

LuaTeX-ja 简体中文字体配置

马起园 苏 杰

2013 年 5 月

1 LuaTeX-ja 项目简介

LuaTeX-ja 项目旨在将 pTeX 处理汉字的机制移植到 LuaTeX 下，当前项目的成员有：北川弘典、前田一貴、八登崇之、黒木裕介、阿部紀行、山本宗宏、本田知亮、齋藤修三郎和馬起園。

LuaTeX 虽然在编码上支持 Unicode，但并不能直接处理汉字断行以及禁则应用，而在 pTeX 系列中则能够处理断行并应用禁则，但并不支持 pdf 输出。LuaTeX 下处理汉字的断行和应用禁则需要使用其内建的诸多 callback 来进行。pTeX 扩展 TFM 为 JFM，但是 JFM 抽象程度更高，该文件涉及到的内容是字体的标点压缩，可以用于多个汉字字体。

当前版本的 LuaTeX-ja 可以应用于 TeX Live 和 W32TeX，但不能在 MikTeX 下使用，因为 MikTeX 下的 LuaTeX 存在 bug。

2 字体使用

2.1 关于 LuaTeX-ja 的字体调用

X_gTeX 在调用字体的时候需要使用 freetype 和 fontconfig 库，所以需要更新字体缓存，但是在 LuaTeX 下就不太一样。LuaTeX 只需要在初次使用的时候刷新字体数据库即可（此数据库是一个 lua 文件）。LuaTeX 下的字体缓存实际上是将字体的各种信息导出并保存的文件，在调用一个新字体的时候会生成一个。LuaTeX 对于字体的处理并没有使用 freetype，而是使用了 fontforge，不过可能由于该库的接口有一些问题，导致部分字体无法在 LuaTeX 下使用。

在使用 LuaTeX-ja 的时候，需要先刷新字体数据库，该命令的使用方法如下：

```
Usage: mkluatexfontdb [OPTION]...
```

```
Rebuild the LuaTeX font database.
```

```
Valid options:
```

```
-f --force           force re-indexing all fonts
-q --quiet           don't output anything
-v --verbose=LEVEL  be more verbose (print the searched directories)
-vv                  print the loaded fonts
-vvv                 print all steps of directory searching
-V --version         print version and exit
-h --help            print this message
```

又是可能遇到通不过的字体，需要将这个字体的绝对路径添加到 otf-blacklist.cnf 中，这个文件的内容如下：

```
% Tackes ages to load
LastResort.ttf % a MacOSX font, but also available for free from unicode.org

% Mac OS X TTC fonts, this list need to be filtered out
/Library/Fonts/AmericanTypewriter.ttc
/Library/Fonts/Baskerville.ttc
/Library/Fonts/Chalkboard.ttc
```

```

/Library/Fonts/Cochin.ttc
/Library/Fonts/Copperplate.ttc
/Library/Fonts/Didot.ttc
/Library/Fonts/Futura.ttc
/Library/Fonts/GillSans.ttc
/Library/Fonts/Hoefler Text.ttc
/Library/Fonts/MarkerFelt.ttc
/Library/Fonts/Optima.ttc
/Library/Fonts/Papyrus.ttc
/Library/Fonts/STHeiti Medium.ttc
/System/Library/Fonts/AquaKana.ttc
/System/Library/Fonts/HelveticaNeue.ttc
/System/Library/Fonts/LucidaGrande.ttc
/System/Library/Fonts/Menlo.ttc
/System/Library/Fonts/STHeiti Light.ttc

```

LuaTeX 调用 TrueType 和 OpenType 字体并没有固定的方式，XeTeX 使用了固定的接口，而 LuaTeX 需要使用 luaotfload 包来进行字体的调用。这两种字体一般都有高级特性，在 TeX Live 或者 W32TeX 中可以使用 `otfinfo` 命令来查看相关的信息：

```
'Otfinfo' reports information about an OpenType font to standard output.
Options specify what information to print.
```

```
Usage: otfinfo [-sfzpg | OPTIONS] [OTFFILES...]
```

Query options:

```

-s, --scripts           Report font's supported scripts.
-f, --features          Report font's GSUB/GPOS features.
-z, --optical-size     Report font's optical size information.
-p, --postscript-name  Report font's PostScript name.
-a, --family           Report font's family name.
-v, --font-version     Report font's version information.
-i, --info             Report font's names and designer/vendor info.
-g, --glyphs          Report font's glyph names.
-t, --tables           Report font's OpenType tables.
-T, --dump-table NAME  Output font's 'NAME' table.

```

Other options:

```

--script=SCRIPT[.LANG] Set script used for --features [latn].
-V, --verbose          Print progress information to standard error.
-h, --help             Print this message and exit.
-q, --quiet            Do not generate any error messages.
--version              Print version number and exit.

```

Report bugs to <ekohler@gmail.com>.

2.2 使用字体的方法

目前 LuaTeX-ja 支持在 plain TeX 和 LaTeX 下使用。如果你使用 texinfo，那么很不幸，你不太可能使用 LuaTeX-ja 来处理中文，因为 texinfo 是针对 pdfTeX 设计的，在 LuaTeX 下使用已经有了一定的不兼容现象，即使完全兼容也需要对 texinfo 中的字体配置进行调整，如果你急需使用 texinfo 来处理中文，请尝试 W32TeX 下的 texinfo，这个发行版中的 texinfo 已经打了补丁。对于 ConTeXt 用户，请使用李延瑞的 zhfonts 模块¹。

在 plain TeX 中使用 LuaTeX-ja 可以在源文件中写入：

```
\input luatexja-core.sty
```

¹见 <https://github.com/liyanrui/zhfonts>

LuaTeX-ja 移植了的`\jfont` 命令，在 plain TeX 下需要通过该命令来控制输出的汉字字体，例如：

```
\jfont\song={name:SimSun:jfm=banjiao} at 10pt
\song 我能吞下玻璃而不伤身体。
```

上文中的 `jfm=banjiao` 用来控制标点压缩的，如果此项未设定，则使用默认的 `ujis` 压缩模式，对于简体中文来讲，可用的模式有：`quanjiao`, `banjiao`, `kaiming`。

而在 LaTeX 下使用则较为简单，使用：

```
\usepackage{luatexja-fontspec}
```

这个包对 `fontspec` 包进行了封装，令其能够较为便利地设定汉字字体。这个包提供的命令如下：

命令	用途
<code>\jfontspec</code>	改变当前汉字字体
<code>\setmainfont</code>	设定文档主汉字字体
<code>\setsansfont</code>	设定文档的无衬线汉字字体（黑体）
<code>\newfontfamily</code>	设定新的汉字字体族命令
<code>\newfontface</code>	设定新的汉字字体命令
<code>\defaultjfontfeatures</code>	默认汉字字体的特性
<code>\addjfontfeatures</code>	设定当前字体的特性

2.3 不可用字体系列

中文字体在丰度上与日文字体对比并不占优势，所以中文 TeX 文档在使用使用字体上没有太大变化。当你在 LuaTeX-ja 下是有部分特色字体的时候，请做好此种字体可能无法使用的准备。目前报错明显的字体有数个：

- 灵格斯词典附带的音标字体，这些字体会安装到系统字体文件夹下，在更新字体数据库的时候会出现程序崩溃的情况，这是字体本身的原因，需要添加到黑名单中
- 康熙字典体，这是中国大陆一位业余字体设计者所设计的字体，由于该作者缺乏相关技术知识，导致此字体的 CMap 出错，无论是完全版还是试用版都会出现问题，此外该字体的 boundingbox 也是错误的，在嵌入 pdf 文档中十分影响阅读
- 信黑体，这个字体也是 CMap 的问题，无法使用

2.4 华文字体系列

在微软提供的 Office 套装中附带了一定数量的中文字体，这些字体是常州华文印刷新技术有限公司制造的。这些字体安装在系统字体文件夹下，在使用 LuaTeX-ja 的时候可以酌情使用。在本文档中，我们推荐简体中文用户使用此套字体，从使用率上看，各大学都会装有微软的操作系统和微软的 Office，可以说已经相当普及，故做推荐。

字体名	文件名	PostScript 名	样例
华文宋体	STSONG.TTF	STSong	我能吞下玻璃而不伤身体
华文中宋	STZHONGS.TTF	STZhongsong	我能吞下玻璃而不伤身体
华文细黑	STXIHEI.TTF	STXihei	我能吞下玻璃而不伤身体
华文楷体	STKAITI.TTF	STKaiti	我能吞下玻璃而不伤身体
华文仿宋	STFANGSO.TTF	STFangsong	我能吞下玻璃而不伤身体

2.5 中易字体系列

在 Windows 系统简体中文版中，附带了数种中文字体。这些字体为中易中标信息技术有限公司制造的。同我们强烈推荐的第一种方案比较，没有中宋。如果按照 CCT 的传统，一般使用黑体替换。

字体名	文件名	PostScript 名	样例
宋体	simsun.ttc	SimSun	我能吞下玻璃而不伤身体
黑体	simhei.ttf	SimHei	我能吞下玻璃而不伤身体
楷体	simkai.ttf	KaiTi	我能吞下玻璃而不伤身体
仿宋	simkai.ttf	FangSong	我能吞下玻璃而不伤身体

2.6 Adobe 字体系列

在 Adobe Reader 简体中文版中，附带了宋体和黑体两种字体。这两种字体实际上是华文字体，但是和华文字体不能混用，因为 Adobe Reader 中的中文字体的基线都进行了调整，不能互相匹配。在 Adobe InDesign 中还附带了楷体和仿宋体。Adobe 的中文字体的 Postscript 名即为文件名去掉后缀名。

字体名	文件名	样例
Adobe 宋体 Std	AdobeSongStd-Light.otf	我能吞下玻璃而不伤身体
Adobe 黑体 Std	AdobeHeitiStd-Regular.otf	我能吞下玻璃而不伤身体
Adobe 楷体 Std	AdobeKaitiStd-Regular.otf	我能吞下玻璃而不伤身体
Adobe 仿宋 Std	AdobeFangsongStd-Regular.otf	我能吞下玻璃而不伤身体

2.7 方正字体系列

方正字体的来源有两种，第一种是使用方正的排版系统的时候会安装到 Windows 系统的字体文件夹下，第二种是针对 Linux 系统来说的，WPS for Linux 附带了部分方正字体。

字体名	文件名	全名	样例
方正书宋_GBK	FZSSK.TTF	FZShuSong-Z01	我能吞下玻璃而不伤身体
方正小标宋_GBK	FZXBSK.TTF	FZXiaoBiaoSong-B05	我能吞下玻璃而不伤身体
方正黑体_GBK	FZHTK.TTF	FZHei-B01	我能吞下玻璃而不伤身体
方正楷体_GBK	FZKTK.TTF	FZKai-Z03	我能吞下玻璃而不伤身体
方正仿宋_GBK	FZFSK.TTF	FZFangSong-Z02	我能吞下玻璃而不伤身体