

The `hyphen.cfg` file for Lua_T_EX

Khaled Hosny, Élie Roux, and Manuel Pégourié-Gonnard

`khaledhosny@eglug.org`

`elie.roux@telecom-bretagne.eu`

`mpg@elzevir.fr`

2010/04/28 v1.3beta

Abstract

This is a modified version of the file `hyphen.cfg` distributed with the `babel` package, with a supporting Lua module, aimed at adapting `babel`'s hyphenation patterns loading mechanism to Lua_T_EX's dynamic pattern loading capabilities.

1 Documentation

Hyphenation patterns should be loaded at runtime with Lua_T_EX: if they appear in the format, they will be rehashed when the format is loaded anyway, which makes the format quite long to load (many seconds even on modern machines) and provides for bad user experience. Hence, it is desirable to load as few patterns as possible in the format, and load on-demand the needed patterns at runtime.

For backward compatibility, Knuth's original patterns for US English are still loaded in the format, as `\language0`.

This package provides a modified version of `hyphen.cfg` adapted to Lua_T_EX, as well as a supporting Lua module. Since a lot of things, especially the catcodes, are not as predictable at runtime than at format creation time, we don't `\input` the usual pattern files, but rather load the patterns using the Lua interface, using a special plain text version of the pattern files kindly provided by the `texhyphen` project.

The modifications applied in this files are highlighted in the code documentation below, but here is a summary:

- not loading patterns in the format except for english
- loading patterns at runtime, except for english
- modified banner

This file checks for the engine, and should continue to work with any engine without any modified behaviour. However, it is recommended to install it in such

a way that the original `hyphen.cfg` from `babel` is found first by any engine other than `LuaTeX`.

2 Package code

2.1 `luatex-hyphen.lua`

```

1
2 luatexhyphen = {}
3
4 luatexhyphen.version = "1.3beta"
5
6 local dbname = "language.dat.lua"
7
8 local function warn (msg, ...)
9     texio.write_nl('luatex-hyphen: '..string.format(msg, ...))
10 end
11
12 luatexhyphen.language_dat = {}
13 local dbfile = kpse.find_file(dbname)
14 if not dbfile then
15     warn("file not found: "..dbname)
16 else
17     luatexhyphen.language_dat = dofile(dbfile)
18 end
19
20 local function lookupname(l)
21     if luatexhyphen.language_dat[l] then
22         return luatexhyphen.language_dat[l], 1
23     else
24         for orig,lt in pairs(luatexhyphen.language_dat) do
25             for _,syn in ipairs(lt.synonyms) do
26                 if syn == l then
27                     return lt, orig
28                 end
29             end
30         end
31     end
32     return nil
33 end
34
35 local language_loaded = { [0] = true }
36
37 function luatexhyphen.loadlanguage(l, id)
38     if language_loaded[id] then return end
39     language_loaded[id] = true
40     local lt, orig = lookupname(l)
41     if not lt or not lt.code then
42         warn("no entry in %s for this language: %s", dbname, l)

```

```

43     return
44 end
45 warn("loading patterns and exceptions for: %s (\\language%s)", orig, id)
46 for _, ext in ipairs({'pat', 'hyp'}) do
47     local n = 'hyph-'..lt.code..'..'..ext..'..txt'
48     local f = kpse.find_file(n)
49     if not f then
50         warn("file not found: %s", n)
51         return
52     end
53     f = io.open(f, 'r')
54     local data = f:read('*a')
55     f:close()
56     if not data then
57         warn("file not readable: %s", f)
58         return
59     end
60     local lobj = lang.new(id)
61     lang.patterns(lobj, data)
62 end
63 end

```

2.2 hyphen.cfg

```

64 \ifx\ProvidesFile\@undefined
65 \def\ProvidesFile#1[#2 #3 #4]{%
66 \wlog{File: #1 #4 #3 <#2>}%

    Use a modified banner for LuaTeX.
67 \ifx\directlua\@undefined
68 \toks8{Babel <#3> and hyphenation patterns for }%
69 \else
70 \toks8{LuaTeX adaptation of babel <#3>
71 and hyphenation patterns for }%
72 \fi

73 \let\ProvidesFile\@undefined
74 }
75 \def\ProvidesLanguage#1[#2 #3 #4]{%
76 \wlog{Language: #1 #4 #3 <#2>}%
77 }
78 \else
79 \let\bbl@tempa\ProvidesFile
80 \def\ProvidesFile#1[#2 #3 #4]{%

    Same here.
81 \ifx\directlua\@undefined
82 \toks8{Babel <#3> and hyphenation patterns for }%
83 \else
84 \toks8{LuaTeX adaptation of babel <#3>
85 and hyphenation patterns for }%
86 \fi

```

```

87 \bbl@tempa#1[#2 #3 #4]%
88 \let\ProvidesFile\bbl@tempa}
89 \def\ProvidesLanguage#1{%
90 \begingroup
91 \catcode'\ 10 %
92 \@makeother\/%
93 \@ifnextchar[%]
94 {\@provideslanguage{#1}}{\@provideslanguage{#1}[]}}
95 \def\@provideslanguage#1[#2]{%
96 \wlog{Language: #1 #2}%
97 \expandafter\xdef\csname ver@#1.1df\endcsname{#2}%
98 \endgroup}
99 \fi
100

```

File identification is modified again.

```

101 \ProvidesFile{hyphen.cfg}
102 [2010/04/26 v3.81-luatex-1.3beta %
103 Language switching mechanism for LuaTeX, adapted from babel v3.81]
104 \ifx\AtBeginDocument\@undefined
105 \input plain.def\relax
106 \fi
107 \ifx\language\@undefined
108 \csname newcount\endcsname\language
109 \fi
110 \ifx\newlanguage\@undefined
111 \csname newcount\endcsname\last@language
112 \else
113 \countdef\last@language=19
114 \fi
115 \ifx\newlanguage\@undefined
116 \def\addlanguage#1{%
117 \global\advance\last@language \@ne
118 \ifnum\last@language<\@cclvi
119 \else
120 \errmessage{No room for a new \string\language!}%
121 \fi
122 \global\chardef#1\last@language
123 \wlog{\string#1 = \string\language\the\last@language}}
124 \else
125 \def\addlanguage{\alloc@9\language\chardef\@cclvi}
126 \fi
127 \def\adddialect#1#2{%
128 \global\chardef#1#2\relax
129 \wlog{\string#1 = a dialect from \string\language#2}}
130 \def\iflanguage#1{%
131 \expandafter\ifx\csname l@#1\endcsname\relax
132 \@nolanerr{#1}%
133 \else
134 \bbl@afterfi{\ifnum\csname l@#1\endcsname=\language

```

```

135     \expandafter\@firstoftwo
136     \else
137     \expandafter\@secondoftwo
138     \fi}%
139 \fi}
140 \edef\selectlanguage{%
141     \noexpand\protect
142     \expandafter\noexpand\csname selectlanguage \endcsname
143 }
144 \ifx\@undefined\protect\let\protect\relax\fi
145 \ifx\documentclass\@undefined
146     \def\xstring{\string\string\string}
147 \else
148     \let\xstring\string
149 \fi
150 \xdef\bbl@language@stack{}
151 \def\bbl@push@language{%
152     \xdef\bbl@language@stack{\language+\bbl@language@stack}%
153 }
154 \def\bbl@pop@lang#1+#2-#3{%
155     \def\language{#1}\xdef#3{#2}%
156 }
157 \def\bbl@pop@language{%
158     \expandafter\bbl@pop@lang\bbl@language@stack-\bbl@language@stack
159     \expandafter\bbl@set@language\expandafter{\language}%
160 }
161 \expandafter\def\csname selectlanguage \endcsname#1{%
162     \bbl@push@language
163     \aftergroup\bbl@pop@language
164     \bbl@set@language{#1}}
165 \def\bbl@set@language#1{%
166     \edef\language{%
167         \ifnum\escapechar=\expandafter'\string#1\@empty
168         \else \string#1\@empty\fi}%
169     \select@language{\language}%
170     \if@filesw
171         \protected@write\@auxout{}\{\string\select@language{\language}\}%
172         \addtocontents{toc}\{\xstring\select@language{\language}\}%
173         \addtocontents{lof}\{\xstring\select@language{\language}\}%
174         \addtocontents{lot}\{\xstring\select@language{\language}\}%
175     \fi}
176 \def\select@language#1{%
177     \expandafter\ifx\csname l@#1\endcsname\relax
178         \@nolanerr{#1}%
179     \else
180         \expandafter\ifx\csname date#1\endcsname\relax
181             \@noopterr{#1}%
182         \else
183             \bbl@patterns{\language}%
184             \originalTeX

```

```

185     \expandafter\def\expandafter\originalTeX
186         \expandafter{\csname noextras#1\endcsname
187             \let\originalTeX\@empty}%
188     \languageshorthands{none}%
189     \babel@beginsave
190     \csname captions#1\endcsname
191     \csname date#1\endcsname
192     \csname extras#1\endcsname\relax
193     \babel@savevariable\lefthyphenmin
194     \babel@savevariable\righthyphenmin
195     \expandafter\ifx\csname #1hyphenmins\endcsname\relax
196     \set@hyphenmins\tw@\thr@@\relax
197     \else
198     \expandafter\expandafter\expandafter\set@hyphenmins
199     \csname #1hyphenmins\endcsname\relax
200     \fi
201     \fi
202     \fi}
203 \long\def\otherlanguage#1{%
204     \csname selectlanguage \endcsname{#1}%
205     \ignorespaces
206     }
207 \long\def\endotherlanguage{%
208     \originalTeX
209     \global\@ignoretrue\ignorespaces
210     }
211 \expandafter\def\csname otherlanguage*\endcsname#1{%
212     \foreign@language{#1}%
213     }
214 \expandafter\def\csname endotherlanguage*\endcsname{%
215     \csname noextras\language\endcsname
216     }
217 \def\foreignlanguage{\protect\csname foreignlanguage \endcsname}
218 \expandafter\def\csname foreignlanguage \endcsname#1#2{%
219     \begingroup
220         \originalTeX
221         \foreign@language{#1}%
222         #2%
223         \csname noextras#1\endcsname
224     \endgroup
225     }
226 \def\foreign@language#1{%
227     \def\language#1}%
228     \expandafter\ifx\csname l@#1\endcsname\relax
229     \@nolanerr{#1}%
230     \else
231     \bbl@patterns{\language}%
232     \languageshorthands{none}%
233     \csname extras#1\endcsname
234     \expandafter\ifx\csname #1hyphenmins\endcsname\relax

```

```

235     \set@hyphenmins\tw@\thr@@\relax
236   \else
237     \expandafter\expandafter\expandafter\set@hyphenmins
238       \csname #1hyphenmins\endcsname\relax
239   \fi
240 \fi
241 }
242 \def\bbl@patterns#1{%

```

With Lua_T_EX, load patterns and exceptions at runtime using functions from the supporting Lua module, which remembers which languages have already been loaded.

Also, since this code will be executed at runtime, be careful while testing if we're using Lua_T_EX.

```

243 \ifx\directlua@\undefined\else
244   \ifx\directlua\relax\else
245     \directlua{
246       if not luatexhyphen then
247         dofile(kpse.find_file("luatex-hyphen.lua"))
248       end
249       luatexhyphen.loadlanguage("\luatexluaescapestring{#1}",
250         \number\csname l@#1\endcsname)
251     }%
252   \fi
253 \fi
254 \language=\expandafter\ifx\csname l@#1:\f@encoding\endcsname\relax
255   \csname l@#1\endcsname
256 \else
257   \csname l@#1:\f@encoding\endcsname
258 \fi\relax
259 }
260 \def\hyphenrules#1{%
261   \expandafter\ifx\csname l@#1\endcsname@\undefined
262     \@nolanerr{#1}%
263   \else
264     \bbl@patterns{#1}%
265     \languageshorthands{none}%
266     \expandafter\ifx\csname #1hyphenmins\endcsname\relax
267       \set@hyphenmins\tw@\thr@@\relax
268     \else
269       \expandafter\expandafter\expandafter\set@hyphenmins
270       \csname #1hyphenmins\endcsname\relax
271     \fi
272   \fi
273 }
274 \def\endhyphenrules{}
275 \def\providehyphenmins#1#2{%
276   \expandafter\ifx\csname #1hyphenmins\endcsname\relax
277     \@namedef{#1hyphenmins}{#2}%
278   \fi}

```

```

279 \def\set@hyphenmins#1#2{\lefthyphenmin#1\rightthyphenmin#2}
280 \def\LdfInit{%
281   \chardef\atcatcode=\catcode'\@
282   \catcode'\@=11\relax
283   \input babel.def\relax
284   \catcode'\@=\atcatcode \let\atcatcode\relax
285   \LdfInit}
286 \ifx\originalTeX\@undefined\let\originalTeX\@empty\fi
287 \ifx\babel@beginsave\@undefined\let\babel@beginsave\relax\fi
288 \ifx\PackageError\@undefined
289   \def\@nolanerr#1{%
290     \errhelp{Your command will be ignored, type <return> to proceed}%
291     \errmessage{You haven't defined the language #1\space yet}}
292   \def\@nopatterns#1{%
293     \message{No hyphenation patterns were loaded for}%
294     \message{the language '#1'}%
295     \message{I will use the patterns loaded for \string\language=0
296       instead}}
297   \def\@noopterr#1{%
298     \errmessage{The option #1 was not specified in \string\usepackage}
299     \errhelp{You may continue, but expect unexpected results}}
300   \def\@activated#1{%
301     \wlog{Package babel Info: Making #1 an active character}}
302 \else
303   \newcommand*\@nolanerr}[1]{%
304     \PackageError{babel}%
305       {You haven't defined the language #1\space yet}%
306       {Your command will be ignored, type <return> to proceed}}
307   \newcommand*\@nopatterns}[1]{%
308     \PackageWarningNoLine{babel}%
309       {No hyphenation patterns were loaded for\MessageBreak
310         the language '#1'\MessageBreak
311         I will use the patterns loaded for \string\language=0
312         instead}}
313   \newcommand*\@noopterr}[1]{%
314     \PackageError{babel}%
315       {You haven't loaded the option #1\space yet}%
316       {You may proceed, but expect unexpected results}}
317   \newcommand*\@activated}[1]{%
318     \PackageInfo{babel}{%
319       Making #1 an active character}}
320 \fi
321 \def\process@line#1#2 #3/{%
322   \ifx=#1
323     \process@synonym#2 /
324   \else
325     \process@language#1#2 #3/%
326   \fi
327 }
328 \toks@{}

```



```

329 \def\process@synonym#1 /{%
330   \ifnum\last@language=\m@ne
331     \expandafter\chardef\csname l@#1\endcsname0\relax
332     \wlog{\string\l@#1=\string\language0}
333     \toks@ \expandafter{\the\toks@
334       \expandafter\let\csname #1hyphenmins\expandafter\endcsname
335       \csname\language\hyphenmins\endcsname}%
336   \else
337     \expandafter\chardef\csname l@#1\endcsname\last@language
338     \wlog{\string\l@#1=\string\language\the\last@language}
339     \expandafter\let\csname #1hyphenmins\expandafter\endcsname
340     \csname\language\hyphenmins\endcsname
341   \fi
342 }
343 \def\process@language#1 #2 #3/{%
344   \expandafter\addlanguage\csname l@#1\endcsname
345   \expandafter\language\csname l@#1\endcsname
346   \def\language{#1}%

```

Yet another banner modification. See below why the test makes sense.

```

347 \ifx\directlua\@undefined
348   \global\toks8\expandafter{\the\toks8#1, }%
349 \else
350   \ifnum\last@language=\z@
351     \global\toks8\expandafter{\the\toks8#1 }%
352   \fi
353 \fi
354 \begingroup
355   \bbl@get@enc#1:\@@@
356   \ifx\bbl@hyph@enc\@empty
357   \else
358     \fontencoding{\bbl@hyph@enc}\selectfont
359   \fi
360   \lefthyphenmin\m@ne

```

Assume the first (that is, zeroth) language in `language.dat` is English. This assumption is very reasonable, since otherwise it would break compatibility with frozen `TeXby` not providing Knuth's original patterns as `\language0`, so we're pretty sure about this point.

We do load this first language, since we want Knuth's patterns to be active as soon as the format is loaded. But once it is done, we don't want to load any other language.

```

361 \ifx\directlua\@undefined
362   \input #2\relax
363 \else
364   \ifnum\last@language=\z@
365     \gdef\bbl@luatex@english@loaded{1}%
366     \input #2\relax
367   \fi
368 \fi

```

```

369 \ifnum\lefthyphenmin=\m@ne
370 \else
371 \expandafter\xdef\csname #1hyphenmins\endcsname{%
372 \the\lefthyphenmin\the\righthyphenmin}%
373 \fi
374 \endgroup
375 \ifnum\the\language=\z@
376 \expandafter\ifx\csname #1hyphenmins\endcsname\relax
377 \set@hyphenmins\tw@\thr@@\relax
378 \else
379 \expandafter\expandafter\expandafter\set@hyphenmins
380 \csname #1hyphenmins\endcsname
381 \fi
382 \the\toks@
383 \fi
384 \toks@{}%
385 \def\bbl@tempa{#3}%
386 \ifx\bbl@tempa\@empty
387 \else
388 \ifx\bbl@tempa\space
389 \else

```

Likewise, don't load hyphenation exceptions now, but rather when we load the patterns. (Anyway, in practice, the third field of `language.dat` is never used since exceptions are defined in the same file as patterns, so it doesn't really matter.)

There are no hyphenation exceptions for english, and since it is frozen, we can rely on this, so no need for a special case for english here.

```

390 \ifx\directlua\@undefined
391 \input #3\relax
392 \fi
393 \fi
394 \fi
395 }
396 \def\bbl@get@enc#1:#2\@@@{%
397 \def\bbl@tempa{#1}%
398 \def\bbl@tempb{#2}%
399 \ifx\bbl@tempb\@empty
400 \let\bbl@hyph@enc\@empty
401 \else
402 \bbl@get@enc#2\@@@
403 \edef\bbl@hyph@enc{\bbl@tempa}%
404 \fi}
405 \openin1 = language.dat
406 \ifeof1
407 \message{I couldn't find the file language.dat,\space
408 I will try the file hyphen.tex}
409 \input hyphen.tex\relax
410 \else
411 \last@language\m@ne
412 \loop

```

```

413 \endlinechar\m@ne
414 \read1 to \bbl@line
415 \endlinechar'\^^M
416 \ifx\bbl@line\@empty
417 \else
418 \edef\bbl@line{\bbl@line\space/}%
419 \expandafter\process@line\bbl@line
420 \fi
421 \iftrue \csname fi\endcsname
422 \csname if\ifeof1 false\else true\fi\endcsname
423 \repeat
424 \language=0
425 \fi
426 \closein1
427 \let\process@language\@undefined
428 \let\process@synonym\@undefined
429 \let\process@line\@undefined
430 \let\bbl@tempa\@undefined
431 \let\bbl@tempb\@undefined
432 \let\bbl@eq\@undefined
433 \let\bbl@line\@undefined
434 \let\bbl@get@enc\@undefined
435 \ifx\addto@hook\@undefined
436 \else
437 \expandafter\addto@hook\expandafter\everyjob\expandafter{%
438 \expandafter\typeout\expandafter{\the\toks8 loaded.}}
439 \fi

```