{minimal}
726 \usepackage{luatex}[2010/03/09]
727 \usepackage{qstest}
728 \IncludeTests{*}
729 \LogTests{log}{*}{*}
730 \makeatletter
731 \def\Check#1{%
732   \Expect*{\the\count@=\the\catcode\count@}%
733       *{\the\count@=#1}%
734 }
735 \newcount\scratch
736 \def\Test#1#2{%
737   \begin{qstest}{CatcodeTable#1}{CatcodeTable#1}%
738     \luatexcatcodetable\csname CatcodeTable#1\endcsname
739     \count@=\z@
740     \loop
741       \scratch=#2\relax
742       \Expect*{\the\count@=\the\catcode\count@}%
743           *{\the\count@=\the\scratch}%
744     \ifnum\count@<\TestMax
745       \advance\count@\@ne
746     \repeat
747   \end{qstest}%
748 }
749 \begingroup
750   % luatex-unicode-letters.tex makes some slots to letters
751   \def\TestMax{169}%
752   \Test{LaTeX}{\the\count\numexpr\TestLaTeX+\count@}%
753 \endgroup
754 \Test{String}{\ifnum\count@=32 10\else 12\fi}
755 \Test{Other}{12}
756 \luatexinitcatcodetable99 %
757 \Test{IniTeX}{%
758   0\relax
759   \begingroup
760     \luatexcatcodetable99 %
761     \global\scratch=\the\catcode\count@
762   \endgroup
763 }
```

3.2.2 Catcode table number stack

```
764 \begin{qstest}{CatcodeTableNumStack}{CatcodeTableNumStack}
765 \def\TestStack#1{%
766   \Expect*{\LuT@NumStack}{#1}%
767 }%
768 \TestStack{0}%
769 \PushCatcodeTableNumStack
770 \TestStack{{0}0}%
771 \@firstofone{%
772   \begingroup
773     \luatexinitcatcodetable12 %
774     \luatexcatcodetable12 %
775     \PushCatcodeTableNumStack
776     \TestStack{{12}{0}0}%
777     \PopCatcodeTableNumStack
778     \TestStack{{0}0}%
779     \PopCatcodeTableNumStack
780     \TestStack{0}%
781     \def\TestWarning{Missing empty stack warning}%
782     \def\@PackageWarning#1#2{\def\TestWarning{empty stack}}%
783     \PopCatcodeTableNumStack
784     \TestStack{0}%
785     \Expect*{\TestWarning}{empty stack}%
786   \endgroup
787 }%
788 \end{qstest}
```

3.2.3 Catcode table stack

```
789 \begin{qstest}{CatcodeTableStack}{CatcodeTableStack}
790 \def\TestStack#1{%
791   \Expect*{\the\CatcodeTableStack}{#1}%
792 }%
793 \TestStack{0}%
794 \IncCatcodeTableStack
795 \TestStack{2}%
796 \IncCatcodeTableStack
797 \TestStack{4}%
798 \begingroup
799   \IncCatcodeTableStack
800   \TestStack{6}%
801 \endgroup
802 \TestStack{6}%
803 \begingroup
804   \DecCatcodeTableStack
805   \TestStack{4}%
806 \endgroup
807 \TestStack{4}%
808 \DecCatcodeTableStack
809 \TestStack{2}%
810 \DecCatcodeTableStack
811 \TestStack{0}%
812 \begingroup
813   \def\TestError{Missing error}%
814   \def\@PackageError#1#2#3{%
815     \def\TestError{Empty stack}%
816   }%
817   \DecCatcodeTableStack
818   \TestStack{0}%
819   \Expect*{\TestError}{Empty stack}%
820 \endgroup
821 \end{qstest}
```

3.2.4 Catcode regime macros

```
822 \begin{qstest}{CatcodeRegime}{CatcodeRegime}
823  \def\TestStacks#1#2#3{%
824    \Expect*{\the\luatexcatcodetable}{#1}%
825    \Expect*{\the\CatcodeTableStack}{#2}%
826    \Expect*{\LuT@NumStack}{#3}%
827  }%
828  \TestStacks{0}{0}{0}%
829  \catcode`\|=7 %
830  \BeginCatcodeRegime\CatcodeTableLaTeX
831    \TestStacks{2}{2}{{0}0}%
832    \Expect*{\the\catcode`\|}{12}%
833  \EndCatcodeRegime
834  \TestStacks{0}{0}{0}%
835  \Expect*{\the\catcode`\|}{7}%
836 \end{qstest}
```

3.3 Attribute allocation

```
837 \begin{qstest}{Attributes}{Attributes}
838  \newattribute\TestAttr
839  \Expect*{\meaning\TestAttr}%
840      *{\string\attribute\number\allocationnumber}%
841  \Expect*{\the\allocationnumber}{0}%
842  \begingroup
843    \newattribute\TestAttr
844    \Expect*{\the\allocationnumber}{1}%
845  \endgroup
846  \Expect*{\the\allocationnumber}{0}%
847  \Expect*{\meaning\TestAttr}*{\string\attribute1}%
848  \Expect*{\the\TestAttr}*{\number\LuT@UnsetAttributeValue}%
849  \def\Test#1{%
850    \setattribute\TestAttr{#1}%
851    \Expect*{\the\TestAttr}{#1}%
852  }%
853  \Test{0}%
854  \Test{1}%
855  \Test{-1}%
856  \Test{123}%
857  \unsetattribute\TestAttr
858  \Expect*{\the\TestAttr}*{\number\LuT@UnsetAttributeValue}%
859  \begingroup
860    \Expect*{\the\TestAttr}*{\number\LuT@UnsetAttributeValue}%
861    \Test{1234}%
862  \endgroup
863  \Expect*{\the\TestAttr}*{\number\LuT@UnsetAttributeValue}%
864 \end{qstest}

865 \@@end
866 ⟨/test4⟩
```

3.4 Short test for plain TEX

```
867 ⟨*test5⟩
868 \input luatex.sty\relax
869 \newattribute\TestAttr
870 \setattribute\TestAttr{10}
871 \unsetattribute\TestAttr
872 \newcatcodetable\TestCTa
873 \begingroup
874  \SetCatcodeRange{`A}{`Z}{12}%
875 \endgroup
876 \BeginCatcodeRegime\CatcodeTableLaTeX
877 \EndCatcodeRegime
```

```
878 \end
879 ⟨/test5⟩
```

4 Installation

4.1 Download

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

CTAN:macros/latex/contrib/oberdiek/luatex.dtx The source file.

CTAN:macros/latex/contrib/oberdiek/luatex.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.

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 luatex.dtx
```

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

```
luatex.sty            → tex/generic/oberdiek/luatex.sty
luatex-loader.sty     → tex/generic/oberdiek/luatex-loader.sty
oberdiek.luatex.lua   → scripts/oberdiek/oberdiek.luatex.lua
luatex.pdf            → doc/latex/oberdiek/luatex.pdf
test/luatex-test1.tex → doc/latex/oberdiek/test/luatex-test1.tex
test/luatex-test2.tex → doc/latex/oberdiek/test/luatex-test2.tex
test/luatex-test3.tex → doc/latex/oberdiek/test/luatex-test3.tex
test/luatex-test4.tex → doc/latex/oberdiek/test/luatex-test4.tex
test/luatex-test5.tex → doc/latex/oberdiek/test/luatex-test5.tex
luatex.dtx            → source/latex/oberdiek/luatex.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`.

---

[1] http://ctan.org/pkg/luatex

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

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{luatex.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 luatex.dtx
makeindex -s gind.ist luatex.idx
pdflatex luatex.dtx
makeindex -s gind.ist luatex.idx
pdflatex luatex.dtx
```

5 Catalogue

The following XML file can be used as source for the TeX Catalogue. The elements `caption` and `description` are imported from the original XML file from the Catalogue. The name of the XML file in the Catalogue is `luatex.xml`.

```
880 ⟨*catalogue⟩
881 <?xml version='1.0' encoding='us-ascii'?>
882 <!DOCTYPE entry SYSTEM 'catalogue.dtd'>
883 <entry datestamp='$Date$' modifier='$Author$' id='luatex'>
884   <name>luatex</name>
885   <caption>The LuaTeX engine.</caption>
886   <authorref id='auth:oberdiek'/>
887   <copyright owner='Heiko Oberdiek' year='2007,2009,2010'/>
888   <license type='lppl1.3'/>
889   <version number='0.6'/>
890   <description>
891     LuaTeX is an extended version of pdfTeX using Lua as an embedded
892     scripting language.  The LuaTeX project&#x2019;s main objective
893     is to provide an open and configurable variant of TeX while at the
894     same time offering downward compatibility.
895     <p/>
896     LuaTeX uses Unicode (as UTF-8) as its default input encoding, and
897     is able to use modern (OpenType) fonts (for both text and mathematics).
898     <p/>
899     It should be noted that LuaTeX is still under development; its
900     specification has been declared stable, but absolute stability
901     may not in practice be assumed.
```

```
902    <p/>
903    The package is part of the <xref refid='oberdiek'>oberdiek</xref> bundle.
904  </description>
905  <documentation details='Package documentation'
906      href='ctan:macros/latex/contrib/oberdiek/luatex.pdf'/>
907  <ctan file='true' path='/macros/latex/contrib/oberdiek/luatex.dtx'/>
908  <miktex location='oberdiek'/>
909  <texlive location='oberdiek'/>
910  <install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip'/>
911 </entry>
912 ⟨/catalogue⟩
```

6 History

[2007/12/12 v0.1]

- First public version.

[2009/04/10 v0.2]

- Requires package ifluatex in version 2.0 to ensure \luatexversion.

- Updates the call of \directlua, the syntax has changed in LuaTeX 0.36.

[2009/12/02 v0.3]

- Unsetting of attributes updated for LuaTeX 0.37.

[2010/03/09 v0.4]

- Support for lua states removed.

- Calling tex.enableprimitives for used primitives.

[2016/05/10 v0.5]

- Use package.searchers

- only conditionally define commands such as \newcatcodetable that are now defined in the LuaLaTeX format.

[2016/05/16 v0.6]

- 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.

26

27