

The `luatexbase-reg`s package

Heiko Oberdiek (primary author of `luatex`)
Élie Roux, Manuel Pégourié-Gonnard, Philipp Gesang*
<https://github.com/lualatex/luatexbase>
lualatex-dev@tug.org

2011/05/24 v0.4

Abstract

This package extends the register allocation scheme of Plain TeX and L^AT_EX to take advantage of the increased number of registers available in LuaTeX.

Contents

1 Documentation	1
2 Implementation	2
2.1 Preliminaries	2
2.2 Ensure etex.sty is loaded	3
2.3 Adapt range	4
2.4 Patch macros that used \mathchardef	4
2.5 Make room for inserts	6
3 Test files	6

1 Documentation

Since the Plain TeX and L^AT_EX formats are both frozen, they fail to take into account the extended resources provided by newer TeX-like engines. This package focuses on the allocation scheme for registers. TeX 82 provides 6 kinds of registers: count, dimen, skip, muskip, box and toks and has 256 registers of each kind. ε-T_EX and most of its descendants add one kind of register (marks) and offers $2^{15} = 32768$ of each kind. LuaTeX provides $2^{16} = 65536$ registers of each kind. (It also provides new register-like resources, but this package addresses only the resources inherited from ε-T_EX.)

More precisely, `luatexbase-reg`s loads the `etex` package (or makes sure it is preloaded in the format) and then adapts it to the new limits of LuaTeX. Thus, all macros defined by the `etex` package are made available (most notably, `\loccount`, `\globcountblk`, `\loccountblk` and alike). However, if a register of some kind has been locally allocated before this package is loaded,

*See “History” in `luatexbase.pdf` for details.

then the number of allocatable registers of this kind will not be extended to 65536. To avoid this, load `luatexbase-reg`s earlier.

The Plain TeX and L^AT_EX formats define a new kind of resource: *inserts* which are merely a family (count, dimen, skip, box) of registers with the same number. Inserts allocation begins at 255 and goes toward 0. Thus we can make room for more inserts by making allocation of count-, dimen-, skip- and box-registers start from 256. With real ε-TeX, it may be a bad idea since registers with index greater than 256 have degraded performance due to implementation details, but with LuaTeX the performance is uniform, so we just do it.

2 Implementation

1 ⟨*texpackage⟩

2.1 Preliminaries

Catcode defenses and reload protection.

```

2 \begingroup\catcode61\catcode48\catcode32=10\relax% = and space
3  \catcode123 1 %
4  \catcode125 2 %
5  \catcode 35 6 % #
6  \toks0\expandafter{\expandafter\endlinechar\the\endlinechar}%
7  \edef\x{\endlinechar13}%
8  \def\y#1 #2 {%
9    \toks0\expandafter{\the\toks0 \catcode#1 \the\catcode#1}%
10   \edef\x{\x \catcode#1 #2}%
11 \y 13 5 % carriage return
12 \y 61 12 % =
13 \y 32 10 % space
14 \y 123 1 % {
15 \y 125 2 % }
16 \y 35 6 % #
17 \y 64 11 % @ (letter)
18 \y 10 12 % new line ^J
19 \y 33 12 % !
20 \y 36 3 % $ $ (in etex.sty)
21 \y 39 12 % '
22 \y 40 12 % (
23 \y 41 12 % )
24 \y 42 12 % * (in etex.sty)
25 \y 43 12 % + (in etex.sty)
26 \y 44 12 % , (in etex.sty)
27 \y 45 12 % -
28 \y 46 12 % .
29 \y 47 12 % /
30 \y 58 12 % :
31 \y 60 12 % < (in etex.sty)
32 \y 62 12 % > (in etex.sty)
33 \y 91 12 % [
34 \y 93 12 % ]
35 \y 94 7 % ^
36 \y 96 12 % '
37 \toks0\expandafter{\the\toks0 \relax\noexpand\endinput}%

```

```

38 \edef\y{\noexpand\expandafter\endgroup%
39   \noexpand\ifx#1\relax \edef#1{\the\toks0}\x\relax%
40   \noexpand\else \noexpand\expandafter\noexpand\endinput%
41   \noexpand\fi}%
42 \expandafter\y\csname luatexbase@regs@sty@endinput\endcsname%

```

Package declaration.

```

43 \begingroup
44 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
45 \def\x#1[#2]{\immediate\write16{Package: #1 #2}}
46 \else
47 \let\x\ProvidesPackage
48 \fi
49 \expandafter\endgroup
50 \x{luatexbase-reg} [2011/05/24 v0.4 Registers allocation for LuaTeX]

```

Make sure *LuaTeX* is used.

```

51 \begingroup\expandafter\expandafter\expandafter\endgroup
52 \expandafter\ifx\csname RequirePackage\endcsname\relax
53 \input ifluatex.sty
54 \else
55 \RequirePackage{ifluatex}
56 \fi
57 \ifluatex\else
58 \begingroup
59 \expandafter\ifx\csname PackageError\endcsname\relax
60 \def\x#1#2#3{\begingroup \newlinechar10
61   \errhelp{#3}\errmessage{Package #1 error: #2}\endgroup}
62 \else
63 \let\x\PackageError
64 \fi
65 \expandafter\endgroup
66 \x{luatexbase-reg}{LuaTeX is required for this package. Aborting.}%
67 This package can only be used with the LuaTeX engine^J%
68 (command 'lualatex' or 'luatex').^J%
69 Package loading has been stopped to prevent additional errors.%
70 \expandafter\luatexbase@regs@sty@endinput%
71 \fi

```

2.2 Ensure *etex.sty* is loaded

If running *L^AT_EX*, load *etex.sty*. If not, either *etex.src* was loaded at format generation time, or we cannot do anything.

```

72 \begingroup\expandafter\expandafter\expandafter\endgroup
73 \expandafter\ifx\csname RequirePackage\endcsname\relax \else
74 \RequirePackage{etex}[1998/03/26]
75 \fi

```

To the best of my (mpg) knowledge, all Plain-based formats built with ε -*T_EX*-enabled engines in *T_EX Live* load *etex.src*. However, let's be careful and check that *etex.sty* or *etex.src* is loaded.

```

76 \begingroup\expandafter\expandafter\expandafter\endgroup
77 \expandafter\ifx\csname et@xins\endcsname\relax

```

```

78 \begingroup
79   \expandafter\ifx\csname PackageWarningNoLine\endcsname\relax
80     \def\x#1#2{\begingroup\newlinechar10
81       \immediate\write16{Package #1 warning: #2}\endgroup}
82   \else
83     \let\x\PackageWarningNoLine
84   \fi
85 \expandafter\endgroup
86 \x{luatexbase-reg}{etex macros not loaded!^^J%
87   Registers allocation scheme will not be extended.}
88 \else

```

2.3 Adapt range

First, increase the upper bound for all kinds of registers. Copy code to avoid defining a macro.

```

89 \ifnum\count270=32768 \count270=65536 \fi
90 \ifnum\count271=32768 \count271=65536 \fi
91 \ifnum\count272=32768 \count272=65536 \fi
92 \ifnum\count273=32768 \count272=65536 \fi
93 \ifnum\count273=32768 \count273=65536 \fi
94 \ifnum\count274=32768 \count274=65536 \fi
95 \ifnum\count275=32768 \count275=65536 \fi
96 \ifnum\count276=32768 \count276=65536 \fi

```

2.4 Patch macros that used \mathchardef

\box registers and \marks were previously defined using \mathchardef since it had the biggest range under ε -TEX (15-bit number). However, this is not enough for LUATEX's extended registers. Fortunately, \chardef's range is extended, and now large enough, so use it everywhere instead of \mathchardef. Do this inside a group and use \toks0 to store the list of actions.

```

97 \begingroup \toks0={}
98   \def\@namedef #1{\expandafter \def\csname#1\endcsname}
99   \def\@outerdef#1{\expandafter\outer\expandafter\def\csname#1\endcsname}

```

Notice that the auxiliary macros will automatically expand to the desired level when necessary, see below.

First, here are the definitions from etex.src, in a form adapted to our needs.

```

100 \def\def@globbox #1#2{@outerdef{#1}{\et@xglob 4 \box #2}}
101 \def\def@locbox #1#2{@namedef {#1}{\et@xloc 4 \box #2}}
102 \def\def@globmarks #1#2{@outerdef{#1}{\et@xglob 6 \marks #2}}
103 \def\def@locmarks #1#2{@namedef {#1}{\et@xloc 6 \marks #2}}
104 \def\def@et@xgblk#1#2{@namedef{#1}##1##2##3##4%
105   {\et@xchkblk ##1##2##4%}
106   {\allocationnumber=\count 26##1
107    \global \advance \count 26##1 by ##4%
108    \global #2##3=\allocationnumber
109    \wlog {\string ##3=\string ##2blk{\number ##4}
110      at \the \allocationnumber}%
111    }%
112  }%
113 \def\def@et@xlblblk#1#2{@namedef{#1}##1##2##3##4%
114   {\et@xchkblk ##1##2##4%}

```

```

115      {\advance \count 27##1 by -##4%
116          \allocationnumber=\count 27##1
117          #2##3=\allocationnumber
118          \wlog {\string ##3=\string ##2blk{\number ##4}
119              at \the \allocationnumber \space (local)%
120              }%
121          }%
122      }}

```

Then, the definitions from `etex.sty` since they are subtly different (`\outer` status, but also optional spaces or = signs).

```

123 \def\alt@globbox #1#2{@namedef{#1}{\et@xglob 4\box #2}}
124 \def\alt@locbox #1#2{@namedef{#1}{\et@xloc 4\box #2}}
125 \def\alt@globmarks #1#2{@namedef{#1}{\et@xglob 6\marks #2}}
126 \def\alt@locmarks #1#2{@namedef{#1}{\et@xloc 6\marks #2}}
127 \def\alt@et@xgblk#1#2{@namedef{#1}##1##2##3##4%
128     {\et@xchkblk##1##2##4}%
129     {\allocationnumber\count26##1%
130         \global\advance\count26##1by##4%
131         \global#2##3\allocationnumber
132         \wlog{\string##3=\string##2blk{\number##4} at
133             \the\allocationnumber}%
134         }%
135     }%
136 \def\alt@et@xlblk#1#2{@namedef{#1}##1##2##3##4%
137     {\et@xchkblk##1##2##4}%
138     {\advance\count27##1-##4%
139         \allocationnumber\count27##1%
140         #2##3\allocationnumber
141         \et@xwlog{\string##3=\string##2blk{\number##4} at
142             \the\allocationnumber\space(local)}%
143         }%
144     }%

```

Now, a macro checking the definitions, and making the appropriate re-definition.

```

145 \def\check@def#1{%
146     \csname def@#1\endcsname{test@#1}\mathchardef
147     \expandafter\ifx\csname test@#1\expandafter\endcsname
148         \csname #1\endcsname
149         \expandafter\let\csname #1\endcsname\relax
150         \toks0\expandafter{\the\toks0\csname def@#1\endcsname{#1}\chardef}
151     \else
152         \csname alt@#1\endcsname{test@#1}\mathchardef
153         \expandafter\ifx\csname test@#1\expandafter\endcsname
154             \csname #1\endcsname
155             \toks0\expandafter{\the\toks0\csname alt@#1\endcsname{#1}\chardef}
156         \else
157             \expandafter\show\csname BAD#1\endcsname
158         \fi
159     \fi}

```

Now, actually do it.

```

160     \check@def{globbox}
161     \check@def{locbox}

```

```

162  \check@def{globmarks}
163  \check@def{locmarks}
164  \check@def{et@xgblk}
165  \check@def{et@xlblk}
166  \expandafter \endgroup
167  \the\toks0

```

2.5 Make room for inserts

Finally, make allocation of `\count`, `\dimen`, `\skip` and `\box` start with numbers > 255, in order to free the lower numbers for insertions. Be careful with `\new...` macros which are `\outer` in Plain, since we're in the middle of an `\if` test.

```

168 \expandafter\let\csname newcount\endcsname\globcount
169 \expandafter\let\csname newdimen\endcsname\globdimen
170 \expandafter\let\csname newskip\endcsname\globskip
171 \expandafter\let\csname newbox\endcsname\globbox
172 \fi

```

That's all folks!

```

173 \luatexbase@regs@sty@endinput%
174 /texpackage>

```

3 Test files

Here we test only the two main formats: Plain T_EX (with `etex.src` loaded) and L^AT_EX, both with the LuaT_EX engine. Those correspond to the `luatex` and `lualatex` commands in T_EX Live.

We want to make sure we can globally and locally allocate 30000 registers of each kind, and still globally allocate 100 `\inserts`. Next we globally allocate a bloc of 3000 registers of each kind, and locally a block of 1000. (Those numbers are not optimal, but they should be enough for testing purposes.)

```

175 <testplain>\input luatexbase-regss.sty
176 <testlalex>\RequirePackage{luatexbase-regss}
177 {*testplain,testlalex}
178 \def\checkregister#1{%
179   \edef\newregister{\expandafter\noexpand\csname new#1\endcsname}%
180   \edef\locregister{\expandafter\noexpand\csname loc#1\endcsname}%
181   \count0 1
182   \loop
183     \newregister\dummy
184     \locregister\dummy
185   \ifnum\count0<30000
186     \advance\count0 1
187   \repeat}
188 \checkregister{count}
189 \checkregister{dimen}
190 \checkregister{skip}
191 \checkregister{muskip}
192 \checkregister{box}
193 \checkregister{toks}
194 \checkregister{marks}
195

```

```
196 \count0 1
197 \loop \ifnum\count0<100
198   \csname newinsert\endcsname\dummy
199   \advance\count0 1
200 \repeat
201
202 \globcountblk \dummy{3000}
203 \globdimenblk \dummy{3000}
204 \globskipblk \dummy{3000}
205 \globmuskipblk\dummy{3000}
206 \globboxblk \dummy{3000}
207 \globtoksblk \dummy{3000}
208 \globmarksblk \dummy{3000}
209
210 \loccountblk \dummy{1000}
211 \locdimenblk \dummy{1000}
212 \locskipblk \dummy{1000}
213 \locmuskipblk \dummy{1000}
214 \locboxblk \dummy{1000}
215 \loctoksblk \dummy{1000}
216 \locmarksblk \dummy{1000}
217 </testplain,testlatex>
218 <testplain>\bye
219 <testlatex>\stop
```