Developer's Guide on SJTUBeamer MIN

Log Creative

1.0 August 13, 2021

Contents

1 Preface

SJTUBeamer [MIN] is a presentation template based on beamer package in LaTeX, to fulfill the enthusiasm of those SJTU users to present their content nicely, benefiting from the technology of TeX typesetting engine.

This is a Developer's Guide on SJTUBeamer [MIN] . The document is written in English because the operation in this guidance could be dangerous. Be careful when playing with those macros.

SJTUBeamer [MIN] — the minimal work set of SJTU VI

[MIN] - *minimal*:	minimal work set of SJTU VI.
[MIN] - *minimalism*:	designed in the style of minimalism.
[MIN] - *minimum*:	minimum shapes to show your content.

2 Compliation

Most problems come from LaTeX compilation. The required packages are in the following list.

pgfplots	tikz	xcolor
pgfplotstable	sansmath	tcolorbox
ctex	biblatex	beamer

The detailed description is documented below.

2.1 MiKTeX

All required packages will be automatically installed if you are using MiKTeX[**?**]. And if you want to use the `latexmk` command, please install Perl[**?**] first. And the compilation command for SJTUBeamer [MIN] is as follows:

```
latexmk -pdf main -interaction=nonstopmode
```

2.2 TeX Live

Since some packages are not default installed in the full release of TeX Live, you have to install the packages manually.

On Ubuntu, you could install `pgf` and `xcolor` and other drawing packages through the following command[**?**]:

```
sudo apt install texlive-pictures
```

To typeset Chinese characters, you would better use `CJKutf8` package (in SJTUBeamer [MIN], set `[cjk=true]`), since it is compatible with all platforms and multiple language support. Surround `CJK` environment to make it work and remember to move all the Unicode characters in the permeable to the `CJK` environment[**?**]:

```
\begin{document}
\begin{CJK}{UTF8}{gbsn}
    \institute[]{}
    \title{}
    \subtitle{}
    \author{}
    \date{}
    % your content here ...
\end{CJK}
\end{document}
```

However, if you stick into `ctex`, you can install through `tlmgr`. If that works, then we call it a day.

```
sudo tlmgr install ctex
```

Sometimes, you installed an old TEX Live, and you have to upgrade the `tlmgr` for the new version. And the process could be very buggy, since the following warning may be shown:

```
unexpected return value from verify_checksum: -5
```

and to upgrade the `tlmgr` is painful on Ubuntu. You should add the following content to `/etc/profile/`, which will add the newest path when the system is booting up[**?**]:

```
export PATH=/usr/local/texlive/2021/bin/x86_64-linux:
/usr/local/texlive/:$PATH
```

Reboot your computer if necessary. Then the compile system will be moved to the newer version of TEX Live. Try to install the corresponding packages through the GUI interface of `tlmgr`:

```
sudo tlmgr update --self
sudo tlmgr gui
```

And if you encountered that

```
Critical Package ctex Error: CTeX fontset 'fandol' is
unavailable in current(ctex) mode.
```

You have to modify your compiling program from pdfLaTeX to XeLaTeX by adding the following magic command on the first line:

```
% !TeX TS-program = xelatex
```

2.3 Boost Up

However, it has been tested that the compilation on SJTUBeamer [MIN] is slow. Since the complex patterns have to be rendered in vector shapes and the bibliography requires multiple times of compilation, the time could be wasted on some repetitive works.

This scenario could be improved by enable `[pattern=none]` option on SJTUBeamer [MIN] and enable `[draft]` option on beamer. The former one will disable all the pattern rendering, and the latter one will ignore all the TOC (table of contents) generating.

The project has been implanted to Overleaf. Here is the link [**?**]. And to make that works, the compilation on TeX Live 2021 has to be implemented. And it is discovered that setting the document information outside the `document` environment will cause a significantly longer compiling time, which may be caused by some improper settings in CTeX package. The workaround of that is to follow the setup mentioned in `CJK` settings: put that info into the body of document[**?**].

Currently, CI is available on Github Actions by compiling on LuaLaTeX. SJTUBeamer [MIN] uses `xu-cheng/latex-action@v2` for the compilation docker [**?**] and relocates the compiling folder to `src/`. After compiling, output the PDF artifact. See `.github/workflows/main.yml` for details.

At the same time, AutoBeamer[**?**] is making its own effort on generating beamer code automatically by some replacing strategies. You could preview your beamer code through conversion on Markdown or the article LaTeX code.

Furthermore, there is space for boosting up the beamer compilation time by making use of multi-core processors. Since it is a frame-based document, and the connection between each frame is loose (only some page numbers and citations need to be calculated), the multi-threaded compilation is possible for the beamer

class. You can glimpse the multi-threaded processing for LaTeX from the package animate. In fact, the author created some batch compiling work[**?**] together with the `-Parallel` parameter in PowerShell 7 to make full use of the concurrent computer architecture.

3 Modular Architecture

By the recommendation from beamer package[**?**], SJTUBeamer MIN uses the same modular architecture to build the template. Like it is in Java, to let the beamer template locate your theme, the style file has to be in the standard name.

`.sty File`	**Description**
`beamercolorthemeSJTUBeamermin.sty`	Define global color schemes.
`beamerfontthemeSJTUBeamermin.sty`	Set the font format.
`beamerinnerthemeSJTUBeamermin.sty`	Specifies all parts inside a frame.
`beameroutherthemeSJTUBeamermin.sty`	The frame header and bottom bar.
`beamerthemeSJTUBeamermin.sty`	Entry point of the theme.

Notice that there are some dependencies (logo files) in the `vi/`. Copying the `vi` folder is necessary. Or you could define the location of the logo file by giving `\logo{\includegraphics{logo.pdf}}`.

3.1 Theme

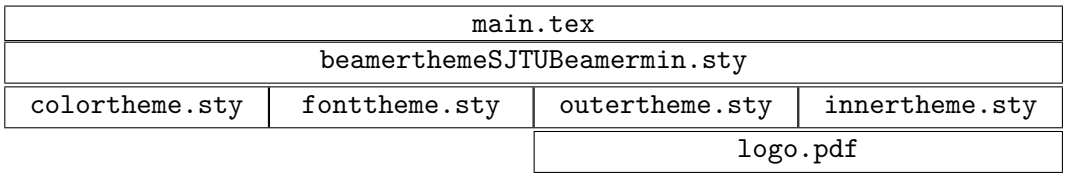
The main theme file `beamerthemeSJTUBeamermin.sty` is the entry point of the theme template. For users, after acquiring the beamer package, `\usetheme` command will serve as the caller of the theme.

```
\documentclass{beamer}
\mode<presentation>
\usetheme{SJTUBeamermin}
```

And this file will preprocess the option passed to the theme. Some options will be affected immediately, while others will get processed in the sub-style files.

3.2 Color

The color style file `beamercolorthemeSJTUBeamermin.sty` is the color setup of the template. Most color schemes are derived from the basic color of SJTU VI[**?**]. And to adapt the color definitions of beamer, the corresponding interface is mapped, see 17.2 in [**?**].



As it is mapped to those beamer interfaces, to use the color, you have to declare the color struct first by

```
\usebeamercolor{palette primary}
\color{palette primary.bg}
```

or simply

```
\usebeamercolor[bg]{palette primary}
```

However, there are scenarios where you cannot put temporary variables in some package options since it expands to `\color{\color{mycolor}}`. In this complex case, the redefinition of those standard colors is required. And that's the reason why `innertheme.sty` gets `color`.

3.3 Font

The font style file `beamerfontthemeSJTUBeamermin.sty` provides the font style of the beamer. In SJTUBeamer [MIN], serif math font is used by

```
\usefonttheme{professionalfonts}
```

which will tell beamer not to meddle with the specific font (in this case, math font) to the sans serif one.

It is especially useful if you don't want to create more compilation errors since some engine doesn't support sans serif math font. The workaround for that is to introduce the package below:

```
\RequirePackage[eulergreek]{sansmath}
```

And SJTUBeamer [MIN] does both.

3.4 Outer

The outer style file `beamerouterthemeSJTUBeamermin.sty` contains the layout of frames. The recommended setup is as follows:

`theme.sty` lang cjk gbt *other settings*	`colortheme.sty`	color
	`fonttheme.sty`	
	`outertheme.sty`	pattern,navigation,lang
	`innertheme.sty`	pattern,color,lang

3.5 Inner

The inner style file `beamerinnerthemeSJTUBeamermin.sty` will customize the main components.

Outer theme and inner theme are the core files for SJTUBeamer [MIN], which will be discussed in the following content.

4 Compatibility

Since the vision of LaTeX is to build an open-source typesetting system for multi-platforms and beamer is on top of that to create an easy-to-configure interface on building presentations, SJTUBeamer [MIN] follows the footstep to make its best on compatibility.

4.1 Beamer Interface

Beamer has designed a system of modern interfaces for those theme creators. SJTUBeamer [MIN] has already followed the modular architecture, as is shown in Section **??**.

And there are more APIs in beamer for each corresponding theme style. There are mainly three ways to modify a theme:

1. **Want to use presets.** Read Part III in the documentation of beamer package [**?**]. You can acquire the doc by the terminal command:

   ```
   texdoc beamer
   ```

   Then, you could choose to use some preset theme, or call the macro to control the appearance of each component.

2. **Want a complete modification.** Read the source code of beamer package [**?**]. If no additional theme is used, beamer will assume you are creating a theme from `default`. And refer to the corresponding theme file suffixed by `default` will give you the bottom mechanism to implement components.

interface	color=	red	blue
palette primary	cprimary	#004098	#9E1F36
palette secondary	csecondary	#298626	#F28101
palette tertiary	ctertiary	#004D4B	#FED201
palette quanternary	cquanternary	#FFFFFF	#000000

3. **Want to solve difficult problems.** Go to TEX Stack Exchange [**?**] for help. Always search before you ask. Then you could probably find some patches and magical formulas to tackle the issue since TEX is a Turing-complete language.

4.2 Mainstream Packages

Mainstream LATEX packages are used to make sure the choice on marcos is maintained currently. Since some engine doesn't support GhostScript well (*e.g.* XeLATEX), SJTUBeamer [MIN] (as well as beamer) uses PGF as the backend for graphics in PostScript. And half of the jobs are done on graphics to implement the requirements of VI.

SJTUBeamer [MIN] doesn't use too many rasterized pictures, since they are not flexible. You could get the Adobe Illustrator files on VI website[**?**]. SJTU VI goes minimalism so that it could be implemented by package Ti*k*Z (which is on top of PGF). You could almost draw any vectorized shapes by referring to Ti*k*Z documentation [**?**]. In short, Ti*k*Z uses node-edge system to create graphs and many Computer Science pictures can be drawn in such a system[**?**]. And if you don't want to mess around with the thousand pages of documentation, Ti*k*ZEdt could help you create that in a WYSIWYG(what you see is what you get) way[**?**], which is a tool to make drafts on patterns.

SJTUBeamer [MIN] also uses additional packages like PGFPLOTS and PGFPLOTSTABLE to draw highly personalized statistic graphs and layout table from CSV (Comma-Seperated Values) respectively. As is mentioned, the author created a tool PGFPLOTSEDT to help such graphs in an interactive way[**?**].

Code blocks are drawn by package tcolorbox, which is also a powerful toolkit to make customized boxes[**?**]. This is almost the most elegant way to make colorful boxes in the current LATEX system.

Some of the packages have been studied by author in LATEX Sparkle Project[**?**]. You can check that out to learn more.

4.3 Engine Support

To be clear, SJTUBeamer [MIN] is not adapt to all kinds of compilers in the current LATEX world.

SJTUBeamer [MIN] make its effort on engine support in the following ways:

Components	SJTUBeamer MIN
head- and footline	•
sidebars	
logo	•
frame title	•

1. **Use beamer interface.** As is mentioned in Section **??**, SJTUBeamer MIN will not create its macro unless there is no substitute in the current version of beamer or it is a common method to implement some features. A good example for this is to make a bottom page, SJTUBeamer MIN mimicked `\maketitle` command to implement `\makebottom` command. A good outcome is that the style file could be separately used with low coupling.

2. **Use mainstream packages.** Mentioned in Section **??**, mainstream packages are widely accepted in many engines. Some top-level marcos are used to increase the readability of the source code, i.e., PGF is lengthy and hard to be maintained.

3. **Use old-fashioned TEX code.** If there is a nice way to implement in TEX, then go TEX. TEX is a box-based typesetting system, which may be mentioned in many Computer Science books. And LATEX is on top of that to provide clear-to-read macros. In some scenarios, the native `\vbox` and `\hbox` command could help calculate the position of characters in a more controllable way. But it is certainly painful to learn. The TEX Book[**?**] is the classic to learn that, but Notes On Programming in TEX[**?**] is more recommended in modern LATEX.

5 Implementation

Now, you may still be confused about how to create a beamer template. Here is a good material about it for a lead-in[**?**], which provides a brief overview. And this part is only focusing on the implementation of SJTUBeamer MIN.

sjtucolordef.dtx sjtuvishape.dtx beamerthemesjtubeamermin.dtx beamercolorthemesjtubeamermin.dtx beamerfontthemesjtubeamermin.dtx beamerinnerthemesjtubeamermin.dtx beameroutterthemesjtubeamermin.dtx

Good Luck with SJTUBeamer MIN !

Developer

Log Creative

Components	SJTUBeamer MIN
Title and part pages	•
Itemize	•
Enumerate	
Description	
Block	
Theorem and proof	
Figures and tables	•
Footnotes	•
Bibliography entries	

	Windows	Unix
pdfLaTeX(CTeX)	√	
pdfLaTeX(CJK)	√	√
XeLaTeX	♢	√
LuaLaTeX	♢	♢

*√ is fully available, while ♢ will have font issues.