

The luamplib package

Hans Hagen, Taco Hoekwater, Elie Roux, Philipp Gesang and Kim Dohyun
Maintainer: LuaLaTeX Maintainers — Support: <lualatex-dev@tug.org>

2014/01/20 v2.3

Abstract

Package to have metapost code typeset directly in a document with LuaTeX.

1 Documentation

This package aims at providing a simple way to typeset directly metapost code in a document with LuaTeX. LuaTeX is built with the lua mplib library, that runs metapost code. This package is basically a wrapper (in Lua) for the Lua mplib functions and some TeX functions to have the output of the mplib functions in the pdf.

The package needs to be in PDF mode in order to output something, as PDF specials are not supported by the DVI format and tools.

The metapost figures are put in a TeX hbox with dimensions adjusted to the metapost code.

Using this package is easy: in Plain, type your metapost code between the macros `\mplibcode` and `\endmplibcode`, and in \LaTeX in the `mplibcode` environment.

The code is from the `luatex-mplib.lua` and `luatex-mplib.tex` files from ConTeXt, they have been adapted to \LaTeX and Plain by Elie Roux and Philipp Gesang, new functionalities have been added by Kim Dohyun. The changes are:

- a \LaTeX environment
- all TeX macros start by `mplib`
- use of `luatexbase` for errors, warnings and declaration
- possibility to use `btex ... etex` to typeset TeX code. `textext()` is a more versatile macro equivalent to `TEX()` from `TEX.mp`. `TEX()` is also allowed unless `TEX.mp` is loaded, which should be always avoided.
- `verbatimtex ... etex` that comes just before `beginfig()` is not ignored, but the TeX code inbetween will be inserted before the following `mplib` hbox. Using this command, each `mplib` box can be freely moved horizontally and/or vertically. All other `verbatimtex ... etex`'s are ignored. *E.G.*

`\mplibcode`

```

verbatimtex \moveright 3cm etex; beignfig(0); ... endfig;
verbatimtex \leavevmode etex; beignfig(1); ... endfig;
verbatimtex \leavevmode\lower 1ex etex; beignfig(2); ... endfig;
verbatimtex \endgraf\moveright 1cm etex; beignfig(3); ... endfig;
\endmplibcode

```

N.B. `\endgraf` should be used instead of `\par` inside `verbatimtex ... etex`.

- Notice that, after each figure is processed, macro `\MPwidth` stores the width value of latest figure; `\MPheight`, the height value.
- Since v2.3, new macros `\everymplib` and `\everyendmplib` redefine token lists `\everymplibtoks` and `\everyendmplibtoks` respectively, which will be automatically inserted at the beginning and ending of each `mplib` code. *E.G.*

```

\everymplib{ verbatimtex \leavevmode etex; beignfig(0); }
\everyendmplib{ endfig; }
\mplibcode % beignfig/endfig not needed; always in horizontal mode
  draw fullcircle scaled 1cm;
\endmplibcode

```

- Since v2.3, `\mpdim` and other raw \TeX commands are allowed inside `mplib` code. This feature is inspired by `gmp.sty` authored by Enrico Gregorio. Please refer the manual of `gmp` package for details. *E.G.*

```

\begin{mplibcode}
  draw origin--(\mpdim{\linewidth},0) withpen pencircle scaled 4
  dashed evenly scaled 4 withcolor \myrulecolor;
\end{mplibcode}

```

N.B. Users should not use the protected variant of `btex ... etex` as provided by `gmp` package. As `luamplib` automatically protects \TeX code inbetween, `\btex` is not supported here.

There are (basically) two formats for `metapost`: *plain* and *metafun*. By default, the *plain* format is used, but you can set the format to be used by future figures at any time using `\mplibsetformat{<format name>}`.

2 Implementation

2.1 Lua module

Use the `luamplib` namespace, since `mplib` is for the `metapost` library itself. `ConTeXt` uses `metapost`.

```

1
2 luamplib          = luamplib or { }
3

```

Identification.

```
4
5 local luamplib      = luamplib
6 luamplib.showlog    = luamplib.showlog or false
7 luamplib.lastlog    = ""
8
9 local err, warn, info, log = luatexbase.provides_module({
10   name      = "luamplib",
11   version   = 2.3,
12   date      = "2014/01/20",
13   description = "Lua package to typeset Metapost with LuaTeX's MPLib.",
14 })
15
16
```

This module is a stripped down version of libraries that are used by ConT_EXt. Provide a few “shortcuts” expected by the imported code.

```
17
18 local format, abs = string.format, math.abs
19
20 local stringgsub    = string.gsub
21 local stringfind    = string.find
22 local stringmatch   = string.match
23 local stringgmatch  = string.gmatch
24 local tableconcat   = table.concat
25 local teksprint     = tex.sprint
26
27 local mplib = require ('mplib')
28 local kpse  = require ('kpse')
29
30 local file = file
31 if not file then
32
```

This is a small trick for L^AT_EX. In L^AT_EX we read the metapost code line by line, but it needs to be passed entirely to process(), so we simply add the lines in data and at the end we call process(data).

A few helpers, taken from l-file.lua.

```
33
34 file = { }
35
36 function file.replacesuffix(filename, suffix)
37   return (stringgsub(filename,"%.[%a%d]+$","")) .. "." .. suffix
38 end
39
40 function file.stripsuffix(filename)
41   return (stringgsub(filename,"%.[%a%d]+$",""))
42 end
43 end
```

As the finder function for mplib, use the kpse library and make it behave like as if MetaPost was used (or almost, since the engine name is not set this way—not sure if this is a problem).

```

44
45 local mpkpse = kpse.new("luatex", "mpost")
46
47 local function finder(name, mode, ftype)
48     if mode == "w" then
49         return name
50     else
51         return mpkpse.find_file(name, ftype)
52     end
53 end
54 luamplib.finder = finder
55

```

The rest of this module is not documented. More info can be found in the LuaTeX manual, articles in user group journals and the files that ship with ConTeXt.

```

56
57 function luamplib.resetlastlog()
58     luamplib.lastlog = ""
59 end
60

```

Below included is section that defines fallbacks for older versions of mplib.

```

61 local mplibone = tonumber(mplib.version()) <= 1.50
62
63 if mplibone then
64
65     luamplib.make = luamplib.make or function(name, mem_name, dump)
66         local t = os.clock()
67         local mpx = mplib.new {
68             ini_version = true,
69             find_file = luamplib.finder,
70             job_name = file.stripsuffix(name)
71         }
72         mpx:execute(format("input %s ;", name))
73         if dump then
74             mpx:execute("dump ;")
75             info("format %s made and dumped for %s in %0.3f seconds", mem_name, name, os.clock()-t)
76         else
77             info("%s read in %0.3f seconds", name, os.clock()-t)
78         end
79         return mpx
80     end
81
82     function luamplib.load(name)
83         local mem_name = file.replacesuffix(name, "mem")
84         local mpx = mplib.new {

```

```

85         ini_version = false,
86         mem_name = mem_name,
87         find_file = luamplib.finder
88     }
89     if not mpx and type(luamplib.make) == "function" then
90         -- when i have time i'll locate the format and dump
91         mpx = luamplib.make(name, mem_name)
92     end
93     if mpx then
94         info("using format %s", mem_name, false)
95         return mpx, nil
96     else
97         return nil, { status = 99, error = "out of memory or invalid format" }
98     end
99 end
100
101 else
102

```

These are the versions called with sufficiently recent mplib.

```

103
104     local preamble = [[
105         boolean mplib ; mplib := true ;
106         let dump = endinput ;
107         let normalfontsize = fontsize;
108         input %s ;
109     ]]
110
111     luamplib.make = luamplib.make or function()
112     end
113
114     function luamplib.load(name)
115         local mpx = mplib.new {
116             ini_version = true,
117             find_file = luamplib.finder,
118         }
119         local result
120         if not mpx then
121             result = { status = 99, error = "out of memory"}
122         else
123             result = mpx:execute(format(preamble, file.replacesuffix(name, "mp")))
124         end
125         luamplib.reporterror(result)
126         return mpx, result
127     end
128
129 end
130
131 local currentformat = "plain"
132

```

```

133 local function setformat (name) --- used in .sty
134     currentformat = name
135 end
136 luamplib.setformat = setformat
137
138
139 luamplib.reporterror = function (result)
140     if not result then
141         err("no result object returned")
142     elseif result.status > 0 then
143         local t, e, l = result.term, result.error, result.log
144         if t then
145             info(t)
146         end
147         if e then
148             err(e)
149         end
150         if not t and not e and l then
151             luamplib.lastlog = luamplib.lastlog .. "\n " .. l
152             log(l)
153         else
154             err("unknown, no error, terminal or log messages")
155         end
156     else
157         return false
158     end
159     return true
160 end
161
162 local function process_indeed (mpx, data)
163     local converted, result = false, {}
164     local mpx = luamplib.load(mpx)
165     if mpx and data then
166         local result = mpx:execute(data)
167         if not result then
168             err("no result object returned")
169         elseif result.status > 0 then
170             err("%s", (result.term or "no-term") .. "\n" .. (result.error or "no-error"))
171         elseif luamplib.showlog then
172             luamplib.lastlog = luamplib.lastlog .. "\n" .. result.term
173             info("%s", result.term or "no-term")
174         elseif result.fig then
175             converted = luamplib.convert(result)
176         else
177             err("unknown error, maybe no beginfig/endfig")
178         end
179     else
180         err("Mem file unloadable. Maybe generated with a different version of mplib?")
181     end
182     return converted, result

```

```

183 end
184 local process = function (data)
185     return process_indeed(currentformat, data)
186 end
187 luamplib.process = process
188
189 local function getobjects(result, figure, f)
190     return figure:objects()
191 end
192
193 local function convert(result, flusher)
194     luamplib.flush(result, flusher)
195     return true -- done
196 end
197 luamplib.convert = convert
198
199 local function pdf_startfigure(n, llx, lly, urx, ury)

```

The following line has been slightly modified by Kim.

```

200     texsprint(format("\mplibstarttoPDF{%f}{%f}{%f}{%f}", llx, lly, urx, ury))
201 end
202
203 local function pdf_stopfigure()
204     texsprint("\mplibstoptoPDF")
205 end
206
207 local function pdf_literalcode(fmt, ...) -- table
208     texsprint(format("\mplibtoPDF{%s}", format(fmt, ...)))
209 end
210 luamplib.pdf_literalcode = pdf_literalcode
211
212 local function pdf_textfigure(font, size, text, width, height, depth)

```

The following three lines have been modified by Kim.

```

213     -- if text == "" then text = "\0" end -- char(0) has gone
214     text = text:gsub(".", function(c)
215         return format("\hbox{\char%i}", string.byte(c)) -- kerning happens in meta-
           post
216     end)
217     texsprint(format("\mplibtexttext{%s}{%f}{%s}{%s}{%f}", font, size, text, 0, -( 7200/ 7227)/65536*depth))
218 end
219 luamplib.pdf_textfigure = pdf_textfigure
220
221 local bend_tolerance = 131/65536
222
223 local rx, sx, sy, ry, tx, ty, divider = 1, 0, 0, 1, 0, 0, 1
224
225 local function pen_characteristics(object)
226     local t = mplib.pen_info(object)
227     rx, ry, sx, sy, tx, ty = t.rx, t.ry, t.sx, t.sy, t.tx, t.ty

```

```

228     divider = sx*sy - rx*ry
229     return not (sx==1 and rx==0 and ry==0 and sy==1 and tx==0 and ty==0), t.width
230 end
231
232 local function concat(px, py) -- no tx, ty here
233     return (sy*px-ry*py)/divider, (sx*py-rx*px)/divider
234 end
235
236 local function curved(ith,pth)
237     local d = pth.left_x - ith.right_x
238     if abs(ith.right_x - ith.x_coord - d) <= bend_tolerance and abs(pth.x_coord - pth.left_x - d) <= bend_tolerance then
239         d = pth.left_y - ith.right_y
240         if abs(ith.right_y - ith.y_coord - d) <= bend_tolerance and abs(pth.y_coord - pth.left_y - d) <= bend_tolerance then
241             return false
242         end
243     end
244     return true
245 end
246
247 local function flushnormalpath(path,open)
248     local pth, ith
249     for i=1,#path do
250         pth = path[i]
251         if not ith then
252             pdf_literalcode("%f %f m",pth.x_coord,pth.y_coord)
253         elseif curved(ith,pth) then
254             pdf_literalcode("%f %f %f %f c",ith.right_x,ith.right_y,pth.left_x,pth.left_y,pth.x_coord,pth.y_coord)
255         else
256             pdf_literalcode("%f %f l",pth.x_coord,pth.y_coord)
257         end
258         ith = pth
259     end
260     if not open then
261         local one = path[1]
262         if curved(pth,one) then
263             pdf_literalcode("%f %f %f %f %f %f c",pth.right_x,pth.right_y,one.left_x,one.left_y,one.x_coord,one.y_coord)
264         else
265             pdf_literalcode("%f %f l",one.x_coord,one.y_coord)
266         end
267     elseif #path == 1 then
268         -- special case .. draw point
269         local one = path[1]
270         pdf_literalcode("%f %f l",one.x_coord,one.y_coord)
271     end
272     return t
273 end
274
275 local function flushconcatpath(path,open)

```



```

276 pdf_literalcode("%f %f %f %f %f cm", sx, rx, ry, sy, tx ,ty)
277 local pth, ith
278 for i=1,#path do
279     pth = path[i]
280     if not ith then
281         pdf_literalcode("%f %f m", concat(pth.x_coord, pth.y_coord))
282     elseif curved(ith, pth) then
283         local a, b = concat(ith.right_x, ith.right_y)
284         local c, d = concat(pth.left_x, pth.left_y)
285         pdf_literalcode("%f %f %f %f %f c", a, b, c, d, concat(pth.x_coord, pth.y_co-
ord))
286     else
287         pdf_literalcode("%f %f l", concat(pth.x_coord, pth.y_coord))
288     end
289     ith = pth
290 end
291 if not open then
292     local one = path[1]
293     if curved(pth, one) then
294         local a, b = concat(pth.right_x, pth.right_y)
295         local c, d = concat(one.left_x, one.left_y)
296         pdf_literalcode("%f %f %f %f %f c", a, b, c, d, concat(one.x_coord, one.y_co-
ord))
297     else
298         pdf_literalcode("%f %f l", concat(one.x_coord, one.y_coord))
299     end
300 elseif #path == 1 then
301     -- special case .. draw point
302     local one = path[1]
303     pdf_literalcode("%f %f l", concat(one.x_coord, one.y_coord))
304 end
305 return t
306 end
307

```

Below code has been contributed by Dohyun Kim. It implements btex / etex functions.

v2.1: `texttext()` is now available, which is equivalent to `TEX()` macro from `TEX.mp`.

`TEX()` is synonym of `texttext()` unless `TEX.mp` is loaded.

v2.2: Transparency and Shading

v2.2: `\everymplib`, `\everyendmplib`, and allows naked \TeX commands.

```

308 local further_split_keys = {
309     ["MPLibTEXboxID"] = true,
310     ["sh_color_a"]     = true,
311     ["sh_color_b"]     = true,
312 }
313
314 local function script2table(s)
315     local t = {}
316     for i in stringmatch(s, "[^\13]+") do
317         local k, v = stringmatch(i, "(.-)=(.)") -- v may contain =.

```

```

318     if k and v then
319         local vv = {}
320         if further_split_keys[k] then
321             for j in stringmatch(v,"^:+") do
322                 vv[#vv+1] = j
323             end
324         end
325         if #vv > 0 then
326             t[k] = vv
327         else
328             t[k] = v
329         end
330     end
331 end
332 return t
333 end
334
335 local mplibcodepreamble = [[
336 vardef rawtexttext (expr t) =
337     if unknown TEXBOX_:
338         image( special "MPlibmkTEXbox=" & t; )
339     else:
340         TEXBOX_ := TEXBOX_ + 1;
341         image (
342             addto currentpicture doublepath unitsquare
343             xscaled TEXBOX_wd[TEXBOX_]
344             yscaled (TEXBOX_ht[TEXBOX_] + TEXBOX_dp[TEXBOX_])
345             shifted (0, -TEXBOX_dp[TEXBOX_])
346             withprescript "MPlibTEXboxID=" &
347                 decimal TEXBOX_ & ":" &
348                 decimal TEXBOX_wd[TEXBOX_] & ":" &
349                 decimal(TEXBOX_ht[TEXBOX_]+TEXBOX_dp[TEXBOX_]);
350         )
351     fi
352 enddef;
353 if known context_mlib:
354     defaultfont := "cmitt10";
355     let infont = normalinfont;
356     let fontsize = normalfontsize;
357     vardef thelabel@#(expr p,z) =
358         if string p :
359             thelabel@#(p infont defaultfont scaled defaultscale,z)
360         else :
361             p shifted (z + labeloffset*mfun_laboff@# -
362                 (mfun_labxf@#*lrcorner p + mfun_labyf@#*ulcorner p +
363                 (1-mfun_labxf@#-mfun_labyf@#)*llcorner p))
364         fi
365     enddef;
366 else:
367     vardef texttext@# (text t) = rawtexttext (t) enddef;

```

```

368 fi
369 def externalfigure primary filename =
370   draw rawtexttext("\includegraphics{"& filename &}")
371 enddef;
372 def TEX = texttext enddef;
373 def fontmapfile primary filename = enddef;
374 def specialVerbatimTeX (text t) = special "MPlibVerbTeX="&t; enddef;
375 def ignoreVerbatimTeX (text t) = enddef;
376 let VerbatimTeX = specialVerbatimTeX;
377 extra_beginfig := extra_beginfig & " let VerbatimTeX = ignoreVerbatimTeX;" ;
378 extra_endfig   := extra_endfig   & " let VerbatimTeX = specialVerbatimTeX;" ;
379 ]]
380
381 local function protecttexttext(data)
382   local everymplib = tex.toks['everymplibtoks'] or ''
383   local everyendmplib = tex.toks['everyendmplibtoks'] or ''
384   data = " " .. everymplib .. data .. everyendmplib
385   data = stringgsub(data,
386     "%f[A-Za-z]btex%f[^A-Za-z]%s*(.)%s*f[A-Za-z]etex%f[^A-Za-z]",
387     function(str)
388       str = stringgsub(str, "'", "&ditto&'")
389       return format("rawtexttext(\unexpanded{\\"%s\})", str)
390     end)
391   data = stringgsub(data,
392     "%f[A-Za-z]verbatimtex%f[^A-Za-z]%s*(.)%s*f[A-Za-z]etex%f[^A-Za-z]",
393     function(str)
394       str = stringgsub(str, "'", "&ditto&'")
395       return format("VerbatimTeX(\unexpanded{\\"%s\})", str)
396     end)
397   data = stringgsub(data, "\".-\\\"", -- hack for parentheses inside quotes
398     function(str)
399       str = stringgsub(str, "%(", "%%%%LEFTPAREN%%%"")
400       str = stringgsub(str, "%)", "%%%%RGHTPAREN%%%"")
401       return str
402     end)
403   data = stringgsub(data, "%f[A-Za-z]TEX%s*b()", "\\unexpanded{%1}")
404   data = stringgsub(data, "%f[A-Za-z]texttext%s*b()", "\\unexpanded{%1}")
405   data = stringgsub(data, "%f[A-Za-z]texttext%[_a-z]+%s*b()", "\\unexpanded{%1}")
406   data = stringgsub(data, "%%%%LEFTPAREN%%%", "(") -- restore
407   data = stringgsub(data, "%%%%RGHTPAREN%%%", ")") -- restore
408   texsprint(data)
409 end
410
411 luamplib.protecttexttext = protecttexttext
412
413 local factor = 65536*(7227/7200)
414
415 local function putTEXboxes (object, prescript)
416   local box = prescript.MPlibTEXboxID
417   local n, tw, th = box[1], box[2], box[3]

```

```

418     if n and tw and th then
419         local op = object.path
420         local first, second, fourth = op[1], op[2], op[4]
421         local tx, ty = first.x_coord, first.y_coord
422         local sx, sy = (second.x_coord - tx)/tw, (fourth.y_coord - ty)/th
423         local rx, ry = (second.y_coord - ty)/tw, (fourth.x_coord - tx)/th
424         if sx == 0 then sx = 0.00001 end
425         if sy == 0 then sy = 0.00001 end
426         pdf_literalcode("q %f %f %f %f %f %f cm",sx,rx,ry,sy,tx,ty)
427         texsprint(format("\mplibputtextbox{%i}",n))
428         pdf_literalcode("Q")
429     end
430 end
431
432 local TeX_code_t = {}
433
434 local function domakeTEXboxes (data)
435     local num = tex.count[14] -- newbox register
436     if data and data.fig then
437         local figures = data.fig
438         for f=1, #figures do
439             TeX_code_t[f] = nil
440             local figure = figures[f]
441             local objects = getobjects(data,figure,f)
442             if objects then
443                 for o=1,#objects do
444                     local object = objects[o]
445                     local prescript = object.prescript
446                     prescript = prescript and script2table(prescript)
447                     local str = prescript and prescript.MPlibmkTEXbox
448                     if str then
449                         num = num + 1
450                         texsprint(format("\setbox%i\hbox{%s}",num,str))
451                     end
452                 end
453             end
454             TeX_code_t[f] = texcode
455         end
456     end
457 end
458
459 end
460 end
461
462 local function makeTEXboxes (data)
463     local mpx = luamplib.load(currentformat)
464     if mpx and data then

```

verbatimtex ... etex before beginfig() is not ignored, but the TeX code inbetween is inserted before the mplib box.

```

465     local result = mpx:execute(mplibcodepreamble .. data)
466     domakeTEXboxes(result)
467 end
468 return data
469 end
470
471 luamplib.makeTEXboxes = makeTEXboxes
472
473 local function processwithTEXboxes (data)
474     local num = tex.count[14] -- the same newbox register
475     local prepreamble = "TEXBOX_ := "..num..";\n"
476     while true do
477         num = num + 1
478         local box = tex.box[num]
479         if not box then break end
480         prepreamble = prepreamble ..
481             "TEXBOX_wd["..num.."] := "..box.width /factor..";\n"..
482             "TEXBOX_ht["..num.."] := "..box.height/factor..";\n"..
483             "TEXBOX_dp["..num.."] := "..box.depth /factor..";\n"
484     end
485     process(prepreamble .. mplibcodepreamble .. data)
486 end
487
488 luamplib.processwithTEXboxes = processwithTEXboxes
489

```

Transparency and Shading

```

490 local pdf_objs = {}
491
492 -- objstr <string> => obj <number>, new <boolean>
493 local function update_pdfobjs (os)
494     local on = pdf_objs[os]
495     if on then
496         return on,false
497     end
498     on = pdf.immediateobj(os)
499     pdf_objs[os] = on
500     return on,true
501 end
502
503 local transparency_modes = { [0] = "Normal",
504     "Normal",      "Multiply",    "Screen",      "Overlay",
505     "SoftLight",   "HardLight",   "ColorDodge",  "ColorBurn",
506     "Darken",      "Lighten",    "Difference",  "Exclusion",
507     "Hue",          "Saturation", "Color",       "Luminosity",
508     "Compatible",
509 }
510
511 local function update_tr_res(res,mode,opaq)
512     local os = format("<</BM /%s/ca %g/CA %g/AIS false>>",mode,opaq,opaq)

```

```

513     local on, new = update_pdfobjs(os)
514     if new then
515         res = res .. format("/MPLibTr%s%g %i 0 R",mode,opaq,on)
516     end
517     return res
518 end
519
520 local function tr_pdf_pageresources(mode,opaq)
521     local res = ""
522     res = update_tr_res(res, "Normal", 1)
523     res = update_tr_res(res, mode, opaq)
524     if res ~= "" then
525         local tpr = tex.pdfpageresources -- respect luaotfload-colors
526         if not stringfind(tpr,"/ExtGState<<.*>>") then
527             tpr = tpr.."/ExtGState<<>>"
528         end
529         tpr = stringgsub(tpr,"/ExtGState<<","%1"..res)
530         tex.set("global","pdfpageresources",tpr)
531     end
532 end
533
534 -- luatexbase.mcb is not yet updated: "finish_pdffile" callback is missing
535
536 local function sh_pdfpageresources(shtype, domain, colorspace, colora, colorb, coordinates)
537     local os, on, new
538     os = format("<</FunctionType 2/Domain [ %s ]/C0 [ %s ]/C1 [ %s ]/N 1>>",
539         domain, colora, colorb)
540     on = update_pdfobjs(os)
541     os = format("<</ShadingType %i/ColorSpace %s/Function %i 0 R/Coords [ %s ]/Ex-
542         tend [ true true ]/AntiAlias true>>",
543         shtype, colorspace, on, coordinates)
544     on, new = update_pdfobjs(os)
545     if not new then
546         return on
547     end
548     local res = format("/MPLibSh%i %i 0 R", on, on)
549     local ppr = pdf.pageresources or ""
550     if not stringfind(ppr,"/Shading<<.*>>") then
551         ppr = ppr.."/Shading<<>>"
552     end
553     pdf.pageresources = stringgsub(ppr,"/Shading<<","%1"..res)
554     return on
555 end
556
557 local function color_normalize(ca,cb)
558     if #cb == 1 then
559         if #ca == 4 then
560             cb[1], cb[2], cb[3], cb[4] = 0, 0, 0, 1-cb[1]
561         else -- #ca = 3
562             cb[1], cb[2], cb[3] = cb[1], cb[1], cb[1]

```

```

562     end
563     elseif #cb == 3 then -- #ca == 4
564         cb[1], cb[2], cb[3], cb[4] = 1-cb[1], 1-cb[2], 1-cb[3], 0
565     end
566 end
567
568 local function do_preobj_color(object,prescript)
569     -- transparency
570     local opaq = prescript and prescript.tr_transparency
571     if opaq then
572         local mode = prescript.tr_alternative or 1
573         mode = transparency_modes[tonumber(mode)]
574         tr_pdf_pageresources(mode,opaq)
575         pdf_literalcode("/MPLibTr%s%g gs",mode,opaq)
576     end
577     -- color
578     local cs = object.color
579     if cs and #cs > 0 then
580         pdf_literalcode(luamplib.colorconverter(cs))
581     end
582     -- shading
583     local sh_type = prescript and prescript.sh_type
584     if sh_type then
585         local domain = prescript.sh_domain
586         local centera = prescript.sh_center_a
587         local centerb = prescript.sh_center_b
588         local colora = prescript.sh_color_a or {0};
589         local colorb = prescript.sh_color_b or {1};
590         if #colora > #colorb then
591             color_normalize(colora,colorb)
592         elseif #colorb > #colora then
593             color_normalize(colorb,colora)
594         end
595         local colorspace
596         if #colorb == 1 then colorspace = "DeviceGray"
597         elseif #colorb == 3 then colorspace = "DeviceRGB"
598         elseif #colorb == 4 then colorspace = "DeviceCMYK"
599         else return opaq
600         end
601         colora = tableconcat(colora, " ")
602         colorb = tableconcat(colorb, " ")
603         local shade_no
604         if sh_type == "linear" then
605             local coordinates = format("%s %s",centera,centerb)
606             shade_no = sh_pdf_pageresources(2,domain,colorspace,colora,colorb,coordinates)
607         elseif sh_type == "circular" then
608             local radiusa = prescript.sh_radius_a
609             local radiusb = prescript.sh_radius_b
610             local coordinates = format("%s %s %s %s",centera,radiusa,centerb,radiusb)
611             shade_no = sh_pdf_pageresources(3,domain,colorspace,colora,colorb,coordinates)

```

```

612     end
613     pdf_literalcode("q /Pattern cs")
614     return opaq,shade_no
615 end
616 return opaq
617 end
618
619 local function do_postobj_color(tr,sh)
620   if sh then
621     pdf_literalcode("W n /MPLibSh%s sh Q",sh)
622   end
623   if tr then
624     pdf_literalcode("/MPLibTrNormal1 gs")
625   end
626 end
627

```

End of btex – etex and Transparency/Shading patch.

```

628
629 local function flush(result,flusher)
630   if result then
631     local figures = result.fig
632     if figures then
633       for f=1, #figures do
634         info("flushing figure %s",f)
635         local figure = figures[f]
636         local objects = getobjects(result,figure,f)
637         local fignum = tonumber(stringmatch(figure:filename(),"([%d]+)$") or figure:
        ure:charcode() or 0)
638         local miterlimit, linecap, linejoin, dashed = -1, -1, -1, false
639         local bbox = figure:boundingbox()
640         local llx, lly, urx, ury = bbox[1], bbox[2], bbox[3], bbox[4] -- faster than un-
        pack
641         if urx < llx then
642           -- invalid
643           pdf_startfigure(fignum,0,0,0,0)
644           pdf_stopfigure()
645         else

```

Insert verbatimtex code before mplib box.

```

646         if TeX_code_t[f] then
647           texsprint(TeX_code_t[f])
648         end
649         pdf_startfigure(fignum,llx,lly,urx,ury)
650         pdf_literalcode("q")
651         if objects then
652           for o=1,#objects do
653             local object      = objects[o]
654             local objecttype  = object.type

```


Change from ConT_EXt code: the following 5 lines are part of the btex...etex patch.
Again, colors are processed at this stage.

```

655             local prescript      = object.prescript
656 prescript = prescript and script2table(prescript) -- pre-
        script is now a table
657             local tr_opaq,shade_no = do_preobj_color(object,prescript)
658 if prescript and prescript.MPlibTEXboxID then
659     putTEXboxes(object,prescript)
660 elseif objecttype == "start_bounds" or objecttype == "stop_bounds" then
661     -- skip
662 elseif objecttype == "start_clip" then
663     pdf_literalcode("q")
664     flushnormalpath(object.path,t,false)
665     pdf_literalcode("W n")
666 elseif objecttype == "stop_clip" then
667     pdf_literalcode("Q")
668     miterlimit, linecap, linejoin, dashed = -1, -1, -1, false
669 elseif objecttype == "special" then
670     -- not supported
671 elseif objecttype == "text" then
672     local ot = object.transform -- 3,4,5,6,1,2
673     pdf_literalcode("q %f %f %f %f %f %f cm",ot[3],ot[4],ot[5],ot[6],ot[1],ot[2])
674     pdf_textfigure(object.font,object.dsize,object.text,object.width,object.height)
675     pdf_literalcode("Q")
676 else

```

Color stuffs are modified and moved to several lines above.

```

677             local ml = object.miterlimit
678 if ml and ml ~= miterlimit then
679     miterlimit = ml
680     pdf_literalcode("%f M",ml)
681 end
682             local lj = object.linejoin
683 if lj and lj ~= linejoin then
684     linejoin = lj
685     pdf_literalcode("%i j",lj)
686 end
687             local lc = object.linecap
688 if lc and lc ~= linecap then
689     linecap = lc
690     pdf_literalcode("%i J",lc)
691 end
692             local dl = object.dash
693 if dl then
694     local d = format("[%s] %i d",tableconcat(dl.dashes or {}, " "),dl.offset)
695     if d ~= dashed then
696         dashed = d
697         pdf_literalcode(dashed)
698     end
699 elseif dashed then

```

```

700         pdf_literalcode("[ ] 0 d")
701         dashed = false
702     end
703     local path = object.path
704     local transformed, penwidth = false, 1
705     local open = path and path[1].left_type and path[#path].right_type
706     local pen = object.pen
707     if pen then
708         if pen.type == 'elliptical' then
709             transformed, penwidth = pen_characteris-
710         pdf_literalcode("%f w", penwidth)
711         if objecttype == 'fill' then
712             objecttype = 'both'
713         end
714         else -- calculated by mplib itself
715             objecttype = 'fill'
716         end
717     end
718     if transformed then
719         pdf_literalcode("q")
720     end
721     if path then
722         if transformed then
723             flushconcatpath(path, open)
724         else
725             flushnormalpath(path, open)
726         end
727     end

```

Change from ConTeXt code: color stuff

```

727         if not shade_no then ----- conflict with shad-
728         ing
729         if objecttype == "fill" then
730             pdf_literalcode("h f")
731         elseif objecttype == "outline" then
732             pdf_literalcode((open and "S") or "h S")
733         elseif objecttype == "both" then
734             pdf_literalcode("h B")
735         end
736     end
737     if transformed then
738         pdf_literalcode("Q")
739     end
740     local path = object.htap
741     if path then
742         if transformed then
743             pdf_literalcode("q")
744         end
745         if transformed then

```

```

746         flushconcatpath(path,open)
747     else
748         flushnormalpath(path,open)
749     end
750     if objecttype == "fill" then
751         pdf_literalcode("h f")
752     elseif objecttype == "outline" then
753         pdf_literalcode((open and "S") or "h S")
754     elseif objecttype == "both" then
755         pdf_literalcode("h B")
756     end
757     if transformed then
758         pdf_literalcode("Q")
759     end
760 end
761 --     if cr then
762 --         pdf_literalcode(cr)
763 --     end
764 end

Added to ConTeXt code: color stuff

765         do_postobj_color(tr_opaq,shade_no)
766     end
767 end
768 pdf_literalcode("Q")
769 pdf_stopfigure()
770 end
771 end
772 end
773 end
774 end
775 luamplib.flush = flush
776
777 local function colorconverter(cr)
778     local n = #cr
779     if n == 4 then
780         local c, m, y, k = cr[1], cr[2], cr[3], cr[4]
781         return format("%.3g %.3g %.3g %.3g k %.3g %.3g %.3g %.3g K",c,m,y,k,c,m,y,k), "0 g 0 G"
782     elseif n == 3 then
783         local r, g, b = cr[1], cr[2], cr[3]
784         return format("%.3g %.3g %.3g rg %.3g %.3g %.3g RG",r,g,b,r,g,b), "0 g 0 G"
785     else
786         local s = cr[1]
787         return format("%.3g g %.3g G",s,s), "0 g 0 G"
788     end
789 end
790 luamplib.colorconverter = colorconverter

```

2.2 T_EX package

```

791 <*package>

    First we need to load some packages.
792 \bgroup\expandafter\expandafter\expandafter\egroup
793 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
794   \input luatexbase-modutils.sty
795 \else
796   \NeedsTeXFormat{LaTeX2e}
797   \ProvidesPackage{luamplib}
798   [2014/01/20 v2.3 mplib package for LuaTeX]
799   \RequirePackage{luatexbase-modutils}
800   \RequirePackage{pdftexcmds}
801 \fi

    Loading of lua code.
802 \RequireLuaModule{luamplib}

    Set the format for metapost.
803 \def\mplibsetformat#1{%
804   \directlua{luamplib.setformat("\luatexluaescapestring{#1}")}}

    MPLib only works in PDF mode, we don't do anything if we are in DVI mode, and
    we output a warning.
805 \ifnum\pdfoutput>0
806   \let\mplibtoPDF\pdfliteral
807 \else
808   %\def\MPLIBtoPDF#1{\special{pdf:literal direct #1}} % not ok yet
809   \def\mplibtoPDF#1{}
810   \expandafter\ifx\csname PackageWarning\endcsname\relax
811     \write16{}
812     \write16{Warning: MPLib only works in PDF mode, no figure will be output.}
813     \write16{}
814   \else
815     \PackageWarning{mplib}{MPLib only works in PDF mode, no figure will be out-
      put.}
816   \fi
817 \fi
818 \def\mplibsetupcatcodes{%
819   %catcode'\{=12 %catcode'\}=12
820   %catcode'\#=12
821   %catcode'\^=12 %catcode'\~=12 %catcode'\_ =12
822   %catcode'\%=12 %% don't in Plain!
823   %catcode'\&=12 %catcode'\$=12
824 }

    Make btex...etex box zero-metric.
825 \def\mplibputtextbox#1{\vbox to 0pt{\vss\hbox to 0pt{\raise\dp#1\copy#1\hss}}}}

    The Plain-specific stuff.
826 \bgroup\expandafter\expandafter\expandafter\egroup
827 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
828 \def\mplibcode{%
829   \begingroup

```

```

830 \bgroup
831 \mplibsetupcatcodes
832 \mplibdocode %
833 }
834 \long\def\mplibdocode#1\endmplibcode{%
835 \egroup
836 \expandafter\def\expandafter\mplibtmp\expandafter
837   {\directlua{luamplib.protecttexttext([==[\unexpanded{#1}]==])}}}%
838 \directlua{luamplib.tmpdata = luamplib.makeTEXboxes([==[\mplibtmp]==])}%
839 \directlua{luamplib.processwithTEXboxes(luamplib.tmpdata)}%
840 \endgroup
841 }
842 \else

    The  $\TeX$ -specific parts: a new environment.
843 \newenvironment{mplibcode}{\toks@{}\ltxdomplibcode}{}
844 \def\ltxdomplibcode{%
845 \begingroup
846 \mplibsetupcatcodes
847 \ltxdomplibcodeindeed %
848 }
849 %
850 \long\def\ltxdomplibcodeindeed#1\end#2{%
851 \endgroup
852 \toks@\expandafter{\the\toks@#1}%
853 \ifnum\pdf@strcmp{#2}{mplibcode}=\z@
854 \expandafter\def\expandafter\reserved@a\expandafter
855   {\directlua{luamplib.protecttexttext([==[\the\toks@]==])}}}%
856 \directlua{luamplib.tmpdata=luamplib.makeTEXboxes([==[\reserved@a]==])}%
857 \directlua{luamplib.processwithTEXboxes(luamplib.tmpdata)}%
858 \end{mplibcode}%
859 \else
860 \toks@\expandafter{\the\toks@\end{#2}}\expandafter\ltxdomplibcode
861 \fi
862 }
863 \fi

    \everymplib & \everyendmplib: macros redefining \everymplibtoks & \ev-
    eryendmplibtoks respectively
864 \newtoks\everymplibtoks
865 \newtoks\everyendmplibtoks
866 \protected\def\everymplib{%
867 \begingroup
868 \mplibsetupcatcodes
869 \expandafter\endgroup
870 \everymplibtoks
871 }
872 \protected\def\everyendmplib{%
873 \begingroup
874 \mplibsetupcatcodes
875 \expandafter\endgroup

```

```

876 \everyendmplibtoks
877 }
878 \def\mpdim#1{ begingroup \the\dimexpr #1\relax\space endgroup } % gmp.sty

    We use a dedicated scratchbox.
879 \ifx\mplibscratchbox\undefined \newbox\mplibscratchbox \fi

    We encapsulate the literals.
880 \def\mplibstarttoPDF#1#2#3#4{%
881   \hbox\bgroup
882   \xdef\MPllx{#1}\xdef\MPlly{#2}%
883   \xdef\MPurx{#3}\xdef\MPury{#4}%
884   \xdef\MPwidth{\the\dimexpr#3bp-#1bp\relax}%
885   \xdef\MPheight{\the\dimexpr#4bp-#2bp\relax}%
886   \parskip0pt%
887   \leftskip0pt%
888   \parindent0pt%
889   \everypar{}%
890   \setbox\mplibscratchbox\vbox\bgroup
891   \noindent
892 }

893 \def\mplibstoptoPDF{%
894   \egroup %
895   \setbox\mplibscratchbox\hbox %
896     {\hskip-\MPllx bp%
897      \raise-\MPlly bp%
898      \box\mplibscratchbox}%
899   \setbox\mplibscratchbox\vbox to \MPheight
900     {\vfill
901      \hsize\MPwidth
902      \wd\mplibscratchbox0pt%
903      \ht\mplibscratchbox0pt%
904      \dp\mplibscratchbox0pt%
905      \box\mplibscratchbox}%
906   \wd\mplibscratchbox\MPwidth
907   \ht\mplibscratchbox\MPheight
908   \box\mplibscratchbox
909   \egroup
910 }

    Text items have a special handler.
911 \def\mplibtexttext#1#2#3#4#5{%
912   \begingroup
913   \setbox\mplibscratchbox\hbox
914     {\font\temp=#1 at #2bp%
915      \temp
916      #3}%
917   \setbox\mplibscratchbox\hbox
918     {\hskip#4 bp%
919      \raise#5 bp%
920      \box\mplibscratchbox}%

```

```

921 \wd\mplibscratchbox0pt%
922 \ht\mplibscratchbox0pt%
923 \dp\mplibscratchbox0pt%
924 \box\mplibscratchbox
925 \endgroup
926 }

    That's all folks!
927 \</package>

```

3 The GNU GPL License v2

The GPL requires the complete license text to be distributed along with the code. I recommend the canonical source, instead: <http://www.gnu.org/licenses/old-licenses/gpl-2.0.html>. But if you insist on an included copy, here it is. You might want to zoom in.

<div><p>GNU GENERAL PUBLIC LICENSE</p><p>Version 2, June 1991</p><p>Copyright © 1989, 1991 Free Software Foundation, Inc.</p><p>51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA</p><p>Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.</p><p>Preamble</p><p>The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.</p><p>When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things. To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.</p><p>For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.</p><p>We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.</p><p>Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original author's reputations.</p><p>Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.</p><p>The precise terms and conditions for copying, distribution and modification follow.</p><p>TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION</p><ol style="list-style-type: none">This License applies to any program or other work which contains a notice placed by the copyright holder stating it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law, that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if it contains constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty, keep intact all the notices that refer to this License and to the absence of any warranty, and give any other recipients of the Program a copy of this License along with the Program.You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:<ol style="list-style-type: none">You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole or no charge to all third parties under the terms of this License.If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)<p>These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be</p></div> <div data-bbox="737 537 1029 1600"><p>on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.</p><p>Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.</p><p>In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.</p><ol style="list-style-type: none">You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:<ol style="list-style-type: none">Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)<p>The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.</p><p>If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.</p><ol style="list-style-type: none">You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program for any work based on the Program, you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims. This section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice. This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.</div> <div data-bbox="1045 537 1338 1600"><ol style="list-style-type: none">The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.<p>NO WARRANTY</p><ol style="list-style-type: none"><p>BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHERE OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.</p><p>IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY SUCCEED AND/OR REINSTATE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.</p><p>END OF TERMS AND CONDITIONS</p><p>Appendix: How to Apply These Terms to Your New Programs</p><p>If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.</p><p>To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.</p><p>one line to give the program's name and a brief idea of what it does. Copyright (C) yyyy name of author</p><p>This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.</p><p>This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.</p><p>You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.</p><p>Also add information on how to contact you by electronic and paper mail.</p><p>If the program is interactive, make it output a short notice like this when it starts in an interactive mode:</p><p>Gnomovision version 69, Copyright (C) yyyy name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.</p><p>The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items—whatever suits your program.</p><p>You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:</p><p>Yoyodyne, Inc., hereby disclaims all copyright interest in the program "Gnomovision" (which makes passes at compilers) written by James Hacker.</p><p>signature of Ty Coon, 1 April 1989 Ty Coon, President of Vice</p><p>This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.</p></div>
--