

L^AT_EX 编译环境配置：Visual Studio Code 配置简介

Dongsheng Deng

February 2, 2019

本文介绍如何配置 Visual Studio Code 作为 L^AT_EX 的编辑器。Github 地址：[vscode-latex](https://github.com/dongshengdeng/vscode-latex)，欢迎提交 issues 和 pull requests。

1 为什么用 Visual Studio Code

Visual Studio Code（以下简称 VS Code）是微软推出的一个编辑器，它的优点你可以自行百度，这里不赘述。对我来说，它最有吸引力的当属在 Windows 系统，它对于中英文字体的渲染。如果你原来用过其他编辑器，你就知道在普通屏幕上，中英文的显示效果简直是灾难。我原来因为编辑器的中文显示（当然还有 Terminal 的吸引力）一度想买 Mac，当然最后因为对性能和颜值的追求并不匹配我的财力，加上 Windows 上有些软件不能舍弃，最后作罢。

注：高分屏加上合适的字体，Sublime Text 的显示效果也非常好。

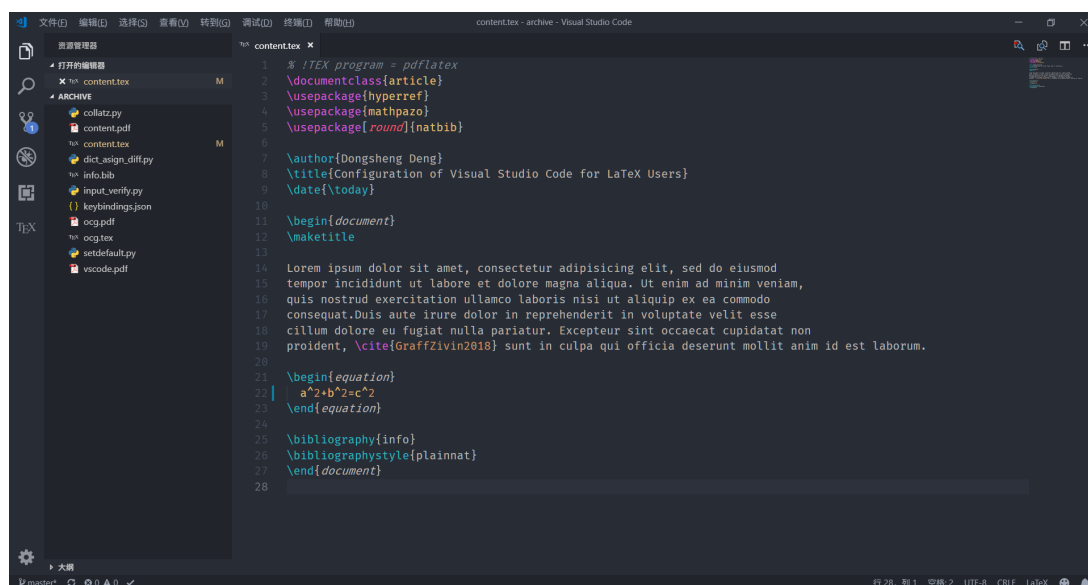


图 1: Visual Studio Code 的界面图

2 准备工作

首先，为了搭建 L^AT_EX 工作环境，你需要安装：

- T_EX Live 或者 MiKTeX（本文以 T_EX Live 2018 为例）
- Visual Studio Code
- L^AT_EX Workshop（VS Code 插件）
- SumatraPDF 阅读器（可选，用于预览 PDF）

在上述软件/插件安装之后,你需要把 \TeX Live 的 bin 目录(`D:/Program Files/texlive/2018/bin/win32`) 以及 SumatraPDF 的路径 (`C:/Program Files (x86)/SumatraPDF`) 添加到系统环境变量 (Path) 中。

2.1 安装插件

VS Code 中插件安装方法如下: 在左侧点击扩展按钮 (Key: `Ctrl+Shift+X`), 然后搜索插件名字 LaTeX Workshop, 选择安装即可。

2.2 添加环境变量


Win10 中将路径添加到环境变量中的步骤如下: 右键我的电脑, 然后选择 属性, 在左侧选择高级系统设置, 然后选择下方的 环境变量, 选择变量 Path 编辑, 将需要添加的路径添加进去即可。

3 配置编译方式与编译组合

VS Code 在 2018 年经历了一次大改之后, 配置比原来简单了。它们把过去的 `tool.chain` 改为了 `recipe`, 其实本质上是一样的。

3.1 编译方式 (tool)

VS Code 默认添加了 3 个编译工具 (tools): 分别是 `latexmk`, `pdflatex` 和 `bibtex` (所有的工具只编译一次)。编译 `tex` 文档方法, 使用右键, 选择 Build LaTeX Project (快捷键: `Ctrl+Alt+B`), 默认使用 `latexmk`, 查看 PDF 文件使用快捷键: `Ctrl+Alt+V`。

为了添加其他的编译方式 (比如 `xelatex`), 我们需要修改 LaTeX Workshop 的配置。打开 LaTeX Workshop 配置的方法如下: 在 VS Code 界面左下角, 点击齿轮按钮 , 选择设置, 然后在设置搜索框内输入 `latex`, 在搜索结果中, 点击 在 `settings.json` 中编辑, 示例如图 2:

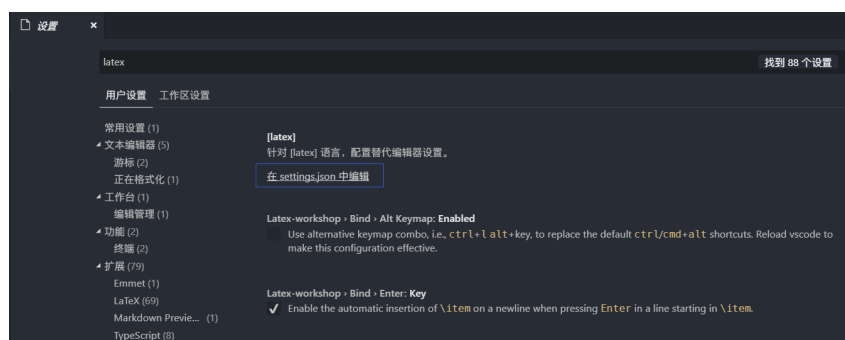


图 2: 打开设置文件

打开配置文件之后, 在右侧 用户设置 粘贴下面 JSON 片段:

```
"latex-workshop.latex.tools": [
  {
    "name": "xelatex",
    "command": "xelatex",
    "args": [
      "-synctex=1",
      "-interaction=nonstopmode",
      "-file-line-error",
      "%DOC%"
    ]
  }
]
```


```

},
{
  "name": "xelatex-with-shell-escape",
  "command": "xelatex",
  "args": [
    "--shell-escape",
    "-synctex=1",
    "-interaction=nonstopmode",
    "-file-line-error",
    "%DOC%"
  ]
},
{
  "name": "pdflatex",
  "command": "pdflatex",
  "args": [
    "-synctex=1",
    "-interaction=nonstopmode",
    "-file-line-error",
    "%DOC%"
  ]
},
{
  "name": "pdflatex-with-shell-escape",
  "command": "pdflatex",
  "args": [
    "--shell-escape",
    "-synctex=1",
    "-interaction=nonstopmode",
    "-file-line-error",
    "%DOC%"
  ]
},
{
  "name": "latexmk",
  "command": "latexmk",
  "args": [
    "-synctex=1",
    "-interaction=nonstopmode",
    "-file-line-error",
    "-pdf",
    "%DOC%"
  ]
},
{
  "name": "bibtex",
  "command": "bibtex",
  "args": [
    "%DOCFILE%"
  ]
}
],

```

注意：虽然左侧插件默认添加了编译方式（`pdflatex` 与 `bibtex`），也必须将其编译方式的设置（比如 `pdflatex` 等）添加到右侧用户设置中。另外上面我们分别为 `pdflatex` 和 `xelatex` 添加了 `--shell-escape` 参数，一个典型的应用场景就是编译包含 `minted` 宏包的文件时。

3.2 编译组合 (recipe)

如果我们要对一个文档/项目完整的编译（比如 `pdflatex -> bibtex -> pdflatex -> pdflatex`）我们需要用到编译组合（`recipes`）。LaTeX Workshop 默认添加了两个 `recipes`，分别是 `latexmk` 和 `pdflatex -> bibtex -> pdflatex*2`，可以通过点击左侧新增的 `TeX` 按钮 ，然后点击 `Build LaTeX project`，选择适合的编译组合。

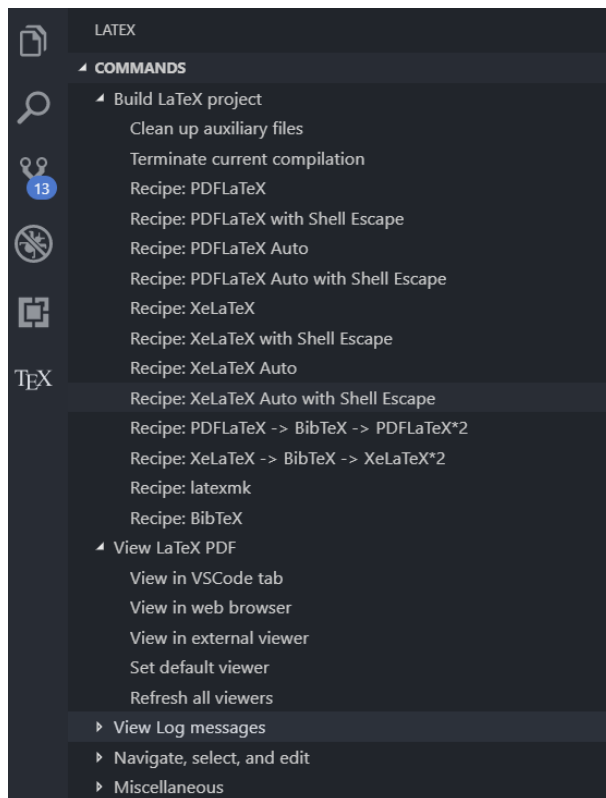


图 3: 编译组合的选择

我们之前添加了 `xelatex` 编译方式，我们这里配置下 `xelatex` 的完整编译链 `xelatex -> bibtex -> xelatex*2`，另外补充单次编译的 `recipes`。方法和之前类似，打开用户配置文件，将如下 JSON 添加到用户配置中即可。添加新的编译组合之后需要重启 VS Code 才能在 `TeX` 按钮下看到。

```
"latex-workshop.latex.recipes": [
  {
    "name": "PDFLaTeX",
    "tools": [
      "pdflatex"
    ]
  },
  {
    "name": "PDFLaTeX with Shell Escape",
    "tools": [
      "pdflatex-with-shell-escape"
    ]
  },
  {
    "name": "PDFLaTeX Auto",
    "tools": [
      "pdflatex-latexmk"
    ]
  }
]
```

```

},
{
  "name": "XeLaTeX",
  "tools": [
    "xelatex"
  ]
},
{
  "name": "XeLaTeX with Shell Escape",
  "tools": [
    "xelatex-with-shell-escape"
  ]
},
{
  "name": "XeLaTeX Auto",
  "tools": [
    "xelatex-latexmk"
  ]
},
{
  "name": "PDFLaTeX -> BibTeX -> PDFLaTeX*2",
  "tools": [
    "pdflatex",
    "bibtex",
    "pdflatex",
    "pdflatex"
  ]
},
{
  "name": "XeLaTeX -> BibTeX -> XeLaTeX*2",
  "tools": [
    "xelatex",
    "bibtex",
    "xelatex",
    "xelatex"
  ]
},
{
  "name": "latexmk",
  "tools": [
    "latexmk"
  ]
},
{
  "name": "BibTeX",
  "tools": [
    "bibtex"
  ]
},
],

```

`settings` 目录下提供了一个比较完整的配置文件 [settings.json](#), 包括了 `latexmk`¹ 与 `pdflatex/xelatex` 以及 `--shell-escape` 的搭配使用。需要的可以下载替换掉用户配置 `tools` 和 `recipe` 部分。另外 `example` 目录下提供一个测试完整编译方式的代码 `tex`, `bib`, `pdf`, 你可以用来测试能否编译。效果图如图 4:

¹参考了嘯行的《编辑器 `texstudio` 和 `vscode` 使用总结》

Configuration of Visual Studio Code for L^AT_EX Users

Dongsheng Deng

February 2, 2019

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, [Graff Zivin et al. \(2018\)](#) sunt in culpa qui officia deserunt mollit anim id est laborum.

$$a^2 + b^2 = c^2 \quad (1)$$

References

Joshua Graff Zivin, Solomon M Hsiang, and Matthew Neidell. Temperature and human capital in the short and long run. *Journal of the Association of Environmental and Resource Economists*, 5(1):77–105, 2018. ISSN 2333-5955.

图 4: 编译效果图

3.3 指定编译方式

在 Sublime Text 或者 T_EX Studio 中, 可以在文件的首行指定编译方式 (`% !TEX program`) 以及主文档 (`% !TEX root`), L^AT_EX Workshop 也把这个功能添加到了其中, 使用方法完全一样。 `% !TEX program` 和 `% !TEX root` 被称为 Magic Command (魔法注释)。

之前的魔法注释还不支持指定参数, 现在已经可以了。

首先可以给它的配置文件里面先加上默认的魔法注释参数, 如下:

```
"latex-workshop.latex.magic.args": [
  "-synctex=1",
  "-interaction=nonstopmode",
  "-file-line-error",
  "%DOC%"
]
"latex-workshop.latex.magic.bib.args": [
  "%DOCFILE%"
],
```

这时候在主文件开头加上魔法注释 `% !TEX program = xelatex` 运行时相当于

`xelatex.exe -synctex=1 -interaction=nonstopmode -file-line-error "yourfile".tex`。

如果你还要用 `bibtex` 编译参考文献的话在上一个魔法注释下面再加一个魔法注释:

`% !BIB program = bibtex`, 如果用的是 `biber`, 后面的参数可以改成 `biber`。

但是你可能想让不同的文档编译时可以带不同的参数, 这时候还可以再加一个魔法注释:

`% !TEX options = -synctex=1 -interaction=nonstopmode -file-line-error "%DOC%"`, 等号后面的参数可以自己按需求修改。

看到这里 `texify` 和 `latexmk` 用户觉得比较实用了。不需要修改配置文件, 直接用魔法注释就可以实现你想要的编译。

下面来说明这两种的魔法注释怎么写。

对于 `texify`: 想用 `xelatex` 的话这么配置魔法注释

```
% !TEX program = texify
% !TEX options = --synctex --pdf --engine=xetex --tex-option="\-interaction=nonstopmode\"
↪ --tex-option="\-file-line-error\" "%DOC%.tex"
```

只用普通的 pdflatex 的话用下面这个魔法注释

```
% !TEX program = texify
% !TEX options = --synctex --pdf --tex-option="\-interaction=nonstopmode\"
↪ --tex-option="\-file-line-error\" "%DOC%.tex"
```

对于 latexmk 用户也是类似的,但是目前有点奇怪如果你使用了 "`latex-workshop.latex.outDir`": "`%DOC%`", 这种指定文件路径的参数,魔法注释明明指定了要生成 pdf,可是它最终只编译出 dvi,可能 latexmk 用户这个魔法注释还是不能非常愉快的玩耍。所以不要去改变辅助文件的存放位置比较好。

指定用 pdflatex 编译

```
% !TEX program = latexmk
% !TEX options = -synctex=1 -interaction=nonstopmode -file-line-error -pdf "%DOC%"
```

指定用 xelatex 编译

```
% !TEX program = latexmk
% !TEX options = -synctex=1 -interaction=nonstopmode -file-line-error -xelatex "%DOC%"
```

下面给一个 MWE:

```
% !TEX program = texify
% !TEX options = --synctex --pdf --tex-option="\-interaction=nonstopmode\"
↪ --tex-option="\-file-line-error\" "%DOC%.tex"
% 魔法注释加在上面,替换称你想用的就行了。记得下面两行选择相应的一行使用。
\documentclass[UTF8]{ctexart}% 测试 pdflatex 时用这个,把下面那句注释掉。
%\documentclass{ctexart}% 测试 xelatex 时用这个,把上面那句注释掉。

\title{A Silly Article}
\author{Me}

\begin{document}
\maketitle

My favorite article is \cite{supercurves}.
我最喜欢的文章是\cite{supercurves}。
\bibliographystyle{plain}
\bibliography{shortbib} % 还需要建立一个 shortbib.bib 文件和这个文件放一起。
\end{document}
```

```
% 这是 shortbib.bib 文件
@ARTICLE{supercurves,
AUTHOR={Minghui Xia},
TITLE={Image Registration by "Super-Curves"},
JOURNAL={IEEE Transactions on Image Processing},
YEAR={1985},
VOLUME = {13},
PAGES={720},
}
```

将上述代码保存,然后使用快捷键 `Ctrl+Alt+B`,系统会自动选择相应的编译方式。如果没有其他问题,就能正常编译。

4 其他配置

4.1 配置快捷键

L^AT_EX Workshop 的快捷键并不友好，我们可以自定义快捷键，方法如下：点击 VS Code 左下角的齿轮（设置），选择 键盘快捷方式。

- 搜索 latex build project，将默认的快捷方式改为 Ctrl+B
- 搜索 latex build with recipe，将其改为 Ctrl+R
- 搜索 latex view pdf file，将其改为 Ctrl+L
- 你还可以补充其他快捷键。

配置好快捷键之后，之后当你指定了编译方式时可以直接使用快捷键 Ctrl+B 编译一次文档。当你需要完整编译整个文档（文献，目录等），使用快捷键 Ctrl+R，选择完整的编译方案即可。是不是方便多了？

补充：为了方便快捷键的绑定，这里直接提供设置文件（keybindings.json）的 [下载](#)，下载之后只需要将其替换用户的快捷键设置即可（默认位置为：C:/Users/<Your User Name>/AppData/Roaming/Code/User）。

4.2 配置阅读器以及自动编译

还有其他几个设置需要提一下，由于笔记本的屏幕很小，我并不习惯使用 VS Code 自带的 PDF 阅读器作为预览的阅读器，可以设置 SumatraPDF 作为 PDF 阅读器。另外，自动编译选项我也选择关闭。

```
"latex-workshop.view.pdf.viewer": "external",  
"latex-workshop.latex.autoBuild.onSave.enabled": false,
```

4.3 Mac 系统 PDF 阅读器配置

如何让 VS Code 在 Mac 上与诸如 Skim 的外部 PDF 浏览器配合工作——特别是 L^AT_EX 的正反跳转，请参考小 L 的 [配置说明](#)。

5 texify 和 latexmk 部分的配置文件

如果你想用配置文件的话，下面我提供texify和latexmk的配置。

```
"latex-workshop.latex.tools": [  
  {  
    "name": "texifyxe",  
    "command": "texify",  
    "args": [  
      "--synctex",  
      "--pdf",  
      "--engine=xetex",  
      "--tex-option=\"-interaction=nonstopmode\"",  
      "--tex-option=\"-file-line-error\"",  
      "%DOC%.tex"  
    ],  
    "env": {}  
  },  
  {  
    "name": "texifypdf",  
    "command": "texify",  
    "args": [  
      "--synctex",
```



```

        "--pdf",
        "--tex-option=\"-interaction=nonstopmode\"",
        "--tex-option=\"-file-line-error\"",
        "%DOC%.tex"
    ],
    "env": {}
},
{
    "name": "latexmkxe",
    "command": "latexmk",
    "args": [
        "-synctex=1",
        "-interaction=nonstopmode",
        "-file-line-error",
        "-xelatex",
        //"-outdir=%OUTDIR%",
        "%DOC%.tex"
    ]
},
{
    "name": "latexmkpdf",
    "command": "latexmk",
    "args": [
        "-synctex=1",
        "-interaction=nonstopmode",
        "-file-line-error",
        "-pdf",
        //"-outdir=%OUTDIR%",
        "%DOC%.tex"
    ]
},
],

"latex-workshop.latex.recipes": [
    {
        "name": "texifyxe",
        "tools": ["texifyxe"]
    },
    {
        "name": "texifypdf",
        "tools": ["texifypdf"]
    },
    {
        "name": "latexmkxe",
        "tools": ["latexmkxe"]
    },
    {
        "name": "latexmkpdf",
        "tools": ["latexmkpdf"]
    },
],

"latex-workshop.latex.magic.args": [
    "-synctex=1",
    "-interaction=nonstopmode",
    "-file-line-error",
    "%DOC%"
]

```

```
],  
  "latex-workshop.latex.magic.bib.args": [  
    "%DOCFILE%"  
  ],  
  
  //"latex-workshop.latex.outDir":"%DOC%",
```

参考资料

- 本文最早的版本 [LaTeX 技巧 916: Visual Studio Code 搭建 LaTeX 编写环境 \(已被删\)](#) .
- 更新版本为 [LaTeX 技巧 932: 如何配置 Visual Studio Code 作为 LaTeX 编辑器 \[新版更新\]](#)
- [Github: LaTeX-Workshop](#)
- [How to turn off matching highlighting](#)