

The luaotfload package

Elie Roux* Khaled Hosny†

2010/07/12 v1.16

Abstract

This package is an adaptation of the ConT_EXt font loading system, providing the ability to load OpenType fonts with extended font loading syntax supporting a large selection of OpenType font features.

Contents

1	Introduction	2
2	Loading fonts	2
3	Font names database	4
3.1	Blacklisting fonts	5
4	Required ConT_EXt files	5
5	Troubleshooting	6
I	luaotfload.lua	6
5.1	Module loading	7
5.2	Post-processing TFM table	9
5.3	ConT _E Xt override	10
II	luaotfload.sty	10
6	The GNU GPL License v2	12

*elie.roux@telecom-bretagne.eu

†khaledhosny@eglug.org

1 Introduction

Font management and installation has always been painful with \TeX . A lot of files are needed for one font (tfm, pfb, map, fd, vf), and as \TeX is 8-bit each font is limited to 256 characters. But the font world has evolved since \TeX , and new font technologies have appeared, most notably the so called *smart font* technologies like OpenType fonts. These fonts can contain a lot of characters, and additional functionalities like ligatures, old-style numbers, small capitals, etc., and support more complex writing systems like Arabic and Indic¹ scripts. They are widely deployed and available for all modern operating systems and are becoming the de facto standard fonts for advanced text layout. Until now the only way to use them directly in the \TeX world was by using them with \XeTeX .

Unlike \XeTeX , \LuaTeX does not provide direct support for using these fonts by default, but it provides a way to hook Lua code in some points of the \TeX processing; for instance, we can improve the font loading system, and text procession, which what this package is about.

2 Loading fonts

luaotfload supports an extended font loading syntax which looks like:

```
\font\foo={\langle prefix\rangle:\langle font name\rangle:\langle font features\rangle} \langle \text{\TeX} font features\rangle
```

The curly brackets are optional and are used for escaping spaces in font names (double quotes can also be used for the same purpose).

Prefix The $\langle prefix \rangle$ be either `file:` or `name:`, which specify whether to use a select the font from its filename or font name, respectively. If no prefix is specified, then `file:` is assumed.

For compatibility with \XeTeX , surrounding the $\langle font name \rangle$ with square brackets is synonymous to using the `file:` prefix.

Accessing fonts by fontname allows loading system installed fonts as well as TEXMF ones, and requires a font names database; see Section 3 for more information.

Font name The $\langle font name \rangle$ can be either a font filename or actual font name based on the $\langle prefix \rangle$ as mentioned above.

Fonts loaded by filename may either include their absolute path in the filesystem or consist of just the filename with a path. If no path is specified then `kpathsea` is used to locate the font (which will typically be in the TEXMF tree or the current directory).

For example,

¹Unfortunately, luaotfload doesn't support Indic scripts right now

```

\font\1={file:ec-lmr10} at 10pt
\font\2={/Users/Shared/Fonts/aldus.otf} at 11pt
\font\3={name:TeX Gyre Pagella} at 9pt

```

Font features $\langle font\ features \rangle$ are a list of items separated by semi-colons, which are either **key=value** font parameters, or switches to enable/disable certain font features in the form of **+feat/-feat**. The supported keys are:

mode

luaotfload has two OpenType processing modes; **base** or **node**. Using **mode=base** only supports a subset of OpenType features and works by mapping those features to traditional T_EX ligature and kerning mechanisms and is a bit faster. Using **mode=node** hopefully supports OpenType fully and works by direct processing of the node list with Lua; it is slower and is not designed to work in math mode.

By default **mode=base** is used, but it is advisable to always enable **node** made, except for math fonts, otherwise many OpenType features will not function properly or even not work at all, especially for advanced scripts like Arabic.

script

OpenType script string, default value is **df1t**. Some fonts don't assign features to the **df1t** script, in which case the script need to be set explicitly.

language

OpenType language string, default value is **latn**.

featurefile

feature files are textual representation of OpenType tables and can be used to extend OpenType features of the font on fly. The file name of the feature file is passed, then features defined in the file can be enabled/disabled like any other feature. The actual syntax is described at <http://fontforge.sourceforge.net/featurefile.html> and http://www.adobe.com/devnet/opentype/afdko/topic_feature_file_syntax.html.

For example, to set a **tkrn** feature from **mykern.fea** file:

```
\font\lmr=Latin Modern Roman:featurefile=mykern.fea;+tkrn
```

color

font color, defined as a triplet of two-digit hexadecimal RGB values, with optionally another value for the transparency (where 00 is completely transparent and FF is opaque.)

For example, to set text in semitransparent red:

```
\font\lmr=Latin Modern Roman:color=FF0000BB
```

protrusion & expansion

Both keys control microtypographic features of the font, namely glyph protrusion and expansion. The value of the key is the name of predefined Lua tables of protrusion and expansion values; see the end of `otfl-font-dum.lua` file for an example of such tables. The only predefined value is `default`.

For example, to enable default protrusion²:

```
\font\lmr=Latin Modern Roman:protrusion=default
```

Non-standard font features `luaotfload` defines some additional font feature not defined in OpenType, currently three features are defined:

- **anum**: replaces European numbers with eastern Arabic numbers or Persian numbers, depending on the value of `language`.
- **tlig**: applies legacy T_EX ligatures (‘ ’ ’ ‘ -- -- ! ‘ ? ‘ <>>).
- **trep**: applies legacy T_EX replacements (‘ ’ ’ ’).

(For X_YT_EX users: these last two are the equivalent of writing `mapping=text-tex` using X_YT_EX's input remapping feature.)

3 Font names database

As introduced in the previous section, `luaotfload` uses a database to keep track of fonts available to LuaT_EX. Using this database, fonts can be loaded by font name as well as filename.

When `luaotfload` is asked to load a font by font name, it will check if font names database exists and load it, or generate a new database if non exists. This is all done automatically without user intervention. When the asked font is missing from the database, it will attempt to update the database and try to find the font again, so that the user can install new fonts without worrying about manually updating the database.

However, it is sometimes desirable to update the database manually, so `luaotfload` provides a `mkluatexfontdb` utility to manually update the database. `mkluatexfontdb` is a lua script that can be either run directly or as an argument to `texlua`, depending on your system³.

The first time the database is generated may take quite some time to process every font on your computer. This is particularly noticeable if it occurs during a typesetting run. Subsequent runs to update the database will be quite fast, however.

`luaotfload` will parse standard places for fonts in your system to build the font database. On Linux, it will read `fontconfig` configuration files to find the font locations; on Windows and Mac OS X, it will search in the standard font

²You also need to set `\pdfprotrudechars2` `\pdfadjustspacing2` to activate protrusion and expansion, respectively. See PDF_T_EX manual for details

³On MS Windows it can be run either by calling the wrapper application `mkluatexfontdb.exe` or with `texlua.exe mkluatexfontdb.lua`

locations, %WINDIR%\Fonts in Windows and ~/Library/Fonts, /Library/Fonts, /System/Library/Fonts, and /Network/Library/Fonts in Mac OS X.

If you do not wish the standard font locations be searched by default but would rather specify the exact locations in which to find your fonts, set the `OSFONTDIR` environment variable instead. When this variable is set, only the specified directories will be searched.

`mkluatexfontdb.lua --help` provides a brief summary of the functionality of the script and includes some advanced options that we have not mentioned here.

3.1 Blacklisting fonts

Some fonts are problematic in LuaTeX, if you found that your document takes too long to compile, or eats all the free memory, you can find the culprit file by running `mkluatexfontdb` utility with `-v` option to see which font file it is stuck with. You can then instruct `luaotfload` to ignore this font by adding it to the blacklist configuration file.

Simply, create a file named `otfl-blacklist.cnf` and added the to be black-listed files, one per line. Then put the file some where `kpse` can find. You can either use the base name or the full path. Any thing after a % sign is ignored.

4 Required ConTeXt files

This package is a wrapper for several files taken from the ConTeXt macro package. The philosophy is to let ConTeXt do all the implementation and update these files from time to time. To do so we did not modify the files taken from ConTeXt, we only changed their names to prevent name clashes. You can thus update the font system of this package simply by updating the files taken from ConTeXt, without (theoretically) changing the `.sty` file nor the main `.lua` file.

The ConTeXt files are renamed by adding the prefix `otfl-` to them (`otfl` as OTF Load). The files are:

- | | | |
|-----------------------------|-----------------------------|-----------------------------|
| • <code>luat-dum.lua</code> | • <code>font-ott.lua</code> | • <code>font-otc.lua</code> |
| • <code>data-con.lua</code> | • <code>font-otf.lua</code> | • <code>font-def.lua</code> |
| • <code>node-inj.lua</code> | • <code>font-otd.lua</code> | • <code>font-ctx.lua</code> |
| • <code>node-dum.lua</code> | • <code>font-oti.lua</code> | • <code>font-map.lua</code> |
| • <code>font-ini.lua</code> | • <code>font-otb.lua</code> | • <code>font-dum.lua</code> |
| • <code>font-tfm.lua</code> | • <code>font-otn.lua</code> | |
| • <code>font-cid.lua</code> | • <code>font-ota.lua</code> | |

The following files have been written for this package:

- | | | |
|-----------------------------|-----------------------------|-----------------------------|
| • <code>font-clr.lua</code> | • <code>font-nms.lua</code> | • <code>luat-ovr.lua</code> |
|-----------------------------|-----------------------------|-----------------------------|

5 Troubleshooting

If you encounter problems with some fonts, please first update to the latest version of this package before reporting a bug, as this package is under active development.

A very common problem is the lack of features for some OpenType fonts even when specified. It can be related to the fact that some fonts do not provide features for the `dflt` script, which is the default one in this package, so you may have to specify the script in the command line, for example:

```
\font\myfont = MyFont.otf:script=latn;+liga;
```

Also remember to set `mode=node` as most OpenType features (such as contextual substitution, `calt`), will not work without it.

File I

luaotfload.lua

First some usual initializations.

```
1 module('luaotfload', package.seeall)
2
3 luaotfload.module = {
4   name      = "luaotfload",
5   version   = 1.15,
6   date      = "2010/07/12",
7   description = "ConTeXt font loading system.",
8   author     = "Elie Roux & Hans Hagen",
9   copyright  = "Elie Roux",
10  license    = "CC0"
11 }
12
13 luatexbase.provides_module(luaotfload.module)
14
```

This is a necessary initialization in order not to rebuild an existing font. Maybe 600 should be replaced by `\pdfpkresolution` or `texconfig.pk_dpi` (and it should be replaced dynamically), but we don't have access (yet) to the `texconfig` table, so we let it be 600. Anyway, it does still work fine even if `\pdfpkresolution` is changed.

```
15
16 kpse.init_prog('', 600, '/')
17
```

Some helper functions.

```
18
19 local format = string.format
20
21 local function log(...)
22   luatexbase.module_log    ('luaotfload', format(...))

```

```

23 end
24
25 local function error(...)
26     luatexbase.module_error ('luaotfload', format(...))
27 end
28
29 local function warning(...)
30     luatexbase.module_warning('luaotfload', format(...))
31 end
32

```

The minimal required LuaTeX version.

```

33
34 local luatex_version = 60
35
36 if tex.luatexversion < luatex_version then
37     warning('LuaTeX v%.2f is old, v%.2f is recommended.',
38             tex.luatexversion/100,
39             luatex_version /100)
40 end
41

```

5.1 Module loading

We load the ConTeXt files with this function. It automatically adds the `otfl-` prefix to it, so that we call it with the actual ConTeXt name.

```

42
43 function luaotfload.loadmodule(name)
44     local tofind = 'otfl-'..name
45     local found = kpse.find_file(tofind,"tex")
46     if found then
47         log('loading file %s.', found)
48         dofile(found)
49     else
50         error('file %s not found.', tofind)
51     end
52 end
53

```

We start loading some lua files. These two are some code not used by ConTeXt at all that allow other modules to be used, it provides some low-level ConTeXt functions.

```

54
55 luaotfload.loadmodule('luat-dum.lua') -- not used in context at all
56 luaotfload.loadmodule('luat-ovr.lua') -- override some luat-dum functions
57 luaotfload.loadmodule('data-con.lua') -- maybe some day we don't need this one
58

```

A hack to remove a warning from `node-dum.lua` as it is ConTeXt specific.

```

59
60 tex.attribute[0] = 0
61

```

Node support modules.

```

62
63 luaotfload.loadmodule('font-ini.lua')
64 luaotfload.loadmodule('node-dum.lua')
65 luaotfload.loadmodule('node-inj.lua')
66

```

By default ConTEXt takes some private attributes for internal use. To avoid attribute clashes with other packages, we override the function that allocates new attributes, making it a wrapper around `luatexbase.new_attribute()`. We also prefix attributes with `otfl@` to avoid possible name clashes.

```

67
68 function attributes.private(name)
69     local attr = 'otfl@' .. name
70     local number = luatexbase.attributes[attr]
71     if not number then
72         number = luatexbase.new_attribute(attr)
73     end
74     return number
75 end
76

```

Font handling modules.

```

77
78 luaotfload.loadmodule('font-tfm.lua')
79 luaotfload.loadmodule('font-cid.lua')
80 luaotfload.loadmodule('font-ott.lua')
81 luaotfload.loadmodule('font-map.lua')
82 luaotfload.loadmodule('font-otf.lua')
83 luaotfload.loadmodule('font-otd.lua')
84 luaotfload.loadmodule('font-oti.lua')
85 luaotfload.loadmodule('font-otb.lua')
86 luaotfload.loadmodule('font-otn.lua')
87 luaotfload.loadmodule('font-ota.lua')
88 luaotfload.loadmodule('font-otc.lua')
89 luaotfload.loadmodule('font-def.lua')
90 luaotfload.loadmodule('font-xtx.lua')
91 luaotfload.loadmodule('font-dum.lua')
92

```

This is a patch for `otfl-font-def.lua`, that defines a reader for ofm fonts, this is necessary if we set the forced field of the specification to `ofm`, we use it only when using `luaotfload`, not `mkluatexfontdb`.

```

93
94 if fonts and fonts.tfm and fonts.tfm.readers then
95     fonts.tfm.readers.ofm = fonts.tfm.readers.tfm

```

```

96 end
97
    luaotfload specific modules.
98
99 luaotfload.loadmodule('font-nms.lua')
100 luaotfload.loadmodule('font-clr.lua')
101

```

5.2 Post-processing TFM table

Here we do some final touches to the loaded TFM table before passing it to the \TeX end.

```

102 local function def_font(...)
103     local fontdata = fonts.define.read(...)
104     if type(fontdata) == "table" and fontdata.shared then

```

First, we add some code to emulate \TeX 's `\fontdimen8`, which stores the caps-height of the font. (Cf. `\fontdimen5` which stores the x-height.)

Falls back to measuring the glyph if the font doesn't contain the necessary information. This needs to be extended for fonts that don't contain an 'X'.

```

105         local capheight
106         local units      = fontdata.units
107         local size        = fontdata.size
108         local otfddata    = fontdata.shared.otfddata
109
110         if otfddata.pfminfo.os2_capheight > 0 then
111             capheight = otfddata.pfminfo.os2_capheight / units * size
112         else
113             if fontdata.characters[string.byte("X")] then
114                 capheight = fontdata.characters[string.byte("X")].height
115             else
116                 capheight = otfddata.metadata.ascent / units * size
117             end
118         end
119         fontdata.parameters[8] = capheight

```

Then we populate MathConstants table, which is required for OpenType math.

```

120         if otfddata.metadata.math then
121             for k,v in next, otfddata.metadata.math do
122                 if k == "RadicalDegreeBottomRaisePercent" then
123                     -- this is a percent
124                     fontdata.MathConstants[k] = v
125                 else
126                     fontdata.MathConstants[k] = v / units * size
127                 end
128             end
129         end
130

```

```

131     end
132     return fontdata
133 end

```

5.3 ConT_EXt override

Here we override some defaults set in ConT_EXt code.

```

134
135 fonts.mode = "node"
136

```

Finally two functions

```

137
138 function luaotfload.register_callbacks()
139     luatexbase.add_to_callback('pre_linebreak_filter',
140                               nodes.simple_font_handler,
141                               'luaotfload.pre_linebreak_filter')
142     luatexbase.add_to_callback('hpack_filter',
143                               nodes.simple_font_handler,
144                               'luaotfload.hpack_filter')
145     luatexbase.reset_callback('define_font')
146     luatexbase.add_to_callback('define_font',
147                               def_font,
148                               'luaotfload.define_font', 1)
149     luatexbase.add_to_callback('find_vf_file',
150                               fonts.vf.find,
151                               'luaotfload.find_vf_file')
152 end
153
154 function luaotfload.unregister_callbacks()
155     luatexbase.remove_from_callback('pre_linebreak_filter',
156                                    'luaotfload.pre_linebreak_filter')
157     luatexbase.remove_from_callback('hpack_filter',
158                                    'luaotfload.hpack_filter')
159     luatexbase.remove_from_callback('define_font',
160                                    'luaotfload.define_font')
161     luatexbase.remove_from_callback('find_vf_file',
162                                    'luaotfload.find_vf_file')
163 end

```

File II

luaotfload.sty

Classical Plain+L^AT_EX package initialization.

```

1 \csname ifluaotfloadloaded\endcsname
2 \let\ifluaotfloadloaded\endinput

```

```

3
4 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
5   \input luatextra.sty
6 \else
7   \NeedsTeXFormat{LaTeX2e}
8   \ProvidesPackage{luaotfload}%
9     [2010/07/12 v1.16 ConTeXt font loading system]
10  \RequirePackage{luatextra}
11 \fi
12
13 \expandafter\edef\csname otfl@AtEnd\endcsname{%
14   \catcode64 \the\catcode64\relax
15 }
16
17 \catcode64 11
18

```

Two small macros to register or unregister the callbacks. Without the callbacks this package is totally turned off.

```

19
20 \def\otfl@off{
21   \directlua{luaotfload.unregister_callbacks()}
22 }
23
24 \def\otfl@on{
25   \directlua{luaotfload.register_callbacks()}
26 }
27

```

We load the lua file, and we turn the package on.

```

28
29 \luatexUseModule{luaotfload}
30
31 \otfl@on
32
33 \otfl@AtEnd

```

6 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.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright © 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

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.

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.

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.

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.

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 authors' reputations. 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.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

1. This License applies to any program or other work which contains a notice placed by the copyright holder saying 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 its contents 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.
2. 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 any option offer warranty protection in exchange for a fee.
3. 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:
 - (a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
 - (b) 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 at no charge to all third parties under the terms of this License.
 - (c) 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.)

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 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.

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.

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.

4. 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:
 - (a) 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,
 - (b) 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, 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,
 - (c) 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.)

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.

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, even though third parties are not compelled to copy the source code along with the object code.

5. 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 each remains in full compliance.
6. 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 (or 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.
7. 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.
8. 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.
9. 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.
10. 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.

11. 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.

NO WARRANTY

12. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN 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.
13. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE 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.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

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.

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.

one line to give the program's name and a brief idea of what it does.

Copyright (C) yyyy name of author

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.

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.

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.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

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.

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.

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:

Yoroyuz, Inc. hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a sublibrary, 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.