

# musixjnp 宏包

## TeX 排版系统下的简谱解决方案

$\beta$ -1.1 版本

深圳外国语学校高三（21）班 方惟佳

2019 年 3 月 7 日修订

对于不熟悉 TeX 的人，我推荐使用其他软件进行音乐排版。在电脑上设置 TeX 和 MusiXTeX 并熟练使用这两款软件是一项大工程，需要耗费很多时间和磁盘空间。

但是，一旦掌握了的话……

——汉斯·古伊根斯，约 1995 年

我认为，上面这句话过时了。

——克里斯多弗·比布里彻，2006 年

本宏包基于 GPL 协议（最新版本）公开，可以照协议自由复制、使用。任何人可以将其全部或部分代码任意使用，但是不能将其自己的产品称作 **musixjnp**，除非是修正程序的漏洞（如本版本仍未解决的`\ifnum`嵌套问题）。

对于某些实现（如字体、减时线连接机制）的改动需要以单独的  $\text{\TeX}$  或  $\text{\LaTeX}$  文档实现。

# 前言

musixjnp 是本人在 MusiX<sub>TEX</sub> 宏集的基础上实现的简谱排版系统，在开发过程中也参考了 musixgre 和 musixper 的代码。在编写过程中，QQ 群上的几位朋友也对我的工作在字体等方面给予了相应的帮助。目前  $\beta$ -1.1 版本的最后一个大更新是平连线的制作。

在  $\text{T}_{\text{E}}\text{X}$  下，五线谱排版已经有 **PMX**（器乐）、**M-Tx**（声乐）等预处理器。它们的存在简化了用户的学习量和输入。但是，对于排版简谱或线一简混排乐谱，目前仍旧需要学习 musixjnp 宏包及 MusiX<sub>TEX</sub> 宏集的相关语法。

适用于简谱的预处理器目前也正在制作中；在预处理器完成之后，用户只需学习预处理器下的简化语法，多数情况下已不再需要了解 MusiX<sub>TEX</sub> 宏集的相应语法。

MusiX<sub>TEX</sub>、**PMX**、**M-Tx** 等的输入可以通过任意的文本编辑器完成。目前，本宏包（包括 MusiX<sub>TEX</sub> 宏集中的其他包）不存在可视化的输入界面。

**注意：**本文还对 MusiX<sub>TEX</sub> 宏集说明书中与简谱相关的部分做了较为简略的翻译；只与五线谱相关的部分则略去。

方惟佳

2019 年 3 月 7 日

# 目录

前言	iii
目录	iv
第一章 MusiX <sub>TEX</sub> 及 musixjnp 简介	1
1.1 音乐排版的特点	1
1.1.1 音乐排版是二维的	1
1.1.2 水平间距	2
1.1.3 输入内容	2
1.1.4 符杠及减时线	3
1.1.5 输入文本	3
1.2 乐谱示例	3
1.3 三段系统	4
1.3.1 简介	4
1.3.2 例子	4
1.3.3 手动改变布局	6
1.3.4 musixflx 的使用	7
1.3.5 致新手：注意空格！	8
1.4 其他	8
1.4.1 从乐谱中提取声部	8
1.4.2 乐谱及音符的大小	8
第二章 输入文档的格式	9
2.1 MusiX <sub>TEX</sub> 的基本规定	9
2.2 常用的设置命令	9
2.2.1 全局乐谱尺寸	9
2.2.2 乐器个数	9
2.2.3 谱号	9
2.2.4 节拍	10
2.2.5 乐器名	10

2.2.6	乐器组 . . . . .	10
2.3	纯简谱谱表常用的设置 . . . . .	10
<b>第三章</b>	<b>输入音符的预备工作</b>	<b>11</b>
3.1	谱表的水平间距 . . . . .	11
3.1.1	音符间距 . . . . .	11

# 第一章 MusiX<sub>TEX</sub> 及 musixjnp 简介

本章不是 MusiX<sub>TEX</sub> 宏集或 musixjnp 宏包的教程，而是对二者能力的概览。

MusiX<sub>TEX</sub> 是一系列宏包及字体的集合，用于在 <sub>TEX</sub> 系统下进行音乐排版。使用该系统时，电脑上需要有正常工作的 <sub>TEX</sub> 发行版。musixjnp 是我在 MusiX<sub>TEX</sub> 及相关宏包上的扩展，目前版本 (β-1.1) 已经能够完成较简单的简谱乐曲及简线混排乐曲。

musixjnp 宏包正如同字模，有音符数字、连线、附加符号等的字形，可以用于排版质量合格的简谱谱面。用户需要通过输入相应的命令，手动指定每个“逻辑符号”的位置。在预处理器完成之后，也可以使用与水平位置无关的命令简化输入。在每个“逻辑符号”（如简谱音符）中，每个字符（如上下加点、连线）的位置均由程序自己计算，足以满足一般需求。

对于歌词，目前可以使用插入文本的方式输入。musixlyr 宏包目前暂不支持简谱，将来可以通过重写命令的方式增加支持。

本宏包是依据 plain <sub>TEX</sub> 开发的，但是在 L<sub>ATEX</sub> 上应该也能正常使用。一般来说，纯乐谱的排版并不需要 L<sub>ATEX</sub>；L<sub>ATEX</sub> 只需要在需要插入简谱示例的文档（如本说明）中使用。将来针对 L<sub>ATEX</sub> 进行的最大工程将会是使用 NFSS 机制进行“文本中插入简谱音符本身”一类的需求，及编写适当的 .sty 文件。

## 1.1 音乐排版的特点

### 1.1.1 音乐排版是二维的

除笛子、人声等的乐谱外，多数乐谱都以二维形式存在：横向是时间（节拍、节奏），在同一时刻内可以竖向排列多个音符，称为“簇”。与此同时，文件本身却是一维的。因此，较有逻辑的音符排列方法正如图 1.1 所示。因此，MusiX<sub>TEX</sub> 宏集中将一个簇（上图中的一列）记录为

$$\backslash notes \dots \& \dots \& \dots \backslash en^1$$

形式，其中 & 用于分割自下而上排列的乐器（或声部）。当一个乐器需要使用多个声部时，各个声部使用 | 字符分隔。

---

<sup>1</sup>命令 `\en` 是 `\enotes` 的缩写，两者等价。

3	6	9	12
2	5	8	11
1	4	7	10

图 1.1: 音符在一维文件中的逻辑排列方法

因此，若输入独唱配钢琴伴奏的乐谱，则应使用

```
\notes ... | ... & ...\en
```

表示每一簇。

每一个簇中，一行谱表上不仅可以放置同时出现的和弦音符，也可以放置较短的连续音符。因此，这表示 MusiX<sub>TEX</sub> 宏集中需要有两种命令，占位命令和不占位命令。后一类命令可以用于表示和弦、演奏记号等。

### 1.1.2 水平间距

计算乐谱的水平间距十分复杂，不在此文中讨论。水平间距的选择主要是通过 3.1 节描述的内容完成的。通过改变上文中 `\notes` 对应的命令，可以指定相对水平间距。

### 1.1.3 输入内容

`musixjnp` 宏包通过类比 MusiX<sub>TEX</sub> 宏集相应的代码，将输入分为以下几类。

- 简谱音符（增时线视作一种特殊的音符）
- 减时线
- 简谱连音线（分为弯连线、平连线）的开始、结束
- 延音线、连音线的开始、结束
- 升降号
- 演奏记号
- 小节线
- 拍号、调号等

在五线谱中，字母表示绝对音高。例如，命令 `\wh a` 表示高 220 Hz 的全音符，`\wh h` 高一个八度等。在简谱中，字母 `c` 则只表示中音 do。

五线谱中的和弦使用 `\zq` 表示无符干的音符进行组合。简谱目前暂不支持和弦。

### 1.1.4 符杠及减时线

这两者都是使用一对命令产生的。第一个命令指明了符杠或减时线的数目和序号。对于符杠来说，还指定了横向位置（默认为当前）、高度、方位（指上方或下方）、斜度。第二个命令指名了结束位置和序号。引入序号的目的是使得符杠或减时线可以重叠。

### 1.1.5 输入文本

`\zcharnote` 命令可以在谱面的任何横、纵位置输入任何符号组合。这项功能允许用户将自定义符号任意地加入到乐谱中。

## 1.2 乐谱示例

下例是《喀秋莎》选段。



相应的代码如下所示。

```
\smallmusicsize % 简谱一般使用较小的字号
\sepbarrules % 简谱小节线不连通
\nobarnumbers % 无小节计数
\instrumentnumber{2}
% 两个乐器（声部）
\setclef18
% 下方声部无五线谱谱号
\setlines10 % 无谱线
\setclef28 % 上方声部相同
\setlines20
\akkoladen{{1}{2}} % 方括号连接
\startextract % 乐谱开始
\Notes\jq{efe}&\jq{ehg}\en
\Notes\Djul fe&\Djul hg\en\bar
\Notes\jq d&\jq f\en
\Notes\Djul cb&\Djul ed\en
\Notes\jq{ca}&\jq{ea}\en\bar
\Notes\jc R&\jc R\en
\Notes\jq d&\jq f\en
\Notes\jc b&\jc d\en
\Notesp\jqp c&\jqp e\en
\Notes\jc c&\jc c\en\bar
\Notes\Djul bL\Djul cb&\Djul bL\Djul cb\en
\Notes\jq{aZ}&\jq{aZ}\en
\endextract % 乐谱结束
```



## 1.3 三段系统

### 1.3.1 简介

$\text{T}_{\text{E}}\text{X}$  默认的断行算法适合于文本，因为每行西文文本都会有足够多的空格，无须过大地调整就可以实现两端对齐。但是，五线谱、简谱往往一行不超过 5 小节，因此小节内部也需要做相应调整——这便超出了  $\text{T}_{\text{E}}\text{X}$  的处理能力。为解决五线谱两端对齐的计算，MusiX<sub>TEX</sub> 原作者发明了一套三段系统。

首先，MusiX<sub>TEX</sub> 系统将水平距离分为两大类：**不可变和可变**。小节线、谱表等前后的距离是不可变的，而音符、休止符前后的间距是可成比例伸缩的。在这个意义上，可变的间距可以定义为基本间距（记为 `\elemskip`）的倍数。例如，**PMX** 中所有的十六分音符一般宽 `1.41\elemskip`。

MusiX<sub>TEX</sub> 的工作之一便是计算每行中 `\elemskip` 的宽度。正确的宽度应该使得每一行都恰好“撑满”行内的所有空间。显然，这个值在每行都不尽相同。

三段系统的第 1 段是使用 `etex` 命令（或其他 plain  $\text{T}_{\text{E}}\text{X}$  引擎）编译原文件，生成后缀为 `.mx1` 的文件。该文件由行宽、段落缩进等信息开始，并在之后列出每小节相应的可变宽度、不可变宽度。

第 3 段的程序是 `musixflx` 程序，用于计算相应的断行和 `\elemskip` 在每行的值，使得乐谱间距匀称，并刚好在行末结尾。这个程序是使用 `LUA` 写出的，拥有跨平台机能。该程序在读取 `.mx1` 文件之后，输出后缀为 `.mx2` 的文件，包括每行的小节数和 `\elemskip` 的值。

最后，文档需要再次通过  $\text{T}_{\text{E}}\text{X}$  编译，读取 `.mx2` 的值，并输出优化后的乐谱布局。

### 1.3.2 例子

以下是三段编译的例子。在进行第 1 段编译时，`\elemskip` 的值尚未确定，所以 MusiX<sub>TEX</sub> 使用一个默认的值进行输出。在此之后，输出大概会如下所示。

```

i· 6 5 | 6 i 5 6 i | 6 5 6 i | 5 - |

5· 6 i 2 | 6 5 3 | 5 2 3 5 | 1 - |

5 5 6 5 3 | 2 1 2 | 5 5 6 5 3 | 2 1 2 |

5· 6 i 2 | 6 5 3 | 5 2 3 5 | 1 - ||

```

此时，每行的间距都相等，但是行并不对齐。在经过 `musixflx` 并重新编译之后，效

果则如下所示。

$$\begin{array}{ccccccc}
 \dot{1} \cdot \underline{6} \underline{5} & | & \underline{\underline{6 \dot{1} 5 6 \dot{1}}} & | & \underline{6 \ 5 \ 6 \ \dot{1}} & | & 5 \ - \ | \\
 \\ 
 \underline{5 \cdot \underline{6 \dot{1} \dot{2}}} & | & \underline{6 \ 5 \ 3} & | & \underline{5 \ 2 \ 3 \ 5} & | & 1 \ - \ | \\
 \\ 
 \underline{5 \ \underline{5 6} 5 \ 3} & | & \underline{2 \ 1 \ 2} & | & \underline{5 \ \underline{5 6} 5 \ 3} & | & \underline{2 \ 1 \ 2} \ | \\
 \\ 
 \underline{5 \cdot \underline{6 \dot{1} \dot{2}}} & | & \underline{6 \ 5 \ 3} & | & \underline{5 \ 2 \ 3 \ 5} & | & 1 \ - \ ||
 \end{array}$$

此时 MusiX<sub>T</sub>E<sub>X</sub> 已经决定了相应的行数，每行末尾对齐，但是第 1 行的音符间距略窄于第 2 行。本例源代码如下。

```

\hsize=120mm
\leftskip=10mm
\smallmusicsize
\nostartrule
\nobarnumbers
\setclef18
\setlines10
\startpiece
\Notesp\iul0\jnp j\en
\notes\nuul0\tjn0h\en
\Notes\jq g\en\bar
\notes\Qjuul hjgh\en
\Notes\jq j\en\bar
\Notes\Djul hg\Djul hj\en\bar
\Notes\jq{gZ}\en\bar
\Notesp\iul0\jnp g\en
\notes\nuul0\tjn0h\en
\Notes\Djul jk\en\bar
\Notes\Djul hg\en
\Notes\jq e\en\bar
\Notes\Djul gd\Djul eg\en\bar
\Notes\jq{cZ}\en\bar
\Notes\iul0\jn g\en
\notes\nuul0\jn g\tjn0h\en
\Notes\Djul ge\en\bar
\Notes\Djul dc\en
\Notes\jq d\en\bar
\Notes\iul0\jn g\en
\notes\nuul0\jn g\tjn0h\en
\Notes\Djul ge\en\bar
\Notes\Djul dc\en
\Notes\jq d\en\bar
\Notesp\iul0\jnp g\en
\notes\nuul0\tjn0h\en
\Notes\Djul jk\en\bar

```

```

\Notes\Djul hg\en
\Notes\jq e\en\bar
\Notes\Djul gd\Djul eg\en\bar
\Notes\jq{cZ}\en\Endpiece

```

### 1.3.3 手动改变布局

三段编译的一个好处是可以通过少数参数调整乐谱布局，即 `\mulooseness`。这个值和  $\text{T}_\text{E}\text{X}$  自身的 `\looseness` 相当——若这个值为 1，程序会将相应段落变得更松散，以在输出时增加 1 行；若为 -1 则减少 1 行，以此类推。此时，谱表和小节替代了行和段落。“小节”是任何不含有强制断行的部分乐谱。强制断行的命令包括 `\stoppiece`、`\endpiece`、`\zstoppiece`、`\Stoppiece`、`\Endpiece`、`\alaligne`、`\zalaligne`、`\alapage` 或 `\zalapage`。若以上都不存在，则整段乐谱为一个小节。

在小节中的任意<sup>2</sup>位置将 `\mulooseness` 赋为非 0 值，则 MusiX $\text{T}_\text{E}\text{X}$  会将此小节用非默认的行数排版

例如，将上例最后一行改为

```
\Notes\jq{cZ}\en\mulooseness=1\Endpiece
```

会产生如下结果。

```

i· 65 | 6i56i | 6 5 6 i | 5 - |
5· 6i 2 | 6 5 3 | 5 2 3 5 | 1 - |
5 565 3 | 2 1 2 | 5 565 3 | 2 1 2 |
5· 6i 2 | 6 5 3 | 5 2 3 5 | 1 - ||

```

而与之相反，

```
\Notes\jq{cZ}\en\mulooseness=-1\Endpiece
```

会产生如下结果。

```

i· 65 | 6i56i | 6 5 6 i | 5 - |
5· 6i 2 | 6 5 3 | 5 2 3 5 | 1 - |

```

<sup>2</sup>为增强可读性，最好在一小节的首、尾。

5 5 6 5 3 | 2 1 2 | 5 5 6 5 3 | 2 1 2 |

5 · 6 1̇ 2̇ | 6 5 3 | 5 2 3 5 | 1 - ||

在手动键入 MusiX<sub>TEX</sub> 输入文件时（在只使用五线谱时已无太大必要，因为有 PMX 的存在），可以使用如下步骤。

1. 逐音符组（\notes 或相似命令）输入内容，需要使用适当的水平间距命令（见 3.1 节），使得可变宽度和音符长度有相应的对应关系。这部分内容将在第三章详细说明。
2.  $\text{TEX} \implies \text{musixflx} \implies \text{TEX}$ .
3. 观察输出文件，判断是否需要手动调整（例如，排版者可能希望乐谱排满若干页）。
4. 若需要调整，删除 mx2 文件，调整 \mulooseness，重新进行编译。

另一种方法是定义 \linegoal 的值，但是此时 \mulooseness 必须为 0。这两个变量在一小节结束后都会自动清零。

对于长篇乐谱（4 页以上），只有 1 个小节和一个 \mulooseness 值是不明智的，因为此时不仅需要两端对齐，也需要乐谱占整数页。这时，可以通过 \alapage 或 \alaligne<sup>3</sup>。

MusiX<sub>TEX</sub> 将伸缩量完全在外部计算，并在排版时已经确定相应的值，这使得符杠、连音线等元素的长度可以严格定义。

### 1.3.4 musixflx 的使用

多数系统上，musixflx 可以直接通过命令行调用。

musixflx (文件名).mx1

d 将调试信息在屏幕上输出

f 将调试信息输出到 (文件名).mx1

s 将行的计算结果直接输出到屏幕

为了简化输入，程序输入的文件名可以带 mx1、tex 或不带后缀名；程序在这三种情况下都能正常使用

---

<sup>3</sup>PMX 也使用这类命令。

### 1.3.5 致新手：注意空格！

因为 MusiX<sub>TEX</sub> 宏集的特点，打谱时对于空格、空行必须格外小心。新手最常见的错误是在一行中间或结尾多加了一个空格。这类空格（或不通过 MusiX<sub>TEX</sub> 命令正常输入的其他字符）不会被 MusiX<sub>TEX</sub> 记录，但是仍然会被 <sub>TEX</sub> 命令输出。这样的错误会使得乐谱排版时出现 `overfull \hbox` 一类的错误，并且会影响布局。

避免这类错误的最好方法是不输入不必要的空格，并在不以控制序列结束的行末加上 % 去除换行导致的空格。

另一个需要注意的点是，不要在音乐范围内使用 `\hskip` 或 `\kern`，除非在宽度为 0 的盒子中，如 `\rlap`、`\llap`、`\zcharnote`、`\uptext` 等。对于可变的宽度，如 `\noteskip`、`\elemskip`、`\afterruleskip`、`\beforruleskip`<sup>4</sup>等，不要进行赋值。

## 1.4 其他

### 1.4.1 从乐谱中提取声部

见第??章。

### 1.4.2 乐谱及音符的大小

MusiX<sub>TEX</sub> 宏集默认的乐谱大小是 20 pt，但是也可以使用 16（简谱使用）、24、29 pt 的尺寸。每个乐器也能有自己的乐谱尺寸（一般小于默认尺寸）。此外，音符、符杠等都能通过命令改变尺寸。

---

<sup>4</sup>注意，`\hardspace` 不在此列，这是 MusiX<sub>TEX</sub> 宏集特许的加入水平间距的方法。

## 第二章 输入文档的格式

### 2.1 MusiX<sub>TEX</sub> 的基本规定

MusiX<sub>TEX</sub> 的输入文件和一般的 T<sub>EX</sub> 文件类似。使用 MusiX<sub>TEX</sub> 宏包的命令是 `\input musixtex`；相应地，使用简谱宏包 `musixjnp` 的命令是 `\input musixjnp`。宏集内的其他宏包也如此调用。

MusiX<sub>TEX</sub> 宏集默认乐谱最多有 6 个声部。如果需要更多的声部，可以使用 `musixadd` 或 `musixmad` 宏包扩展到 9 或 12 个。如果还需要更多声部，可以使用 `\setmaxinstruments`、`\setmaxgroups`、`\setmaxslurs`、`\setmaxtrills`、`\setmaxoctlines` 等命令手动修改。

### 2.2 常用的设置命令

#### 2.2.1 全局乐谱尺寸

MusiX<sub>TEX</sub> 设置了 4 种乐谱尺寸：小 (16 pt)、中 (20 pt, 默认)、大 (24 pt)、特大 (29 pt)。或需要非默认尺寸，分别使用命令 `\smallmusicsize`、`\largemusicsize`、`\Largemusicsize`。在使用该命令时，所有的音乐内容都会相应缩放。

简谱推荐使用“小”尺寸；线简混排时参考第??章的内容。

#### 2.2.2 乐器个数

命令 `\instrumentnumber n` 表示乐谱中乐器的个数，默认为 1。

#### 2.2.3 谱号

各谱表的默认谱号是高音谱号。修改谱号的命令为 `\setclef{n}{s1s2s3s4...}`。 $n$  是乐器从下往上数的序号， $s_1$ 、 $s_2$  等是该乐器从下往上的谱表序号对应的谱号。

简谱应当使用 8（没有谱号）。

### 2.2.4 节拍

所有声部通用的节拍记号可以使用 `\generalmeter{m}` 命令。 $m$  是拍号相应的命令，简谱使用 `\inijfrac{a}{b}`（乐谱首）或 `\jfrac{a}{b}`（乐谱中切换拍号）， $a$  为上方的数， $b$  为下方的数。

若使某乐器的节拍与全局设置不同，则应该使用 `\setmeter{n}{\{m_1\}\{m_2\}\{m_3\}\{m_4\}}`，与谱表相似。

### 2.2.5 乐器名

乐器名使用 `\setname{n}{乐器名}` 命令设置，程序会自动将乐器名称居中放置在 `\parindent` 宽度的范围内。

### 2.2.6 乐器组

默认情况下，乐器间没有连接。若需要排合唱谱（如上一章中的《喀秋莎》例谱），则需要方括号分组。插入方括号的命令是

```
\songtop{n}
\songbottom{m}
```

若需要多个方括号，则使用类似的 `\grouptop` 和 `\groupbottom` 命令。

## 2.3 纯简谱谱表常用的设置

除上述内容以外，简谱谱表还经常需要使用 `\nostartrules`（取消每行左侧的连线）和 `\sepbarrules`（每个乐器的小节线分开）两条命令。

## 第三章 输入音符的预备工作

### 3.1 谱表的水平间距

#### 3.1.1 音符间距

Mu<sub>si</sub>X<sub>T</sub><sub>E</sub>X 提供了一系列设置音符间水平间距的命令,其默认意义如表 3.1 所示。

命令	间距		建议的时值
	(等差)	(等比)	
<code>\znotes ... &amp; ... &amp; ... \en</code>	0	0	特殊情况
<code>\notes ... &amp; ... &amp; ... \en</code>	2.00	2.00	<u>5</u> 16 分
<code>\notesp ... &amp; ... &amp; ... \en</code>	2.50	2.38	<u>5</u> · 附点 16 分、三连音 8 分
<code>\Notes ... &amp; ... &amp; ... \en</code>	3.00	2.83	<u>5</u> 8 分
<code>\Notesp ... &amp; ... &amp; ... \en</code>	3.50	3.36	<u>5</u> · 附点 8 分、三连音 4 分
<code>\NNotes ... &amp; ... &amp; ... \en</code>	4.00	4.00	<b>5</b> 4 分
<code>\NNotesp ... &amp; ... &amp; ... \en</code>	4.50	4.76	<b>5</b> · 附点 4 分、三连音 2 分

表 3.1: 水平间距命令