

LuaTeX-ja 宏包

LuaTeX-ja 项目团队

目次

第 I 编 用户手册	2
1 引言	2
1.1 开发背景	2
1.2 与 pTeX 的差异所在	2
1.3 一些约定	3
1.4 关于本项目	3
2 初次使用	3
2.1 安装	3
3 LuaTeX-ja 在 plain TeX 格式下的使用	4
4 LuaTeX-ja 在 LaTeX 2_ε 格式下的使用	4
4.1 日文排版设置	4
4.2 fontspec	5
5 LuaTeX-ja 参数设定	5
5.1 JAchar 范围的设定	5
5.2 kanjiskip 和 xkanjiskip	7
5.3 xkanjiskip 插入设定	7
5.4 基线浮动	8
5.5 裁剪框标记	8

第 I 编

用户手册

1 引言

LuaTeX-ja 宏包是应用于 LuaTeX 引擎上的高质量日语文档排版宏包。

1.1 开发背景

一般情况下, TeX 下的日语文档输出, 是 ASCII pTeX (TeX 的一个扩展) 及其衍生软件来完成的。pTeX 作为 TeX 的一个扩展引擎, 在生成高质量的日语文档时, 规避了繁杂的宏编写。但是在和同时期的引擎相比之下,

pTeX 的处境未免有些尴尬: pTeX 已经远远落后于 ϵ -TeX 和 pdfTeX, 此外也没有跟上计算机上对日文处理的演进 (比如, UTF-8 编码, TrueType 字体, OpenType 字体)。

最近开发的 pTeX 扩展, 即 upTeX (Unicode 下的 pTeX 实现) 和 ϵ -pTeX (pTeX 和 ϵ -TeX 的融合版本), 虽然在部分情况上弥补了上述的差距, 但是差距依然存在。

不过, LuaTeX 的出现改变了整个状况。用户可以通过使用 Lua 语言的 “callback” 来调整 LuaTeX 的内部处理机制。所以, 没有必要去通过修改引擎的源代码来支持日文排版, 相反, 我们需要做的仅仅是编写其当处理 callback 的 Lua 脚本。

1.2 与 pTeX 的差异所在

LuaTeX-ja 宏包在设计上, 受 pTeX 影响很大。最初开发的主要议题是实现 pTeX 的特性。不过, **LuaTeX-ja 不是简简单单的移植 pTeX, 很多不自然的特征和现象都被移出了。**

下面列举出了一些和 pTeX 的差异:

- 一个日文字体是由三部分构成的元组: 实际的字体 (如小塚明朝, IPA 明朝), 日文字体测度 (JFM) 和变体字符串。
- pTeX 中, 日文字符之后的断行并不允许 (也不产生空格), 其他在源码中的断行是可以随处允许的。不过, 因为 LuaTeX 的特殊关系, LuaTeX-ja 并没有这个功能。
- 插在日文字符和其他字符之间的胶/出格 (我们将此称为 **JAgglue**) 是重现实现的。
 - 在 LuaTeX 中, 内部的字符处理是 “基于 node 的” (例如: `of{}fice` 不会避免合字), **JAgglue** 的插入处理, 现在也是 “基于 node 的”。
 - 此外, 两个字符之间的 node 在断行时不起作用的 (例如, `\specialnode`), 还有意大利体校正带来的出格在插入处理中也是被忽略的。
 - **警告: 鉴于以上两点, 在 pTeX 中分割 JAgglue 处理的多种方法不再生效。** 明确地说, 下列两种方法不再生效:

```
    ちよ{}つと    ちよ\つと
```

如果想得到此种结果, 请使用空盒子替代:

```
    ちよ\hbox{}つと
```
 - 处理过程中, 两个在 “真实” 字体上具区别的日文字体可以被识别出来。
- 当下, LuaTeX-ja 并不支持直行排版。

1.3 一些约定

在本文档中, 有下面一些约定:

- 所有的日文字符为 **JAchar**, 所有的其他字符为 **ALchar**
- `primitive`, 该词在本文档中不仅表示 LuaTeX 的基本控制命令, 也包括 LuaTeX-ja 的相关的基本控制命令
- 所有的自然数从 0 开始
- 本文档中文档是根据日文文档翻译并添加部分材料组织而成的, 所以在大部分篇幅上还是主要依赖于日文原文档

1.4 关于本项目

项目 wiki 本项目 wiki 正在不断编写中。

- <http://sourceforge.jp/projects/luatex-ja/wiki/FrontPage%28en%29> (英文)
- <http://sourceforge.jp/projects/luatex-ja/wiki/FrontPage> (日文)
- <http://sourceforge.jp/projects/luatex-ja/wiki/FrontPage%28zh%29> (中文)

开发者

- 北川弘典
- 前田一贵
- 八登崇之
- 黒木裕介
- 阿部纪行
- 山本宗宏
- 本田知亮
- 斋藤修三郎
- 马起园

2 初次使用

2.1 安装

在安装 LuaTeX-ja 之前, 请确认:

- LuaTeX (版本号为大于 0.65) 和相关支持宏包。
如果用户使用的是 TeXLive2011 以及最新版本的 W32TeX, 都可以不考虑此项。
- LuaTeX-ja 的源码:)
- `xunicode` 宏包, 当前版本必须为 *v0.981(2011/09/09)*。
如果你使用 `fontspec` 宏包, `xunicode` 必须存在。但是请注意该包版本, 其他版本可能不会正常工作。

安装方法如下:

1. 用下列诸多方法下载源码归档。当前, LuaTeX-ja 并无稳定版本。
 - 从 git 中复制

```
$ git clone git://git.sourceforge.jp/gitroot/luatex-ja/luatexja.git
```
 - 下载 master 分支 HEAD 的 tar.gz 归档
<http://git.sourceforge.jp/view?p=luatex-ja/luatexja.git;a=snapshot;h=HEAD;sf=tgz>
 - 现在 LuaTeX-ja 已经包含于 CTAN 中 (在 `macros/luatex/generic/luatexja` 文件夹) 和 W32TeX 中 (`luatexja.tar.gz`)。这些版本基于 master 版本。
2. 用户需要将获取的 LuaTeX-ja 宏包的 `src/` 放置到你所用的发行版中的 `TEXMFLOCAL` 树下 `tex/` 文件夹 (如果不存在, 请建立) 下即可, 查询该环境变量需要使用:

```
$ kpeswhich -expand-var "$TEXMFLOCAL"
```
3. 刷新数据库:

```
$ texhash  
$ mkluatexfontdb
```

3 LuaTeX-ja 在 plain TeX 格式下的使用

对于日本用户，日文的排版可以直接使用：

```
\input luatexja.sty
```

这一行会默认调用 `luatexja.sty` 文件，该文件作用相当于 pTeX 中的 `ptex.tex`。该文件中设定了两款非嵌入字体：Ryumin-Light（明朝体）和 GothicBBB-Medium（哥特体）。这些字体分别有三种大小：`\tenmin` 和 `\tengt`（10pt）；`\sevenmin` 和 `\seventgt`（7pt）；`\fivemin` 和 `\fivegt`（5pt）。

有几点请用户注意：

- 上述的两款字体需要你的 pdf 阅读器带有良好的非嵌入字体的支持，这里建议使用 Adobe Reader，该阅读器使用了 Kozuka Mincho 字体。
- 在定义字体的时候，可以使用“级”（Q）单位， $1Q = 0.25\text{mm}$ ，在 LuaTeX-ja 中可以使用 `\jq` 命令调用该长度单位
- 一般情况下，相同大小日文字体比西文字体要大一下（中文字体也是如此），所以定义日文字体的时候需要一个缩放率：0.962216
- 在 `JAchar` 和 `ALchar` 之间插入的胶大小为： $(0.25 \times 0.962216 \times 10\text{pt})_{-1\text{pt}}^{+1\text{pt}} = 2.40554\text{pt}_{-1\text{pt}}^{+1\text{pt}}$

4 LuaTeX-ja 在 LaTeX 2_ε 格式下的使用

4.1 日文排版设置

在 LaTeX 2_ε 下使用 LuaTeX-ja 比较便利：

```
\usepackage{luatexja}
```

该宏包的作用相当于 pLaTeX 中的 `plfonts.dtx` 和 `pldefs.ltx`。

- 在该宏包中设定了 JY3 编码，这个编码用来调用日文字体
- 该宏包定义了两个字体族：`mc` 和 `gt`。

字体	字体族	<code>\mdseries</code>	<code>\bfseries</code>	缩放率
<i>mincho</i>	<code>mc</code>	Ryumin-Light	GothicBBB-Medium	0.962216
<i>gothic</i>	<code>gt</code>	GothicBBB-Medium	GothicBBB-Medium	0.962216

- 在数学模式下，所有的字符使用 `mc` 字体族来排印

上述使用的宏包，只满足了最小的日文环境设定需求，并为满足所有的日文排版需求。LuaTeX-ja 宏包提供了两个文档类：`jcasses`（pLaTeX 标准文档类）和 `jsclasses`（奥村晴彦）。用户可自行选择两种文档类。

4.2 fontspec

在 LuaTeX-ja 中，和 `fontspec` 共存的一个包为 `luatexja-fontspec`。这个包自动加载 `fontspec` 和 `luatexja` 宏包。

在 `luatexja-fontspec`，定义了如下七条命令，这些命令和 `fontspec` 的相关命令对比如下：

日文字体	<code>\jfontspec</code>	<code>\setmainjfont</code>	<code>\setsansjfont</code>	<code>\newfontfamily</code>
西文字体	<code>\fontspec</code>	<code>\setmainfont</code>	<code>\setsansfont</code>	<code>\newfontfamily</code>
日文字体	<code>\newfontface</code>	<code>\defaultjfontfeatures</code>	<code>\addjfontfeatures</code>	
西文字体	<code>\newfontface</code>	<code>\defaultfontfeatures</code>	<code>\addfontfeatures</code>	

5 LuaTeX-ja 参数设定

LuaTeX-ja 包含大量的参数,以控制排版细节。设定这些参数需要使用命令: `\ltjsetparameter` 和 `\ltjgetparameter` 命令。

5.1 JAchar 范围的设定

在设定 **JAchar** 之前,需要分配一个小于 217 的自然数。如:

```
\ltjdefcharrange{100}{"10000-"1FFFF,'漢}
```

请注意这个设定是全局性的,不建议在文档正文中进行设定。

在范围设定好了之后,需要进行 `jacharrange` 的设定:

```
\ltjsetparameter{jacharrange={-1, +2, +3, -4, -5, +6, +7, +8}}
```

这里定义了 8 个范围,在每个范围之前使用 “+” 或 “-” 进行设定,其中如果为 -, 则代表该范围为 **ALchar**, 如果为 +, 则该范围视作 **JAchar**。

LuaTeX-ja 默认设定了 8 个范围,这些范围来源于下列数据:

- Unicode 6.0
- Adobe-Japan1-6 与 Unicode 之间的映射 `Adobe-Japan1-UCS2`
- 八登崇之的 upTeX 宏包: `PXbase`

范围 8^J ISO 8859-1 (Latin-1 补充) 的上半部和 JIS X 0208 (日文基本字符集) 的重叠部分,包含下列字符:

- | | |
|-------------------------------|-----------------------------------|
| • § (U+00A7, Section Sign) | • (U+00B4, Spacing acute) |
| • ¨ (U+00A8, Diaeresis) | • (U+00B6, Paragraph sign) |
| • ° (U+00B0, Degree sign) | • × (U+00D7, Multiplication sign) |
| • ± (U+00B1, Plus-minus sign) | • ÷ (U+00F7, Division Sign) |

范围 1^A 包含于 Adobe-Japan1-6 中的拉丁字符,此范围包含下列 Unicode 区域,但不包括上述提到过的范围 8:

- | | |
|---------------------------|---------------------------|
| • U+0080–U+00FF: 拉丁字母补充-1 | • U+02B0–U+02FF: 进格修饰符元 |
| • U+0100–U+017F: 拉丁字母扩充-A | • U+0300–U+036F: 组合音标附加符号 |
| • U+0180–U+024F: 拉丁字母扩充-B | • U+1E00–U+1EFF: 拉丁字母扩充附加 |
| • U+0250–U+02AF: 国际音标扩充 | |

范围 2^J 希腊文和西里尔字母,使用 JIS X 0208 的大部分日文字体包含这些字符:

- U+0370–U+03FF: 希腊字母
- U+0400–U+04FF: 西里尔字母
- U+1F00–U+1FFF: 希腊文扩充

范围 3^J 标点以及杂项符号:

- U+2000–U+206F: 一般标点符号
- U+2070–U+209F: 上标及下标
- U+20A0–U+20CF: 货币符号
- U+20D0–U+20FF: 符号用组合附加符号
- U+2100–U+214F: 类字母符号
- U+2150–U+218F: 数字形式
- U+2190–U+21FF: 箭头符号
- U+2200–U+22FF: 数学运算符号
- U+2300–U+23FF: 杂项技术符号
- U+2400–U+243F: 控制图像
- U+2500–U+257F: 制表符
- U+2580–U+259F: 区块元素
- U+25A0–U+25FF: 几何形状
- U+2600–U+26FF: 杂项符号
- U+2700–U+27BF: 什锦符号
- U+2900–U+297F: 补充性箭头-B
- U+2980–U+29FF: 混合数学符号-B
- U+2B00–U+2BFF: 杂项符号和箭头符号
- U+E000–U+F8FF: 私用区域

范围 4^A 通常情况下不包含于日文字体的部分。本范围包含有其他范围尚未涵盖部分。故，我们直接给出定义:

```
\ltjdefcharrange{4}{%
    "500-"10FF, "1200-"1DFF, "2440-"245F, "27C0-"28FF, "2A00-"2AFF,
    "2C00-"2E7F, "4DC0-"4DFF, "A4D0-"A82F, "A840-"ABFF, "FB50-"FE0F,
    "FE20-"FE2F, "FE70-"FEFF, "FB00-"FB4F, "10000-"1FFFF} % non-Japanese
```

范围 5^A 代替以及补充私有使用区域。

范围 6^J 日文字符。

- U+2460–U+24FF: 圈状字母数字
- U+2E80–U+2EFF: CJK 部首补充
- U+3000–U+303F: CJK 标点符号
- U+3040–U+309F: 平假名
- U+30A0–U+30FF: 片假名
- U+3190–U+319F: 汉文标注号
- U+31F0–U+31FF: 片假名音标补充
- U+3200–U+32FF: 圈状 CJK 字母及月份
- U+3300–U+33FF: CJK 兼容
- U+3400–U+4DBF: CJK 统一表意文字扩充 A
- U+4E00–U+9FFF: CJK 统一表意文字
- U+F900–U+FAFF: CJK 兼容表意文字
- U+FE10–U+FE1F: 直行标点
- U+FE30–U+FE4F: CJK 兼容形式
- U+FE50–U+FE6F: 小写变体
- U+20000–U+2FFFF: (补充字符)

范围 7^J 不包含于 Adobe-Japan1-6 的 CJK 字符。

- U+1100–U+11FF: 谚文字母
- U+2F00–U+2FDF: 康熙部首
- U+2FF0–U+2FFF: 汉字结构描述字符
- U+3100–U+312F: 注音字母
- U+3130–U+318F: 谚文兼容字母
- U+31A0–U+31BF: 注音字母扩充
- U+31C0–U+31EF: CJK 笔划
- U+A000–U+A48F: 彝文音节

- U+A490–U+A4CF: 彝文字母
- U+A830–U+A83F: 一般印度数字
- U+AC00–U+D7AF: 谚文音节
- U+D7B0–U+D7FF: 谚文字母扩充-B

5.2 kanjiskip 和 xkanjiskip

JAglue 分为下列三类范畴:

- JFM 设定的胶或出格值。如果在一个日文字符附近使用 `\inhibitglue`, 则胶便不会插入。
- 两个 **JA**char 之间默认插入的胶 (`kanjiskip`)
- **JA**char 和 **AL**char 之间默认插入的胶 (`xkanjiskip`)

`kanjiskip` 和 `xkanjiskip` 的设定如下所示:

```
\ltjsetparameter{kanjiskip={0pt plus 0.4pt minus 0.4pt},
                  xkanjiskip={0.25\zw plus 1pt minus 1pt}}
```

当 JFM 包含“`kanjiskip` 理想宽度”和/或“`xkanjiskip` 理想宽度”数据时, 上述设定产生作用。如果想用 JFM 中的数据, 请设定 `kanjiskip` 或 `xkanjiskip` 为 `\maxdimen`。

5.3 xkanjiskip 插入设定

并不是在所有的 **JA**char 和 **AL**char 周围插入 `xkanjiskip` 都是合适的。比如, 在开标点之后插入 `xkanjiskip` 并不合适 [如, 比较“(あ”和“(あ”]。LuaTeX-ja 可以通过设定 **JA**char 的 `jaxspmode` 以及 **AL**char 的 `alxspmode` 来控制 `xkanjiskip` 在字符前后的插入。

```
1 \ltjsetparameter{jaxspmode={'あ,preonly},
                  alxspmode={'\!,postonly}}
2 pあq い!う
```

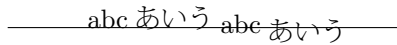
第二个参数 `preonly` 表示的含义为“允许在该字符前插入 `xkanjiskip`, 但不允许在该字符之后插入”。其他参数还有 `postonly`, `allow` 和 `inhibit`。[TODO]

用户如果想开启/关闭 `kanjiskip` 和 `xkanjiskip` 的插入, 设定 `autospacing` 和 `autoxspacing` 参数为 `true/false` 即可。

5.4 基线浮动

为了确保日文字体和西文字体能够对其, 有时需要浮动其中一者的基线。在 pTeX 中, 此项设定由设定 `\yabaselineshift` 为非零长度 (西文字体基线应向上下浮动)。不过, 如果文档中的主要语言不是日文, 那么最好上浮日文字体的基线, 西文字体不变。如上所述, LuaTeX-ja 可以独立设定西文字体的基线 (`yabaselineshift` 参数) 和日文字体的基线 (`yjabaselineshift` 参数)。

```
1 \vrule width 150pt height 0.4pt depth 0pt\hskip
   -120pt
2 \ltjsetparameter{yjabaselineshift=0pt,
                  yalbaselineshift=0pt}abcあいう
3 \ltjsetparameter{yjabaselineshift=5pt,
                  yalbaselineshift=2pt}abcあいう
```



上述水平线为此行基线。

这里还有一个有趣的副作用: 不同大小的字符可以通过适当调整这两个参数而在一行中垂直居中。下面是一个例子 (注意, 参数值并没有精心调整):

```

1 xyz漢字
2 {\scriptsize
3 \ltjsetparameter{yjbaselineshift=-1pt,          xyz 漢字 XYZ ひらがな abc かな
4   yalbaselineshift=-1pt}
5   XYZひらがな
6 }abcかな

```

5.5 裁剪框标记

裁剪框标记是在一页的四角和水平/垂直中央放置的标记。在日文中，裁剪框被称为“トンボ”。pL^AT_EX 和 LuaT_EX-j_a 均在底层支持裁剪框标记。需要下列步骤来实现：

1. 首先，首先定义页面左上角将会出现的注记。这由向**@bannertoken** 分配一个 token 列完成。例如，下列所示将会设定注记为“filename (YYYY-MM-DD hh:mm)”：

```

\makeatletter

\hour\time \divide\hour by 60 \@tempcnta\hour \multiply\@tempcnta 60\relax
\minute\time \advance\minute-\@tempcnta
\@bannertoken{%
  \jobname\space(\number\year-\two@digits\month-\two@digits\day
  \space\two@digits\hour:\two@digits\minute)}%

```

2. [TODO]