

# SumatraPDF 阅读器的配置方法<sup>1</sup>

啸行<sup>2</sup>

2021 年 4 月 26 日

<sup>1</sup><https://github.com/OsbertWang/latex-editor-sumatrapdf>

<sup>2</sup>[ranwang.osbert@outlook.com](mailto:ranwang.osbert@outlook.com)

## 摘要

SumatraPDF 是一款常见的、功能比较丰富的、适合浏览 L<sup>A</sup>T<sub>E</sub>X 编译结果的轻量级 PDF 阅读器。以前 C<sub>T</sub><sub>E</sub>X 套装将它集成以配合 WinEdt 使用, 现如今依然有很多 WinEdt 用户青睐于这样的搭配, [SumatraPDF 官网](#)也给出了 [Use Sumatra as a pre-viewer for LaTeX editors](#) 的手册, 其中就包含了 TeXnicCenter、WinEdt、Emacs 和 Vim。本文将整合 SumatraPDF 阅读器该怎么设置——以 TeXStudio 和 VS Code 为例和以 Notepad++ 为 LaTeX 编辑器以及以 SumatraPDF 进行正反向搜索的配置方法两篇文章。

全文假设 main.tex 已经被编译为 main.pdf, 其中 main.tex 内容如下:

```
1 \documentclass{article}
2 \usepackage{lipsum}
3 \begin{document}
4 \section{main file}
5 \lipsum[1]
6 \input{subfile.tex}
7 \end{document}
```

而 subfile.tex 内容如下:

```
1 \section{subfile}
2 \lipsum[2]
3 \[ a = b \]
```

# 1 TeXStudio

假设 `<SumatraPDFROOT>` 是 SumatraPDF 的安装路径. 在配置之前, 我首先简要介绍下使用命令行的方法.

事实上, SumatraPDF 允许通过添加命令行参数来形成不同的阅读效果. 对 L<sup>A</sup>T<sub>E</sub>X 用户而言, 最重要的效果就是正反向搜索. 在这里给出我所使用的命令行代码.

```
1 <SumatraPDFROOT>\SumatraPDF.exe -reuse-instance -forward-search subfile.tex 2 -  
inverse-search "\"<TeXStudioROOT>\texstudio.exe\" \"%f\" -line %1" main.pdf
```

其中,

```
1 -reuse-instance
```

表示只打开一个 SumatraPDF 窗口;

```
1 -forward-search subfile.tex 2
```

表示对 `subfile.tex` 的第 2 行进行正向搜索 (在 SumatraPDF 中表现为 `main.pdf` 对应位置变蓝);

```
1 -inverse-search "\"<TeXStudioROOT>\texstudio.exe\" \"%f\" -line %1" main.pdf
```

表示使用 TeXStudio 对 `main.pdf` 进行反向搜索 (表现为双击 `main.pdf` 的某部分会在 TeXStudio 中返回对应的 `tex` 文件的某行).

明白了命令行的原理就很容易在软件中进行设置了. 首先在 `Options` `Configure TeXstudio` 中选择 `Show Advanced Options`, 然后在 `Build` `User commands` 中添加

```
1 "<SumatraPDFDIR>\SumatraPDF.exe" -reuse-instance -forward-search ?c:rme" @ -inverse-  
search "<TeXstudioDIR>\texstudio.exe %f -line %1" ?p{pdf}:ame"
```

将其命名为 `user0:sumatrapdf`. 这里的 `?c:rme`、`@`、`%`、`?p{pdf}:ame` 均是 TeXStudio 自己定义的有特殊含义的字符. 接下来, 在 `Build` `Build & View` 中将 `txs:///compile | txs:///view` 改为 `txs:///compile | txs:///user0`. 最后在 `Menus` 中将 `View` 的命令由 `txs:///view` 改为 `txs:///user0`. 完成以上设置后, 关闭窗口. 这时, 用户使用快捷键 `F5` 和 `F7` 均可打开 SumatraPDF 并且实现了正反向搜索.

如果以上设置没能正确实现正反向搜索, 那么可以将 `user0:sumatrapdf` 改为

```
1 dde:/// "<SumatraPDFDIR>\SumatraPDF.exe":SUMATRA/control/[ForwardSearch(?p{pdf}:ame",?  
c:rme",@,0,0,1)]
```

其他不变. 而后用 SumatraPDF 打开编译完的 pdf 文件, 将

```
1 "<TeXstudioROOT>\texstudio.exe" "%f" -line %1
```

写入 `Settings` `Options` `Set inverse search command line`. 至此完成了逆向搜索, 双击 pdf 文件便可回到 `TeXStudio` 中对应代码的行首.

## 2 VS Code

同样先介绍命令行

```
1 <SumatraPDFROOT>\SumatraPDF.exe -reuse-instance -forward-search main.tex 4 -inverse-  
search "\"<VSCodeROOT>\bin\code.cmd\" -r -g \"%f:%l\"\" main.pdf
```

其中 <VSCodeROOT> 是 VS Code 的安装路径, code.cmd 是软件已经写好的打开 VS Code 的批处理文件. 与前面不同的是 -inverse-search 参数对应了不同的编辑器.

接下来, 用户安装 LaTeX-Workshop 插件. 然后打开设置的 json 文件, 在其中添加以下代码:

```
1 "latex-workshop.view.pdf.viewer": "external",  
2 "latex-workshop.view.pdf.ref.viewer": "external",  
3 "latex-workshop.view.pdf.external.viewer.command": "<SumatraPDFROOT>/SumatraPDF.exe",  
4 "latex-workshop.view.pdf.external.viewer.args": [  
5     "-inverse-search",  
6     "\"<VSCodeROOT>/bin/code.cmd\" -r -g \"%f:%l\"",  
7     "%PDF%"  
8 ],  
9 "latex-workshop.view.pdf.external.synctex.command": "<SumatraPDFROOT>/SumatraPDF.exe",  
10 "latex-workshop.view.pdf.external.synctex.args": [  
11     "-forward-search",  
12     "%TEX%",  
13     "%LINE%",  
14     "%PDF%",  
15 ],
```

注意在 json 文件中要将路径中的 \ 写作 /. 下面略加解释以上设置的用意.

为了能够使用 SumatraPDF 进行正反向搜索, 首先需要将

```
1 "latex-workshop.view.pdf.viewer"
```

和

```
1 "latex-workshop.view.pdf.ref.viewer"
```

都设置为外部 (external). 然后在

```
1 "latex-workshop.view.pdf.external.viewer.command"
```

和

```
1 "latex-workshop.view.pdf.external.viewer.args"
```

中分别设置打开 pdf 阅读器的命令和所需参数, 这些参数用于进行反向搜索. 类似地, 正面搜索的命令和参数分别定义在

```
1 "latex-workshop.view.pdf.external.synctex.command"
```

和

```
1 "latex-workshop.view.pdf.external.synctex.args"
```

完成以上设置, 用户使用时在左侧点击 `View LaTeX PDF` 便可打开 SumatraPDF, 之后点击 `navigate, select and edit` `SyncTeX from cursor` 便可进行正向搜索. 当然, VS Code 还提供了快捷键供用户使用, 用户可以自行查询或更改为自己喜欢的按键组合.

## 3 Notepad++

Notepad++ 配合 SumatraPDF 阅读器需要一些插件, 例如 NppExec 和 Customize Toolbar, 它们的安装方式很简单, 在 **插件** >> **插件管理** 中即可找到.

首先用 Notepad++ 打开一个 tex 文件. 通过快捷键 **F6** 调出 Execute 对话框. 在窗口内输入如下代码:

```
1 cd $(CURRENT_DIRECTORY)
2 xelatex -synctex=1 -interaction=nonstopmode $(NAME_PART)
```

简要介绍一下代码含义. `$(CURRENT_DIRECTORY)` 是 NppExec 插件的设置变量, 表示当前文件所在的目录; `$(NAME_PART)` 表示当前文件文件名 (不含扩展名). 这两句代码的含义就很好理解了, 进入当前目录, 对当前文件用 `xelatex` 以指定参数进行编译. 将这段临时脚本起名保存为 `XeLaTeX`, 点击 **OK** 按钮即开始这段脚本. 类似地, 我们可以创建其他编译命令的执行脚本, 不一一赘述.

为了实现 SumatraPDF 的正反向搜索, 预览 pdf 文件的脚本应输入如下代码:

```
1 cd $(CURRENT_DIRECTORY)
2 NPP_run "<SumatraPDFROOT>\SumatraPDF.exe" -reuse-instance -forward-search $(NAME_PART)
  .tex $(CURRENT_LINE) -inverse-search "\"<Notepad++ROOT>\Notepad++.exe\" -n%l %f"
  $(NAME_PART).pdf
```

其中, `Npp_run` 是 NppExec 附带的命令, 表示执行外部程序和命令; `$(CURRENT_LINE)` 表示文件的当前行. 若要形成编译链, 连续使用几个既已保存的脚本, 可以不用将脚本命令重复复制, 可以使用 NppExec 附带命令 `Npp_exec` 执行相应脚本即可. 例如, 已经设置保存了 `XeLaTeX`、`bibtex` 和 `View PDF` 三个脚本, 若需创建一个标准的完整编译过程的脚本, 可以在 Execute 窗口输入如下代码并保存即可

```
1 Npp_exec "XeLaTeX"
2 Npp_exec "bibtex"
3 Npp_exec "XeLaTeX"
4 Npp_exec "XeLaTeX"
5 Npp_exec "View PDF"
```

利用 **插件** >> **NppExec** >> **Advanced Options** 可以进行一些高级设置, 将已经保存的脚本加入到菜单项中. **Menu item** 可以选择需要添加到菜单的脚本, 并为菜单项设置名称; **Menu item\*** 中勾选 **Place to the Macros submenu** 后, 菜单项将添加在 **宏 (M)** 菜单项之下.

在 **宏 (M)** >> **管理快捷键** 可以设置快捷键. 在 **插件命令** 选项卡可以找到添加到菜单项的命令, 利用对话框设置快捷键即可.

快捷键 **Shift** + **F6** 可以调出 NppExec 控制台输出过滤. 在 **HighLight** 选项卡中, 我们可以设置控制台信息过滤条件 `1.%LINE%`, 并设置高亮颜色和粗体下划线样式. 在勾选最前

面的 ，启用这个条件后，我们可以编辑一个简单的 tex 文件：

```
1 \documentclass{article}
2 \begin{document}
3 Hello! \latex
4 \end{document}
```

我们故意写错  $\LaTeX$  为  $\latex$ ，进行编译后，系统将给出报错信息，控制台回显的报错信息中：

```
1 1.4 Hello! \latex
```

双击这条报错信息，在文本编辑器内，光标将定位到出错这一行的行首。这个设置实现了错误快速定位。

Customize Toolbar 安装完成后，**插件**》**Customize Toolbar**》**Custom Buttons** 可以启动用户自定义工具条按钮。按钮的配置是通过文件

```
1 %APPDATA%\Notepad++\plugins\config\CustomizeToolbar.btn
```

实现的。文件示例如下：

```
1 宏(M),pdfLaTeX,,pdflatex.bmp
2 宏(M),XeLaTeX,,xelatex.bmp
3 宏(M),bibtex,,bibtex.bmp
4 宏(M),biber,,biber.bmp
5 宏(M),makeindex,,makeindex.bmp
6 宏(M),View PDF,,adobe.bmp
7 宏(M),Clear,,trash.bmp
8 宏(M),Latexmk,,latexmk.bmp
```

其中，每一行均为 4 个逗号，分隔成的 5 部分，前 4 部分为各级菜单项名称，必须与当前菜单选用的界面语言一致，最后一项为按钮显示图标文件名，图标文件必须为  $16 \times 16$  的 bmp 格式图像。特别需要注意的是：CustomizeToolbar.btn 的文件编码格式必须保存为 **UCS-2 Little Endian 含 BOM**，文件行尾格式必须为 **Windows (CR LF)**。