

The luatexbase-cctb package

Manuel Pégourié-Gonnard
mpg@elzevir.fr

Élie Roux
elie.roux@telecom-bretagne.eu

v0.2a 2010-05-27

Abstract

In addition to the registers existing in $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ and $\epsilon\text{-}\mathrm{T}_{\mathrm{E}}\mathrm{X}$, $\mathrm{LuaT}_{\mathrm{E}}\mathrm{X}$ introduces a new concept: catcode tables. This package takes care of catcode table allocation just like Plain $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ and $\mathrm{LaT}_{\mathrm{E}}\mathrm{X}$ do for other registers.

Contents

1	Documentation	1
2	Implementation	2
2.1	$\mathrm{T}_{\mathrm{E}}\mathrm{X}$ package	2
2.1.1	Preliminaries	2
2.1.2	Load supporting Lua module	4
2.1.3	Primitives needed	4
2.1.4	User macros	4
2.1.5	Predefined tables	5
2.2	Lua module	7
3	Test files	7

1 Documentation

The main macro defined here is `\newluatexcatcodetable`. It behaves the same as `\newbox`. Additionally, the newly allocated catcode table is initialised to the catcodes of $\mathrm{IniT}_{\mathrm{E}}\mathrm{X}$. In order to help you define the catcode tables (once they are allocated), two helper macros are available.

`\setcatcoderange{⟨from⟩}{⟨to⟩}{⟨value⟩}`

Set all characters code in the range $\langle from \rangle - \langle to \rangle$ to the given catcode $\langle value \rangle$.

`\setluatexcatcodetable{⟨table⟩}{⟨catcode statements⟩}`

Set a previously allocated $\langle table \rangle$ to the catcodes given by executing $\langle catcode\ statements \rangle$. Note that $\langle table \rangle$ must not be the current active catcode table. You may, however, load another catcode table in your $\langle catcode\ statements \rangle$.

For your convenience, a few catcode tables are predefined:

- `\CatcodeTableIniTeX`: Ini \TeX catcodes.
- `\CatcodeTableString`: the catcode regime used by `\string` and `\meaning`: everything has catcode 12, except space (U+0020) that has catcode 10.
- `\CatcodeTableOther`: everything (included space) has catcode 12.
- `\CatcodeTableLaTeX`: basic \LaTeX 2_ε catcodes.
- `\CatcodeTableLaTeXAtLetter`: same as above, but `@` is a letter.
- `\CatcodeTableExpl`: catcodes used by \LaTeX 3 with `\ExplSyntaxOn`. Be aware that this does not provide the exact same environment as `\ExplSyntaxOn`: most noticeably, some booleans are not set, and `\endlinechar` is not adjusted (it should be 32).

Various Lua function accept a catcode table number as argument. In order to use them, the package writer needs to know the number of an allocated catcode table. Since `\chardef` is used for the definition of the control sequence, this is rather easy to do. However, for extra ease of use, the numbers are also directly accessible from Lua as the value of the table `luatexbase.catcodetables`, whose keys is the name of the control sequence (without any leading backslash). Moreover, nickames are available for the predefined catcode tables:

- `CatcodeTableIniTeX = ini`,
- `CatcodeTableString = string`,
- `CatcodeTableOther = other`,
- `CatcodeTableLaTeX = latex`,
- `CatcodeTableLaTeXAtLetter = latex-atletter = latex-package`,
- `CatcodeTableExpl = expl = expl3`,

2 Implementation

2.1 \TeX package

1 $\langle *texpackage \rangle$

2.1.1 Preliminaries

Reload protection, especially for Plain \TeX .

```
2 \csname lltxb@cctb@loaded\endcsname
3 \expandafter\let\csname lltxb@cctb@loaded\endcsname\endinput
```

Catcode defenses.

```

4 \begingroup
5   \catcode123 1 % {
6   \catcode125 2 % }
7   \catcode 35 6 % #
8   \toks0{}%
9   \def\x{}%
10  \def\y#1 #2 {%
11    \toks0\expandafter{\the\toks0 \catcode#1 \the\catcode#1}%
12    \edef\x{\x \catcode#1 #2}}%
13  \y 123 1 % {
14  \y 125 2 % }
15  \y 35 6 % #
16  \y 10 12 % ^^J
17  \y 34 12 % "
18  \y 36 3 % $ $
19  \y 39 12 % '
20  \y 40 12 % (
21  \y 41 12 % )
22  \y 42 12 % *
23  \y 43 12 % +
24  \y 44 12 % ,
25  \y 45 12 % -
26  \y 46 12 % .
27  \y 47 12 % /
28  \y 60 12 % <
29  \y 61 12 % =
30  \y 64 11 % @ (letter)
31  \y 62 12 % >
32  \y 95 12 % _ (other)
33  \y 96 12 % `
34  \edef\y#1{\endgroup\edef#1{\the\toks0\relax}\x}%
35 \expandafter\y\csname lltxb@cctb@AtEnd\endcsname

```

Package declaration.

```

36 \begingroup
37   \expandafter\ifx\csname ProvidesPackage\endcsname\relax
38     \def\x#1[#2]{\immediate\write16{Package: #1 #2}}
39   \else
40     \let\x\ProvidesPackage
41   \fi
42 \expandafter\endgroup
43 \x{luatexbase-cctb}[2010/05/27 v0.2a Catcodetable allocation for LuaTeX]

```

Make sure LuaTeX is used.

```

44 \begingroup\expandafter\expandafter\expandafter\endgroup
45 \expandafter\ifx\csname RequirePackage\endcsname\relax
46   \input ifluatex.sty
47 \else
48   \RequirePackage{ifluatex}

```

```

49 \fi
50 \ifluatex\else
51   \begingroup
52     \expandafter\ifx\csname PackageWarningNoLine\endcsname\relax
53       \def\x#1#2{\begingroup\newlinechar10
54         \immediate\write16{Package #1 warning: #2}\endgroup}
55     \else
56       \let\x\PackageWarningNoLine
57   \fi
58 \expandafter\endgroup
59 \x{luatexbase-cctb}{LuaTeX is required for this package. Aborting.}
60 \lltxb@cctb@AtEnd
61 \expandafter\endinput
62 \fi

```

2.1.2 Load supporting Lua module

First load luatexbase-loader (hence luatexbase-compat), then the supporting Lua module.

```

63 \begingroup\expandafter\expandafter\expandafter\endgroup
64 \expandafter\ifx\csname RequirePackage\endcsname\relax
65   \input luatexbase-loader.sty
66 \else
67   \RequirePackage{luatexbase-loader}
68 \fi
69 \luatexbase@directlua{require('luatexbase.cctb')}

```

2.1.3 Primitives needed

Load luatexbase-compat.

```

70 \begingroup\expandafter\expandafter\expandafter\endgroup
71 \expandafter\ifx\csname RequirePackage\endcsname\relax
72   \input luatexbase-compat.sty
73 \else
74   \RequirePackage{luatexbase-compat}
75 \fi

```

Make sure the primitives we need are available.

```

76 \luatexbase@ensure@primitive{luaescapestring}
77 \luatexbase@ensure@primitive{catcodetable}
78 \luatexbase@ensure@primitive{initcatcodetable}
79 \luatexbase@ensure@primitive{savecatcodetable}

```

2.1.4 User macros

The allocation macro. Allocate tables starting with 1, since table 0 is reserved for IniTeX catcodes by LuaTeX.

```

80 \newcount\lltxb@catcodetable@alloc
81 \lltxb@catcodetable@alloc\z@
82 \def\newluatexcatcodetable#1{%

```

```

83 \ifnum\llt\b@catcodetable@alloc<65535\relax
84 \global\advance\llt\b@catcodetable@alloc\@ne
85 \allocationnumber\llt\b@catcodetable@alloc
86 \global\chardef#1\allocationnumber
87 \luatexinitcatcodetable\allocationnumber
88 \begingroup\escapechar\m@ne
89 \luatexbase@directlua{luatexbase.catcodetabledef_from_tex(
90   '\luatexluaescapestring{\string#1}', '\number\allocationnumber')}%
91 \endgroup
92 \wlog{\string#1=\string\luatexcatcodetable\the\allocationnumber}%
93 \else
94 \errmessage{No room for a new \string\luatexcatcodetable}%
95 \fi}

```

A small patch to get two new counters in Plain too.

```

96 \expandafter\ifx\csname @tempcnta\endcsname\relax
97 \csname newcount\endcsname\@tempcnta
98 \fi
99 \expandafter\ifx\csname @tempcntb\endcsname\relax
100 \csname newcount\endcsname\@tempcntb
101 \fi

```

Set the catcodes for a range of characters.

```

102 \def\setcatcoderange#1#2#3{%
103 \edef\luaSCR@temp{%
104 \noexpand\@tempcnta=\the\@tempcnta
105 \noexpand\@tempcntb=\the\@tempcntb
106 \noexpand\count@=\the\count@
107 \relax}%
108 \@tempcnta=#1\relax
109 \@tempcntb=#2\relax
110 \count@=#3\relax
111 \loop\unless\ifnum\@tempcnta>\@tempcntb
112 \catcode\@tempcnta=\count@
113 \advance\@tempcnta\@ne
114 \repeat
115 \luaSCR@temp}

```

Set a catcode table.

```

116 \def\setluatexcatcodetable#1#2{%
117 \begingroup
118 #2%
119 \luatexsavecatcodetable#1%
120 \endgroup}

```

2.1.5 Predefined tables

The IniTeX catcode table needs no extra initialisation.

```

121 \newluatexcatcodetable\CatcodeTableIniTeX

```

The String and Other catcode tables.

```
122 \newluatexcatcodetable\CatcodeTableString
123 \setluatexcatcodetable\CatcodeTableString{%
124   \luatexcatcodetable\CatcodeTableIniTeX
125   \catcode0 12 % nul
126   \catcode13 12 % carriage return
127   \catcode37 12 % percent
128   \setcatcoderange{65}{90}{12}% A-Z
129   \setcatcoderange{97}{122}{12}% a-z
130   \catcode92 12 % backslash
131   \catcode127 12 }

132 \newluatexcatcodetable\CatcodeTableOther
133 \setluatexcatcodetable\CatcodeTableOther{%
134   \luatexcatcodetable\CatcodeTableString
135   \catcode32 12 }
```

Tables for L^AT_EX 2_ε.

```
136 \newluatexcatcodetable\CatcodeTableLaTeX
137 \setluatexcatcodetable\CatcodeTableLaTeX{%
138   \luatexcatcodetable\CatcodeTableIniTeX
139   \setcatcoderange{0}{31}{15}%
140   \catcode9 10 % tab
141   \catcode12 13 % form feed
142   \catcode13 5 % carriage return
143   \catcode35 6 % hash
144   \catcode36 3 % dollar
145   \catcode38 4 % ampersand
146   \catcode94 7 % circumflex
147   \catcode95 8 % underscore
148   \catcode123 1 % brace left
149   \catcode125 2 % brace right
150   \catcode126 13 % tilde
151   \catcode127 15 }

152 \newluatexcatcodetable\CatcodeTableLaTeXAtLetter
153 \setluatexcatcodetable\CatcodeTableLaTeXAtLetter{%
154   \luatexcatcodetable\CatcodeTableLaTeX
155   \catcode64=11 }
```

A table for expl3.

```
156 \newluatexcatcodetable\CatcodeTableExpl
157 \setluatexcatcodetable\CatcodeTableExpl{%
158   \luatexcatcodetable\CatcodeTableLaTeX
159   \catcode126 10 % tilde is a space char
160   \catcode32 9 % space is ignored
161   \catcode9 9 % tab also ignored
162   \catcode95 11 % underscore letter
163   \catcode58 11 % colon letter
164 }
```

Finally do the shortcuts.

```
165 \luatexbase@directlua{luatexbase.catcodetable_do_shortcuts()}
```

That's all, folks!

```
166 \lltxb@cctb@AtEnd
```

```
167 \</texpackage>
```

2.2 Lua module

```
168 <lua module>
```

```
169 module('luatexbase', package.seeall)
```

The number associated to a CS name is remembered in the `catcodetables` table.

```
170 catcodetables = {}
```

```
171 function catcodetabledef_from_tex(name, number)
```

```
172     catcodetables[name] = tonumber(number)
```

```
173 end
```

The next function creates some shortcuts for better readability in lua code. This makes `luatexbase.catcodetables.latex` equivalent to `luatexbase.catcodetables.CatcodeTableLaTeX`.

```
174 function catcodetable_do_shortcuts()
```

```
175     local cat = catcodetables
```

```
176     cat['latex'] = cat.CatcodeTableLaTeX
```

```
177     cat['latex-package'] = cat.CatcodeTableLaTeXAtLetter
```

```
178     cat['latex-atletter'] = cat.CatcodeTableLaTeXAtLetter
```

```
179     cat['ini'] = cat.CatcodeTableIniTeX
```

```
180     cat['expl3'] = cat.CatcodeTableExpl
```

```
181     cat['expl'] = cat.CatcodeTableExpl
```

```
182     cat['string'] = cat.CatcodeTableString
```

```
183     cat['other'] = cat.CatcodeTableOther
```

```
184 end
```

```
185 </lua module>
```

3 Test files

The tests done are very basic: we just make sure that the package loads correctly and the macros don't generate any error, under both LaTeX en Plain TeX.

```
186 <testplain>\input luatexbase-cctb.sty
```

```
187 <testlatex>\RequirePackage{luatexbase-cctb}
```

```
188 <*testplain,testlatex>
```

```
189 \begingroup \catcode64 11 \global\let\lua\luatexbase@directlua \endgroup
```

Also check that the catcodetable's number is remembered well, independently of the current value of `\escapechar`.

```
190 \newluatexcatcodetable\testcctb
```

```
191 \lua{assert(luatexbase.catcodetables.testcctb)}
```

```
192 \begingroup
```

```
193 \escapechar64
```

```
194 \newluatexcatcodetable\anothercctb
```

```
195 \endgroup
```

```
196 \lua{assert(luatexbase.catcodetables.anothercctb)}
```

Now, play a little bit with predefined tables.

```
197 \luatexcatcodetable\CatcodeTableLaTeXAtLetter
198 \ifnum\catcode64=11 \else \ERROR \fi
199 \luatexcatcodetable\CatcodeTableLaTeX
200 \ifnum\catcode64=12 \else \ERROR \fi
201 <testlatex>\documentclass{minimal}
202 \lua{%
203   tex.sprint('\string\setbox0=\string\hbox{')
204   tex.sprint(luatexbase.catcodetables.string, "\string\undef # _^&")
205   tex.sprint('}')
206 }
207 </testplain,testlatex>
208 <testplain>\bye
209 <testlatex>\stop
```