nuxtex.txt

2024年9月10日

LaTeX quickfix and SyncTeX plugin for Vim/Neovim on Linux

1 INTRODUCTION

NuxTeX is a (La)TeX editting support plugin for Vim and Neovim. This plugin supports —quickfix— function and SyncTeX. It is desigined to minimalize the lerning points for user. So the goal is the very small configuration for neccesarry to work and use the default Vim commands interface as possible. So the plugin specified commands you should to learn are a few. In the most cases, the only you should memory is the forward search command. I think the most plugins released by today increase the learning costs that Vim has already had a lot of things to learn. Is it really useful? For the point of view, I keep in mind to make simple this plugin as described in above. The —quickfix— is the default Vim/Neovim command interface. So it is activated by —:compiler— command and you can use —:make—, —:cfile— and more default —quickfix— commands with optimized output for the (La)TeX. This is simlilar as —tex.vim—, but much more better especially for unnecessary message reduction. The SyncTeX function supports GNOME Document Viewer(Evince), Atril, Xreader and Zathura. Both forward and backward search are supported. It is not neccesarry to complex configuration for these pdf viewers to work backward search function. This plugin also supports multiple source project like using
input command.

The reference is ([**?**]).

2 DEPENDENCIES

NuxTeX depends on python-dbus and urllib with python3 to work SyncTeX and ofcource LaTeX distoribution like texlive to compile the documents. The python components will be installed default for the most of desktop Linux distoributions.

3 USAGE

1. Install this plugin and dependencies (if it is neccesarry). See —nuxtex-dependencies—.

2. Set up the document compilation method. See detail in —nuxtex-set-compiler—.

3. Configure `g:nuxtex_viewer_type` as the pdf viewer which you would like to use. The value can be chosed from 'evince', 'atril', 'xreader' and 'zathura'. See `nuxtex-synctex`.

4. Open the LaTeX source and edit it. Once you would like to compile the source, first, choose the compiler plugin as NuxTeX by ':compiler nuxtex'. After that, —:make— to compile. If you don't

have any error, go next step. Otherwise, do —:copen— and check the error occured line and fix the root cause. Then, try compile again while there are no error.

5. After compilatoin succeed, you can jump both from source to pdf and pdf to source. To jump to pdf, just —¡localleader¿¡localleader¿nf— on the LaTeX source. To jump from pdf to source, Ctrl-¡Left Click¿ on the point you would like to see the source on the viewer.

$$\dot{x}(t) = ax(t) + bu(t) \tag{1}$$

4 Set compiler

You can compile the (La)TeX document using by `:make` or `:lmake`. Before execute these commands, you should set up the compilation method. There are some methods. I will explain some examples.

Example 1: Use make shell command

By default, `:make` command execute make shell command. You can also use this option on NuxTeX. In this case, you shoule prepare Makefile described compilation method. If you would like to compile the 'foo.tex' document by latex shell command, you should put the 'Makefile' file in the same directory of the 'foo.tex' and it is possible to describe as below.

Makefile:

```
>
DIR␣=␣./
DVI␣=␣$(DIR)foo.dvi
PDF␣=␣$(DIR)foo.pdf

$(PDF):␣␣␣␣$(DVI)
dvipdfmx␣-o␣$(PDF)␣$<

$(DVI):␣␣␣␣foo.tex
latex␣-synctex=1␣-output-directory=$(DIR)␣$<

clean:;␣latexmk␣-C␣--outdir=$(DIR)
<
```

Example 2: Set `b:nuxtex_makeprg` or `g:nuxtex_makeprg`

It is possible to set directly LaTeX compiler without create Makefile. You can set the compile method by `b:nuxtex_makeprg` or `g:nuxtex_makeprg`. The `b:nuxtex_makeprg` is only applied on the buffer configured itself and `g:nuxtex_makeprg` is applied in the whole vim session. These variables should be configured before execute ':compiler nuxtex' firstly in the session because they are read when executing ':compiler nuxtex' and once executed, it will not be loaded more than one time in the buffer even if the command executed in the buffer twice or more. If both the variables are configured, `b:nuxtex_makeprg` will be prefered than `g:nuxtex_makeprg`. These values will overwrite `makeprg`.

In case of latex:

```
>
let␣b:nuxtex_makeprg␣=␣'latex␣-synctex=1␣%:p'
<

You␣can␣set␣the␣compiler␣globally␣in␣the␣vimrc.
>
```

```
let␣g:nuxtex_makeprg␣=␣’latex␣-synctex=1␣%:p’
<
```

Example 3: Use latexmk

Example 2 case is only possible to generate dvi by `:make` command. To generate pdf at one time, you can use latexmk.

```
>
let␣g:nuxtex_makeprg␣=␣"latexmk␣-pdfdvi␣-latex=latex␣-synctex=1␣-e␣\"\\$dvipdf=’dvipdfmx␣\
<
```

Example 4: Directly update `makeprg`. As described in above, `b:nuxtex_makeprg` and `b:nuxtex_makeprg` will update `makeprg` when ‘:compiler nuxtex‘ has been done. You can also update the compile method by directly update `makeprg`. In this method, you can update compiler after once ‘:compiler nuxtex‘ has been executed. You may edit some scripts if you would like to multiple compilers in multiple laungage without using `b:nuxtex_makeprg` or `g:nuxtex_makeprg`.

```
>
let␣&makeprg␣=␣"latexmk␣-pdfdvi␣-latex=latex␣-synctex=1␣-e␣\"\\$dvipdf=’dvipdfmx␣\\%O␣\\%S
<
```

It is also possible to change only buffer local compile option.

```
>
let␣&l:makeprg␣=␣"latexmk␣-pdfdvi␣-latex=latex␣-synctex=1␣-e␣\"\\$dvipdf=’dvipdfmx␣\\%O␣\\
<
```

5 Compile (La)TeX documents

Once set up the compile method at —nuxtex-set-compiler—, then, you can compile the documents by —:make— or —:lmake—. The first command will output the compiler message in the quickfix list and the last one will output in the location list. How to open these lists is described in —nuxtex-quickfix—.

6 Load quickfix list from compiler log

If you would like to load the past compile error or warning to quickfix list, you can load *.log file by —:cfile— command after set up —nuxtex-set-compiler—. The *.log file is generated by (La)TeX compiler default. For example, if ‘latex foo.tex‘ was executed, ‘foo.log‘ will be generated automatically. Then, you will launch vim, and can import the log file by these commands.

¿ :compiler nuxtex :cfile foo.log :copen ¡

If you would like to load the log to location list, you can use —:lfile— command instead of —:cfile—.

¿ :compiler nuxtex :lfile foo.log :lopen ¡

7 Supported —quickfix— commands

NuxTeX currently supports below quickfix commands. Once execute ‘:compiler nuxtex‘, these commands will optimize quickfix/loclist outputs for (La)TeX compiler. After execute these commands, you can see the quickfix list by —:copen— or —:cwindow— and the location list by —:lopen— or —:lwindow—. See details in —quickfix—.

- —:make—
- —:lmake—
- —:c[file]—
- —:l[file]—
- —:cb[uffer]—
- —:lb[uffer]—
- —:cg[etfile]—
- —:lg[etfile]—
- —:cgetb[uffer]—
- —:lgetb[uffer]—

Bellow commands are currently not supported. These results will NOT be fixed by NuxTeX.

- —:caddf[ile]—
- —:laddf[ile]—
- —:cad[dbuffer]—
- —:laddb[uffer]—

8 Forward and backward search

Once the (La)TeX document compilation succeed, it is possible to jump from the source to the corresponding point of the generated pdf (forward search) or jump from pdf to the corresponding point of the source (backward search). This is called SyncTeX feature. NuxTeX supports this feature. Also, there are some pdf viewers support SyncTeX. NuxTeX supports below viewers.

- GNOME Document Viewer (Evince)
- Atril
- Xreader
- Zathura

These viewers are constantlly used in GNU/Linux distributions. Why NuxTeX supports these viewers is that they have DBus SyncTeX interfaces. These interfaces broadcast (La)TeX source point when the backward search is executed. So you have not to set up the complex configuration for the viewers. The DBus protocol is focused on GNU/Linux so because of this, NuxTeX is focused on the platform.

Before using forward search, you should choose a pdf viewer to open the product. This is possible by configure `g:nuxtex_viewer_type`. The parameter can be set ’evince’, ’atril’, ’xreader’ and ’zathura’.

Example:

```
>
"To␣choose␣Evince
let␣g:nuxtex_viewer_type␣=␣'evince'

"To␣choose␣Atril
let␣g:nuxtex_viewer_type␣=␣'atril'

"To␣choose␣Xreader
let␣g:nuxtex_viewer_type␣=␣'xreader'
```

```
"To␣choose␣Zathura
let␣g:nuxtex_viewer_type␣=␣'zathura'
<
```

If the parameter has not be set, by the default, Evince will be used for open the product.

After set the parameter, it is possible to forward search by type `<localleader><localleader>nf` on the (La)TeX source. You can execute backward search by Ctrl-Left click on the pdf viewers. The backward search function will be activated after the once forward search command was executed.

NuxTeX SyncTeX function supports multple file project. The backward search function will search the (La)TeX source stored in the buffer. Also, the forward search function supports multiple files. For this feature, the plugin have to search the output pdf. The algorithm search the source by below order.

1. Check the file path described in `b:nuxtex_output_pdf` if it was set.

2. If `b:nuxtex_output_pdf` was not set, check the file path written in ‘

3. If ‘search the output pdf for parental directory recursively. If ‘gzip‘ command is installed on the system and set `g:nuxtex_gz_parse` as `v:true`(this is the default configuration), the plugin will analyze *.synctex.gz file and check the file is whether it is for the (La)TeX source. This is useful for if there are multiple *.gz file on the own and parental directory of the source. In this case, the plugin will search matched *.synctex.gz file and tell the pdf viewer to the location of source and the pare of the source and pdf.

On the 3. section, the plugin assume the output pdf and *.synctex.gz file is on the same directory.

9 OPTIONS

```
b:nuxtex_makeprg *b:nuxtex_makeprg*
Type:␣string
This␣value␣defines␣the␣compiler␣command␣to␣execute␣if␣the␣compiler
plugin␣selected␣as␣nuxtex.␣This␣is␣similar␣as␣|g:nuxtex_makeprg|.

g:nuxtex_makeprg *g:nuxtex_makeprg*

Type:␣string
This␣value␣defines␣the␣compiler␣command␣to␣execute␣if␣the␣compiler
plugin␣selected␣as␣nuxtex.␣This␣parameter␣should␣be␣set␣before␣execute
`:compiler␣nuxtex`.

Example:
>
let␣g:nuxtex_makeprg␣=␣"latexmk␣-pdfdvi␣-latex=latex␣-synctex=1␣-e␣\"\\$dvipdf='dvipdfmx␣\
<
This␣example␣for␣using␣latexmk␣to␣compile␣`latex`␣command,␣convert␣dvi
by␣`dvipdfmx`␣and␣use␣`bibtex`.
See␣also␣|nuxtex-set-compiler|.

b:nuxtex_output_pdf *b:nuxtex_output_pdf*
Type:␣string
This␣value␣is␣user␣manual␣configuration␣of␣␣output␣pdf␣file␣generated
from␣the␣(La)TeX␣source.␣This␣value␣is␣buffer␣variable,␣so␣it␣should
```

```
be␣set␣on␣the␣buffer␣opened␣the␣(La)TeX␣source␣you␣would␣like␣to
check␣the␣output.

Example:
>
let␣b:nuxtex_output_pdf␣=␣'/path/to/foo/bar.pdf'
<

g:nuxtex_force_quickfix *g:nuxtex_force_quickfix*
Type:␣boolean
Default:␣|v:false|
This␣value␣activate␣NuxTeX␣quickfix␣feature␣forcibly␣even␣if␣any
compiler␣plugin␣selected.␣This␣plugin␣detect␣compiler␣plugin␣set␣for
it␣for␣the␣buffer␣in␣the␣|QuickFixCmdPre|␣timing.␣So␣user␣or␣other
plugins␣select␣other␣buffer␣in␣the␣timing,␣NuxTeX␣cannot␣detect␣the
target␣buffer␣set␣as␣`:compiler␣nuxtex`␣in␣past.
It␣is␣useful␣for␣in␣this␣case,␣but␣it␣is␣not␣recommended␣to␣set␣the
parameter.␣Set␣`let␣g:nuxtex_force_quickfix␣=␣v:true`␣to␣activate
this␣feature.

g:nuxtex_gz_parse *g:nuxtex_gz_parse*
Type:␣boolean
Default:␣|v:true|
This␣variable␣activate␣or␣deactivate␣*.synctex.gz␣file␣parse␣for
Forward␣search␣feature.␣If␣this␣variable␣set␣as␣|v:true|␣and␣`gzip`
command␣is␣installed␣in␣the␣$PATH,␣this␣plugin␣will␣analyze
*.synctex.gz␣file␣when␣Forward␣search␣command␣is␣executed␣and␣choose
correct␣*.synctex.gz␣file␣generated␣from␣the␣(La)TeX␣source.␣This␣is
useful␣for␣there␣are␣multiple␣*.synctex.gz␣files␣are␣in␣the␣directory
by␣some␣reasons.
If␣there␣are␣any␣reasons,␣it␣is␣recommended␣to␣set␣no␣configuration
for␣the␣parameter.␣If␣you␣would␣like␣to␣set␣this␣parameter␣as␣|v:true|
but␣`gzip`␣command␣has␣not␣been␣installe␣in␣the␣$PATH,␣you␣can␣set
|g:nuxtex_gzip_path|.
See␣also␣|nuxtex-synctex|.

g:nuxtex_gzip_path *g:nuxtex_gzip_path*
Type:␣string
It␣is␣possible␣to␣set␣the␣`gzip`␣command␣path␣by␣setting␣this␣variable.
It␣is␣useful␣if␣|g:nuxtex_gz_parse|␣is␣set␣as␣v:true,␣but␣`gzip`
command␣is␣not␣installed␣in␣the␣$PATH.

Example:
>
let␣g:nuxtex_gzip_path␣=␣'/usr/local/bin/gzip'
<

g:nuxtex_open_method *g:nuxtex_open_method*
Type:␣char
Default:␣'t'
```

This␣variable␣for␣configure␣the␣rule␣for␣manipulating␣the␣buffer␣not opened␣in␣any␣windows␣(but␣stored␣in␣the␣buffer)␣when␣the␣Backward search␣function␣executed.
The␣default␣value␣is␣'t'␣and␣it␣means␣it␣will␣open␣the␣stored␣buffer in␣the␣new␣tab␣when␣the␣Backward␣feature␣select␣the␣buffer␣called␣from pdf␣viewer.
This␣variable␣can␣be␣set␣as␣below.

't'Open␣stored␣buffer␣in␣the␣new␣tab.

'h'Split␣the␣currrent␣window␣and␣open␣stored␣buffer␣in␣the␣left␣hand␣side of␣the␣current␣window.

'j'Vertical␣split␣the␣currrent␣window␣and␣open␣the␣stored␣buffer␣in␣the below␣of␣the␣current␣window.

'k'Split␣the␣currrent␣window␣and␣open␣stored␣buffer␣in␣the␣right␣hand␣side of␣the␣current␣window.

'l'Vertical␣split␣the␣current␣window␣and␣open␣the␣stored␣buffer␣in␣the above␣side␣of␣the␣current␣window.

'H'Open␣the␣new␣window␣in␣the␣left␣side␣of␣the␣current␣tab␣and␣display the␣stored␣buffer.

'J'Open␣the␣new␣window␣in␣the␣bottom␣of␣the␣current␣tab␣and␣display␣the stored␣buffer.

'K'Open␣the␣new␣window␣in␣the␣top␣of␣the␣current␣tab␣and␣display␣the stored␣buffer.

'L'Open␣the␣new␣window␣in␣the␣right␣side␣of␣the␣current␣tab␣and␣display the␣stored␣buffer.

'c'Open␣the␣stored␣buffer␣in␣the␣current␣window.

g:nuxtex_python_cmd *g:nuxtex_python_cmd*
Type:␣string
Default:␣'python3'
This␣variable␣is␣for␣to␣configure␣the␣path␣and␣execution␣file␣name of␣the␣python.␣The␣plugin␣use␣python␣and␣some␣libraries␣to␣execute SyncTeX␣feature.␣So␣it␣should␣be␣set␣as␣correct␣python␣command␣to execute␣the␣feature.

Example:
>
let␣g:nuxtex_python_cmd␣=␣'/usr/local/bin/python'
<

g:nuxtex_sys_enc *g:nuxtex_sys_enc*

```
Type:␣string
This␣variable␣is␣used␣in␣analyzing␣*.synctex.gz␣file.␣It␣defines␣‘gzip‘
command␣stdout␣|encoding|.␣NuxTeX␣will␣convert␣the␣stdout␣|encoding|
from␣this␣variable␣to␣editor␣defined␣|&enc|.␣This␣feature␣stands␣on
|iconv()|.

Example:
>
let␣g:nuxtex_sys_enc␣=␣"utf-8"
>

g:nuxtex_viewer_type *g:nuxtex_viewer_type*
Type:␣string
Default:␣’evince’
This␣variable␣is␣used␣for␣select␣pdf␣viewer␣used␣in␣SyncTeX␣feature.
It␣is␣possible␣to␣select␣from␣’evince’,␣’atril’,␣’xreader’␣and
’zathura’.␣If␣it␣is␣not␣selected,␣GNOME␣Document␣Viewer␣(Evince)␣will
be␣used␣in␣the␣plugin␣default.
See␣also␣|nuxtex-synctex|.

Example:
>
"␣Select␣pdf␣viewer␣as␣GNOME␣Document␣Viewer␣(Evince)
let␣g:nuxtex_viewer_type␣=␣’evince’

"␣Select␣pdf␣viewer␣as␣Atril␣(Default␣viewer␣of␣MATE␣Desktop)
let␣g:nuxtex_viewer_type␣=␣’atril’

"␣Select␣pdf␣viewer␣as␣Xreader␣(Default␣viewer␣of␣Linux␣Mint␣Cinnamon)
let␣g:nuxtex_viewer_type␣=␣’xreader’

"␣Select␣pdf␣viewer␣as␣Zathura␣(It␣has␣Vim␣like␣keybind)
let␣g:nuxtex_viewer_type␣=␣’zathura’
>

g:nuxtex_zathura_cmd *g:nuxtex_zathura_cmd*
Type:␣string
Default:␣’zathura’
This␣valiable␣defines␣the␣path␣and␣execution␣file␣name␣of␣the␣zathura.
It␣is␣useful␣for␣if␣you␣would␣like␣to␣use␣zathura␣not␣installed␣in␣the
\$PATH.␣It␣is␣only␣activated␣if␣|g:nuxtex_viewer_type|␣is␣set␣as
’zathura’.

Example:
>
let␣g:nuxtex_zathura_cmd␣=␣’/usr/local/bin/zathura’
<

g:nuxtex_zathura_opt *g:nuxtex_zathura_opt*
Type:␣string
```

```
Default:␣'--synctex-forward␣"@line:@col:@tex"␣"@pdf"'
This␣variable␣defines␣the␣zathura␣command␣option␣for␣Forward␣search
feature.␣The␣below␣strings␣will␣be␣substituted␣in␣the␣command.

-␣'@line'␣will␣be␣replaced␣in␣the␣current␣buffer␣line␣number.
-␣'@col'␣will␣be␣replaced␣in␣the␣current␣buffer␣column␣number.
-␣'@tex'␣will␣be␣replaced␣in␣the␣current␣buffer␣file␣name.
-␣'@pdf'␣will␣be␣replaced␣in␣the␣(La)TeX␣output␣pdf␣file␣name␣found␣in␣the
␣␣␣␣␣␣␣␣␣logic␣described␣in␣|nuxtex-synctex|.

For␣example,␣in␣case␣of␣|g:nuxtex_zathura_cmd|␣set␣as␣'zathura',␣this
variable␣set␣as␣'--synctex-forward␣"@line:@col:@tex"␣"@pdf"',␣the
cursor␣position␣is␣100␣line␣and␣5␣column␣of␣the␣'foo.tex'␣and␣the
output␣pdf␣file␣is␣'/path/to/foo.pdf',␣the␣plugin␣will␣execute␣command
as␣below.
>
zathura␣--synctex-forward␣"100:5:foo.tex"␣"/path/to/foo.pdf"
<

This␣variable␣is␣used␣in␣if␣|g:nuxtex_viewer_type|␣set␣as␣'zathura'.
```

10 Mappings

```
*:<localleader><localleader>nf*
<localleader><localleader>nf Execute␣forward␣search.␣The␣current␣cursor␣position
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣and␣the␣(La)TeX␣source␣file␣path␣will␣be␣notified␣to
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣the␣pdf␣viewer␣set␣in␣|g:nuxtex_viewer_type|.
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣It␣is␣only␣activeted␣if␣the␣|filetype|␣of␣the␣current
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣buffer␣is␣'tex'.␣It␣is␣possible␣to␣check␣by␣the␣result
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣of␣the␣`echo␣&ft`.
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣This␣command␣behaviour␣is␣same␣as
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣|:NuxtexFwd|.
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣See␣also␣|nuxtex-synctex|.
```

11 CHANGELOG

```
0.1 2024/3/10
First␣release

vim:tw=78:ts=8:ft=help:norl:
```

参考文献

[1] Mark: Unknown title, Special Text, vol1, pp.24–27 (2020)

[2] Mike: UFO text, Special Text, vol1, pp. 28–30 (2020)

参考文献