
VNote Documentation

Release 1.11.1

Le Tan

Feb 13, 2019

1	Why VNote	3
1.1	What is VNote	3
1.2	Why Another Markdown Wheel	3
2	Getting Started	5
2.1	Main Interface	5
2.2	Ready To Go	7
3	Build VNote	9
3.1	Get the Source Code of VNote	9
3.2	Get Qt 5.9	9
3.3	Windows	9
3.4	Linux	9
3.5	MacOS	10
4	Notes Management	13
4.1	Notebook	13
4.2	Folders	14
4.3	Notes	14
5	Snippet	15
5.1	Snippet Management	15
5.2	Define A Snippet	16
5.3	Apply A Snippet	16
5.4	Examples	16
6	Magic Word	19
6.1	Built-In Magic Words	19
6.2	Custom Magic Words	20
6.3	Magic Word In Snippet	21
7	Template	23
8	Themes and Styles	25
8.1	Themes	25
8.2	Editor Styles	30
8.3	Rendering Styles	32

8.4	Code Block Rendering Styles	33
9	Export	35
9.1	Sources	36
9.2	Options	37
9.3	Formats	37
10	Full-Text Search	41
10.1	Keywords	42
10.2	Scope	42
10.3	Object	42
10.4	Target	42
10.5	File Pattern	42
11	Frequently Asked Questions	43
11.1	How to specify customized MathJax script?	43
11.2	How to make VNote more “portable”?	44
11.3	How to draw diagrams using Mermaid or Flowchart.js?	44
11.4	How to open a note in another tab?	45
11.5	Interface is freezed	46
11.6	Cursor is not visible in edit mode	46
11.7	Crash when opening a note	46
11.8	Menu does not work in fullscreen mode	46



Designed specially for Markdown, [VNote](#) is a Vim-inspired note-taking application, which knows programmers and Markdown better.

VNote is open source, and available on [Github](#).

There are documents in several languages.

- [English](#)
- [Simplified Chinese](#) ()
- [Japanese](#) ()

The main documentation for the site is organized into a couple of sections:

- [User Documentation](#)
- [Advanced Documentation](#)
- [Developer Documentation](#)
- [About VNote](#)

1.1 What is VNote

VNote is a Vim-inspired **Note**-taking application, designed specially for **Markdown**. It is not yet just another Markdown editor. VNote is designed to be a powerful Markdown editor with sweet note management, or a note-taking application with pleasant Markdown experience.

VNote is free, and open source. You could get VNote for Linux, Windows, and macOS.

1.2 Why Another Markdown Wheel

1.2.1 Insights About Markdown

Markdown, as a simple mark language, unlike rich text, was born with a **gap between edit and read**. There are about three ways to handle this gap:

1. As one extreme, some editors just treat Markdown as **plain text**. Users may lose themselves in the messy black characters. It is **hard to keep track** of the information of the note.
2. Most Markdown editors use two side-by-side panels to **edit and preview Markdown notes simultaneously**. It makes things easier since users could see a pleasant typesetting and layout while editing the text. However, two panels may occupy the whole screen and users keep moving eyes left and right, which will be a big distraction.
3. As another extreme, some editors convert the Markdown elements in place immediately after the typing, which makes editing Markdown just like editing rich text document in Word with some shortcuts. This may conflict with what Markdown is intended to be.

Since most editors choose the second way to handle the gap, people always think of preview when it comes to Markdown. It may be a **MISUNDERSTANDING** about Markdown. Designed as a simple mark language, Markdown is intended to help to keep track of the information of the text when editing and provide a beautiful typesetting when reading after being converted to HTML.

1.2.2 Tradeoff: VNote's Way

VNote tries to minimize the gap and provide the best-effort WYSIWYG for Markdown via **tuned syntax highlights and some other features**. By helping to keep track of the content, there is no need to preview or alter the text immediately after being typed.

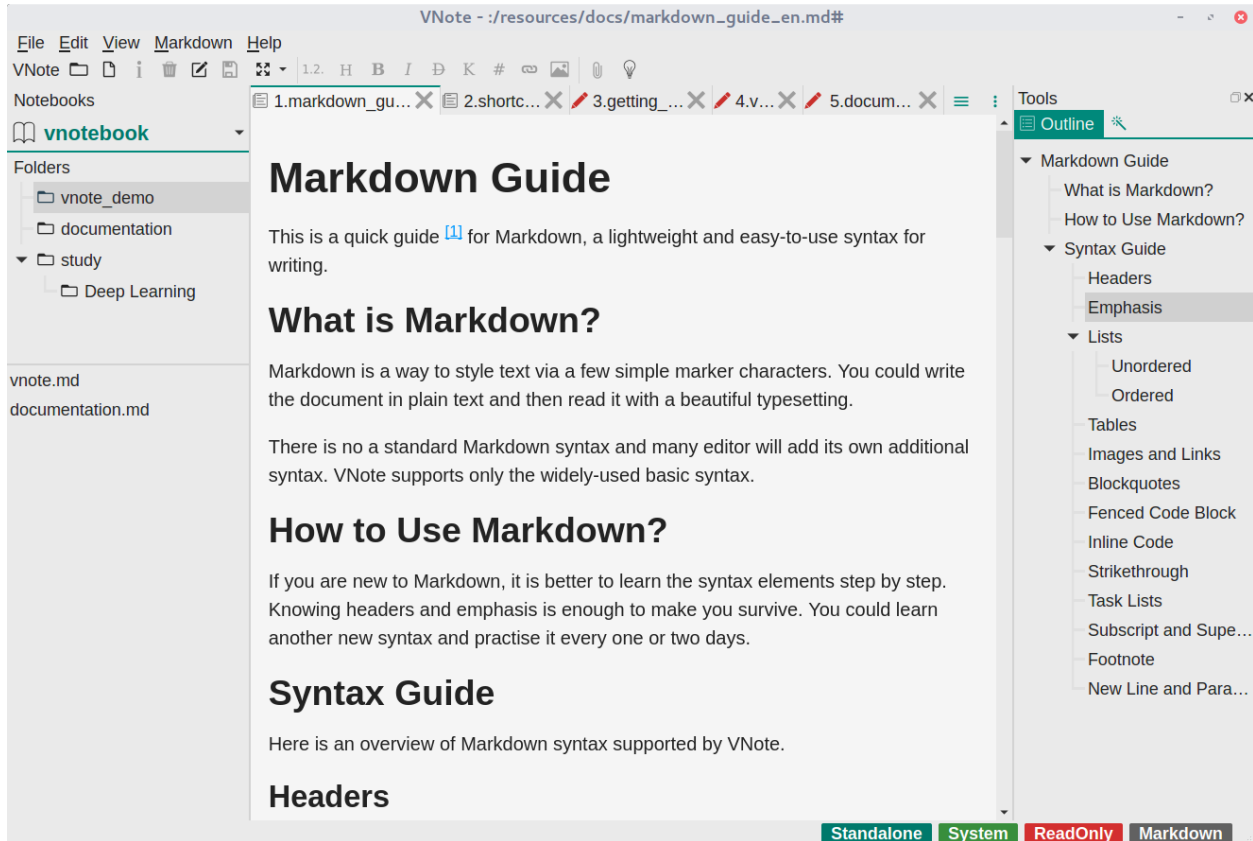
VNote is distributed on **Linux**, **Windows**, and **macOS**. You could always download standalone packages from the [Downloads](#) page.

It is recommended to download and use the **stable releases** of VNote. If you prefer to use some latest features of VNote, you could try the latest continuous build from `master` branch.

Besides the standalone packages, some package managers may already package and distribute VNote, such as *AUR*, and *homebrew cask*.

2.1 Main Interface

Here is a snapshot of VNote's main interface.



2.1.1 Menubar and Toolbar

VNote provides some settings via **Menubar** at the top of the window and provides some handy shortcuts via **Toolbar** right below the menubar. Place mouse over the menu item or button, there is a tooltip or status tip to help illustrating the function of that item or button.

2.1.2 Notebooks, Folders, and Notes

VNote uses **Notebooks**, **Folders**, and **Notes** hierarchy for note management. The left-side panel is used to navigate through your notebooks, folders, and notes.

At the first time you launch VNote, you will be prompted to create your first notebook. A notebook corresponds to a directory in your file system. You should choose an empty directory as your *Notebook Root Folder*. All the contents of this notebook will be stored within the Notebook Root Folder.

Once you have a notebook, you could create folders in this notebook. VNote supports infinite levels of folders.

The bottom panel at the left side will list all the notes of current folder. You could create a new note via the context menu.

2.1.3 Content Area

The middle area which occupies most of the space is the **Content Area**, which is used to read and edit notes. VNote supports opening multiple notes by tabs.

2.1.4 Tools Dock Widget

At the right-side panel, there are multiple tools to facilitate your reading and editing, such as **Outline** and **Snippets**.

2.2 Ready To Go

That is it! Now you could read, edit, and manage your notes in VNote!

You need **Qt 5.9** to build VNote from source.

3.1 Get the Source Code of VNote

VNote's source code is available on [Github](#). You could download the ZIP archive of the code. Please be aware of that VNote depends on some submodules, so you should also download the source codes of these modules.

The recommended way is using **git** like this:

```
git clone https://github.com/tamllok/vnote.git vnote.git
cd vnote.git
git submodule update --init
```

3.2 Get Qt 5.9

You could get the standalone Qt SDK from [Qt Downloads](#). For users in China, you could speed up the download via the [TUNA Mirrors](#).

3.3 Windows

On Windows, you need **Visual Studio 2015** or above to compile VNote.

Open **Qt Creator** and open `vnote.git\VNote.pro` as project. Now you are ready to tune and compile VNote!

3.4 Linux

In Ubuntu, you could get Qt 5.9 from PPA like this:

```
sudo add-apt-repository ppa:beineri/opt-qt591-trusty -y
sudo apt-get update -qq
sudo apt-get -y install qt59base qt59webengine
sudo apt-get -y install qt59webchannel qt59svg qt59location qt59tools qt59translations
source /opt/qt*/bin/qt*-env.sh
```

Then compile and install VNote like this:

```
cd vnote.git
mkdir build
cd build
qmake ../VNote.pro
make
sudo make install
```

3.4.1 Fcix

If you use **Fcix** as the input method, you need to copy the missing library `libfcitxplatforminputcontextplugin.so` to Qt's plugin directory.

To find the place of `libfcitxplatforminputcontextplugin.so`, you could execute:

```
fcitx-diagnose | grep libfcitxplatforminputcontextplugin.so
```

If there is no such lib, you may need to install and configure Fcix for Qt5 correctly before continue.

Then you need to copy the lib to Qt's plugin directory:

```
<path_to_Qt_installation_directory>/5.9.3/gcc_64/plugins/platforminputcontexts/
```

3.4.2 OpenSSL

VNote needs **openssl 1.0** for networking. To verify if it has been setup correctly, you could check for updates in the *Help* menu of VNote. If VNote fails to check for updates, then you need to copy libraries of openssl to Qt's library directory.

After the installation of openssl, you could find two lib files:

```
/usr/lib/libcrypto.so.1.0.0
/usr/lib/libssl.so.1.0.0
```

Copy these two files to Qt's library directory:

```
<path_to_Qt_installation_directory>/5.9.3/gcc_64/lib/
```

In Qt's library directory, create symlinks for these two files:

```
ln -s libcrypto.so.1.0.0 libcrypto.so
ln -s libssl.so.1.0.0 libssl.so
```

3.5 MacOS

If you prefer command line on macOS, you could follow these steps.

1. Install Xcode and Homebrew;
2. Install Qt 5.9.1 via Homebrew:

```
brew install qt@5.9.1
```

3. In the project directory, create `build_macos.sh` like this:

```
QTDIR="/usr/local/opt/qt@5.9.1"
PATH="$QTDIR/bin:$PATH"
LDFLAGS=-L$QTDIR/lib
CPPFLAGS=-I$QTDIR/include

mkdir -p build
cd build
qmake -v
qmake CONFIG-=debug CONFIG+=release ../VNote.pro
make -j2
```

4. Make `build_macos.sh` executable and run it:

```
chmod +x build_macos.sh
./build_macos.sh
```

5. Now you got the bundle `path/to/project/build/src/VNote.app`.

VNote adopts **notebooks-folders-notes** hierarchy for notes management. A notebook corresponds to a directory in the file system, which is called **Notebook Root Folder**. Folders of a notebook correspond to directories within the Notebook Root Folder. Notes inside a folder corresponds to files within that directory.

4.1 Notebook

Notebook is an independent, self-explanatory container in VNote. A notebook is a **Notebook Root Folder** in the file system. The root folder contains all the notes and configuration files of that notebook.

4.1.1 Create A Notebook

You could create a new notebook by specifying following fields:

- **Notebook Name**
 - Name of your notebook in VNote. It is only used to identify your notebook in VNote. It will not be written into the configuration of the notebook.
- **Notebook Root Folder**
 - Choose an **EMPTY** directory in your system to hold all the contents of this notebook. This chosen directory is assumed to be in the control of VNote.
- **Image Folder**
 - This is the name of the folder used to store local images of notes. VNote uses a given folder which has the same parent folder of the notes to store images of those notes.
- **Attachment Folder**
 - This is the name of the folder used to store attachment files of notes.

4.1.2 Migrate and Import A Notebook

A notebook is an independent directory in the file system, so you could just copy or synchronize the *Notebook Root Folder* to migrate a notebook.

You could import an existing notebook into VNote by selecting its *Notebook Root Folder* when creating a notebook. VNote will try to read the configuration files to restore the notebook.

Combining these, you could create your notebooks in a directory which is synchronized via third-party service, such as Dropbox and OneDrive, and then in another computer, you could import that directory into VNote as a notebook. With this, you could use VNote to edit and manage your notes, which will be synchronized by other trusted services, both at home and at work.

4.2 Folders

The hierarchy of folders within a notebook is the same as that of the directories within the *Notebook Root Folder*. You could create as many as possible levels of folders theoretically.

4.3 Notes

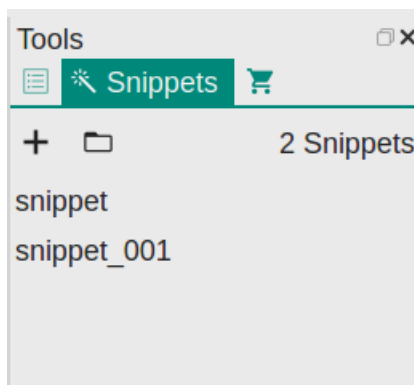
You could import external files into VNote as notes by the `New Notes From Files` action in `File` menu. VNote will **copy** selected files into current folder as notes.

In theory, a note in VNote could use any suffix. Notes with suffix `.md` will be treated as Markdown files.

Snippet is designed to facilitate the repeated input of some words.

5.1 Snippet Management

You could manage snippets via the `Tools` dock widget.



In this panel, we could:

1. Create a snippet;
2. Open snippet folder;VNote uses a folder in the configuration folder to store all the snippets and use configuration file `snippet.json` for management. Each snippet corresponds to one file.
3. View and edit the information about a snippet;
4. Sort and delete snippets;
5. Apply (insert) a snippet into the editor;

5.2 Define A Snippet

A snippet contains:

- Snippet Name
 - Identifies a snippet. It is also the name of the corresponding file in the snippet folder.
- Snippet Type
 - Plain text or HTML. Currently only plain text is supported.
- Shortcut
 - You could designate one of the 26 characters (a to z) to insert snippet quickly.
- Cursor Mark
 - VNote uses Cursor Mark to mark the position of the cursor after applying a snippet. Should appears only once in the `Content`.
- Selection Mark
 - VNote uses Selection Mark to mark the position to insert the selected text before applying a snippet. Selection mark could appear multiple times in the `Content`. After applying a snippet, all the selection marks will be replaced with the selected text.
- Content
 - The content of the snippet which will be inserted at the position of current cursor while applying a snippet.

5.3 Apply A Snippet

5.3.1 Snippet Panel

Place the cursor at the proper position and double click a snippet in the snippet panel to apply a snippet.

5.3.2 Shortcuts

You could use **Captain Mode** to apply a snippet quickly. In edit mode, press `Ctrl+E S` to activate snippet selection dialog, which shows all the snippets **with shortcut defined**. Press the corresponding shortcut key to apply a snippet.

5.4 Examples

5.4.1 Insert Code Block with CPP

```
```cpp
@@
```
```

5.4.2 Comment Out Selected Text

```
<!-- $$$@ -->
```

5.4.3 Tag for Color

```
<font color=red>$$$@</font>
```

Or:

```
<mark>$$$@</mark>
```


Magic Words are some characters that have special meanings. They will be evaluated to some pre-defined characters. For one simple example, `date` could be evaluated to the date of today.

VNote support snippets in most of the input widgets. For example, you could use `snippet` for the note name when creating a note.

`%da%` work log.md will be evaluated as `20180128 work log.md` since `%da%` is a magic word which is defined as the date of today in the form `YYYYMMDD`.

In the editor, you could type `%da%`, and then press shortcut `Ctrl+E M`, which will evaluate the word under cursor as magic word.

For example, type following words:

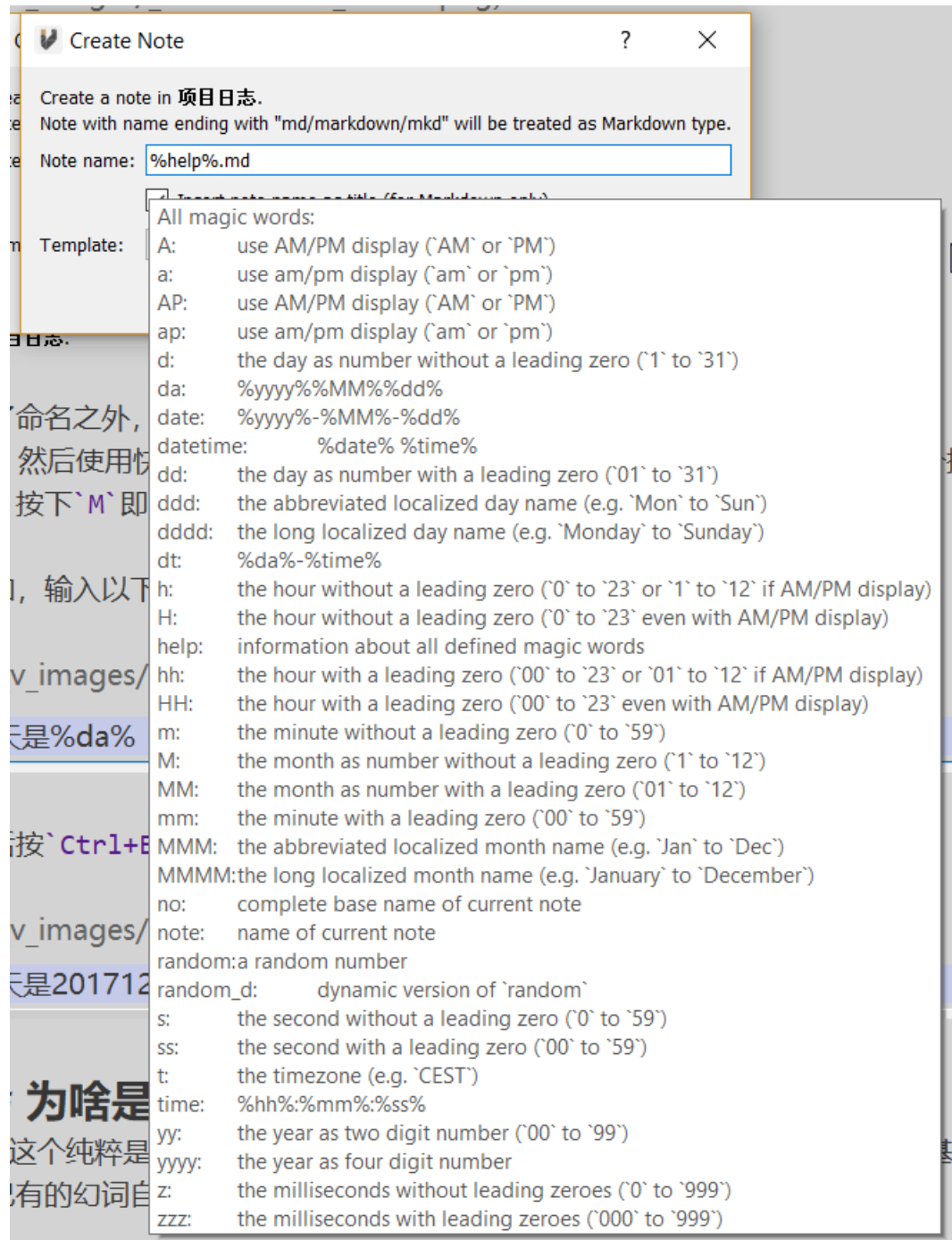
```
Today is %da%
```

Then press `Ctrl+E M`, it will be changed to:

```
Today is 20180128
```

6.1 Built-In Magic Words

VNote has defined lots of magic words. In the input dialog, type `%help%` to show a list of defined magic words.



6.2 Custom Magic Words

Edit the `vnote.ini` file in configuration folder like this:

```
[magic_words]
1\name=vnote
1\definition="vnote is a great tool! -- Written %datetime%"
```

(continues on next page)

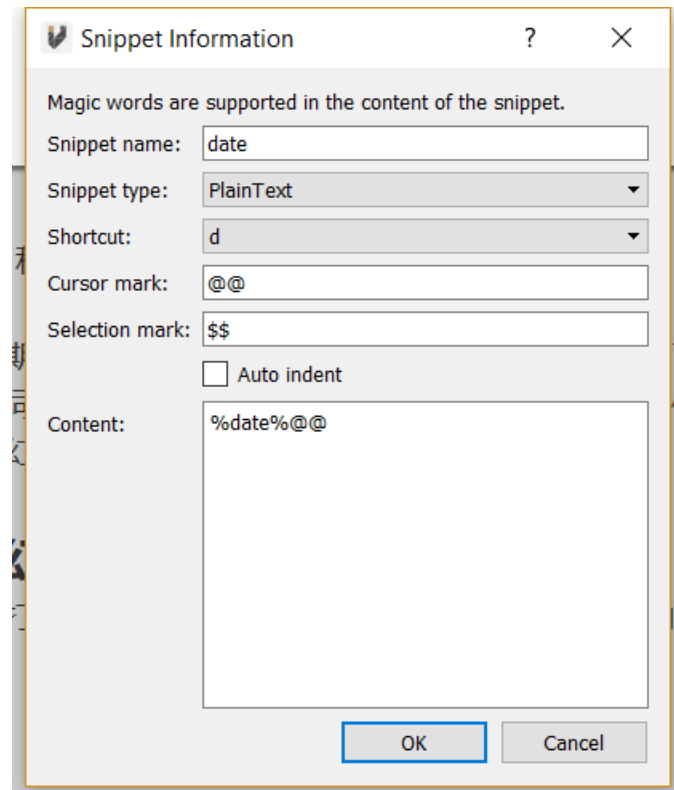
(continued from previous page)

```
2\name=hw
2\definition="hello world!"
size=2
```

Now we got two magic words `vnote` (based on another magic word `datetime`) and `hw`.

6.3 Magic Word In Snippet

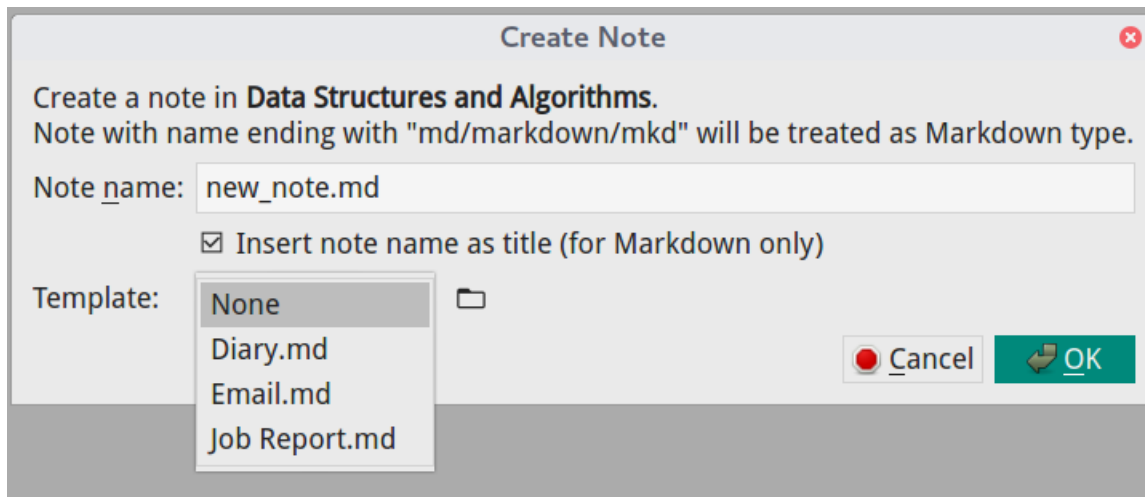
Snippet also supports magic word. Now we could define a snippet to insert current date like this:




Now in edit mode, we could hit `Ctrl+E S D` to insert current date.

VNote supports creating a note from a template.

When creating a note, you could choose one template in the dialog.



VNote stores template files in the `templates` subfolder of the configuration folders. One file corresponds to one template. You could access the template folder by clicking the “folder” icon  besides the combo box.

You could add or delete template files in the template folder via system’s file browser.

VNote supports **Magic Word** in template. For example, you could write a template like this:

```
# %no%
This is a template using **Magic Word** to insert note name as the title,
↪automatically.
```

`%no%` is a magic word which will be evaluated to the note name (without suffix). Hence if the note name is `week report.md`, then the new note will look like this:

```
# week report
This is a template using **Magic Word** to insert note name as the title,
↳automatically.
```

8.1 Themes

A **theme** specifies the look of VNote, the style of the editor and read mode, and the syntax highlight style of code block.

A theme corresponds to a folder in the `themes` subfolder in the configuration folder with structure:

```
themes/  
├── v_pure  
│   ├── arrow_dropdown.svg  
│   ├── branch_closed.svg  
│   ├── branch_end.svg  
│   ├── branch_more.svg  
│   ├── branch_open.svg  
│   ├── checkbox_checked.svg  
│   ├── checkbox_unchecked.svg  
│   ├── close_grey.svg  
│   ├── close.svg  
│   ├── down_disabled.svg  
│   ├── down.svg  
│   ├── float.svg  
│   ├── left_disabled.svg  
│   ├── left.svg  
│   ├── line.svg  
│   ├── menu_checkbox.svg  
│   ├── menu_radiobutton.svg  
│   ├── radiobutton_checked.svg  
│   ├── radiobutton_unchecked.svg  
│   ├── right_disabled.svg  
│   ├── right.svg  
│   ├── up_disabled.svg  
│   ├── up.svg  
│   └── v_pure_codeblock.css
```

(continues on next page)

(continued from previous page)

```

├── v_pure.css
├── v_pure.mdhl
├── v_pure.palette
└── v_pure.qss

```

- `v_pure.palette`: the main entry of a theme, which specifies other files and styles;
- `v_pure.qss`: file for **Qt Style Sheet**, which specifies the look of all the widgets;
- `v_pure.mdhl`: file for the style of Markdown editor, using [The Syntax of PEG Markdown Highlight Stylesheets](#);
- `v_pure.css`: file for the style of read mode;
- `v_pure_codeblock.css`: file for the style of the code block highlight in read mode, using [Highlight.js](#);

8.1.1 Palette of Theme

Let's look into the `.palette` file. The file is in INI format.

[metadata]

This section specifies other style files this theme will use.

```

; File path could be absolute path or relative path (related to this file).
; Use @color_tag to reference a style.

[metadata]
qss_file=v_pure.qss
mdhl_file=v_pure.mdhl
css_file=v_pure.css
codeblock_css_file=v_pure_codeblock.css
version=2

```

[phony]

This section is used to define **variables** for fundamental, abstract color attributes. A variable could be referenced by `@variable_name` to define another variable.

These variables are referenced by other sections, so you are free to choose and define your own ones.

```

[phony]
; Abstract color attributes.
master_fg=#F5F5F5
master_bg=#00897B
master_light_bg=#80CBC4
master_dark_bg=#00796B
master_focus_bg=#009688
master_hover_bg=#009688
master_pressed_bg=#00796B

base_fg=#222222
base_bg=#EAEAEA

```

(continues on next page)

(continued from previous page)

```
main_fg=@base_fg
main_bg=@base_bg

title_fg=@base_fg
title_bg=@base_bg

disabled_fg=#9E9E9E

content_fg=@base_fg
content_bg=@base_bg

border_bg=#D3D3D3

separator_bg=#D3D3D3

hover_fg=@base_fg
hover_bg=#D0D0D0

selected_fg=@base_fg
selected_bg=#BDBDBD

active_fg=@selected_fg
active_bg=@selected_bg

inactive_fg=@selected_fg
inactive_bg=#D3D3D3

focus_fg=@selected_fg
focus_bg=@selected_bg

pressed_fg=@base_fg
pressed_bg=#B2B2B2

edit_fg=#222222
edit_bg=#F5F5F5
edit_focus_bg=#E0F2F1
edit_focus_border=@master_bg
edit_selection_fg=@edit_fg
edit_selection_bg=@master_light_bg

icon_fg=#222222
icon_disabled_fg=@disabled_fg

danger_fg=#F5F5F5
danger_bg=#C9302C
danger_focus_bg=#D9534F
danger_hover_bg=#D9534F
danger_pressed_bg=#AC2925
```

[soft_defined]

This section define variables which will be used by VNote code. You **MUST** define these variables to make VNote looks right.

```
[soft_defined]
; VAvatar.
; The foreground color of the avatar when Captain mode is triggered.
avatar_captain_mode_fg=@master_fg
; The background color of the avatar when Captain mode is triggered.
avatar_captain_mode_bg=@master_bg

; Style of the label in Navigation mode.
navigation_label_fg=@master_fg
navigation_label_bg=@master_bg

; Style of the bubble of VButtonWithWidget.
bubble_fg=@master_fg
bubble_bg=@master_bg

; Icons' foreground.
danger_icon_fg=@danger_bg
item_icon_fg=@icon_fg
title_icon_fg=@icon_fg

; V VimIndicator.
vim_indicator_key_label_fg=@base_fg
vim_indicator_mode_label_fg=@base_fg
vim_indicator_cmd_edit_pending_bg=@selected_bg

; V TabIndicator.
tab_indicator_label_fg=@base_fg

; Html template.
template_title_flash_light_fg=@master_light_bg
template_title_flash_dark_fg=@master_bg

; Search hit items in list or tree view.
search_hit_item_fg=@selected_fg
search_hit_item_bg=@master_light_bg
```

[widgets]

This section defines variables to be used in qss file to define concrete style of different widgets. They are referenced by the qss file.

```
[widgets]
; Widget color attributes.

; QWidget.
widget_fg=@base_fg

; Separator of dock widgets.
dock_separator_bg=@border_bg
dock_separator_hover_bg=@border_bg

; Menubar.
menubar_bg=@main_bg
menubar_fg=@main_fg
menubar_item_selected_bg=@selected_bg
```

(continues on next page)

(continued from previous page)

```

; Menu.
menu_bg=@base_bg
menu_fg=@base_fg
menu_item_disabled_fg=@disabled_fg
menu_item_selected_fg=@selected_fg
menu_item_selected_bg=@selected_bg
menu_separator_bg=@separator_bg
menu_icon_fg=@icon_fg
menu_icon_danger_fg=@danger_icon_fg

```

The qss file may look like this if you are curious:

```

/* QWidget */
QWidget
{
    color: @widget_fg;
    font-family: "Hiragino Sans GB", "", "Microsoft YaHei", "", "Microsoft YaHei UI",
    ↪ "WenQuanYi Micro Hei", "", "Dengxian", "", "STXihei", "", "Liberation Sans", "Droid_
    ↪ Sans", "NSimSun", "", "SimSun", "", "Helvetica", "sans-serif", "Tahoma", "Arial",
    ↪ "Verdana", "Geneva", "Georgia", "Times New Roman";
}

QWidget[NotebookPanel="true"] {
    padding-left: 3px;
}
/* End QWidget */

/* QMainWindow */
QMainWindow {
    color: @base_fg;
    background: @base_bg;
}

QMainWindow::separator {
    background: @dock_separator_bg;
    width: 2px;
    height: 2px;
}

QMainWindow::separator:hover {
    background: @dock_separator_hover_bg;
}
/* End QMainWindow */

QMenuBar {
    border: none;
    background: @menubar_bg;
    color: @menubar_fg;
}

QMenuBar::item:selected {
    background: @menubar_item_selected_bg;
}

```

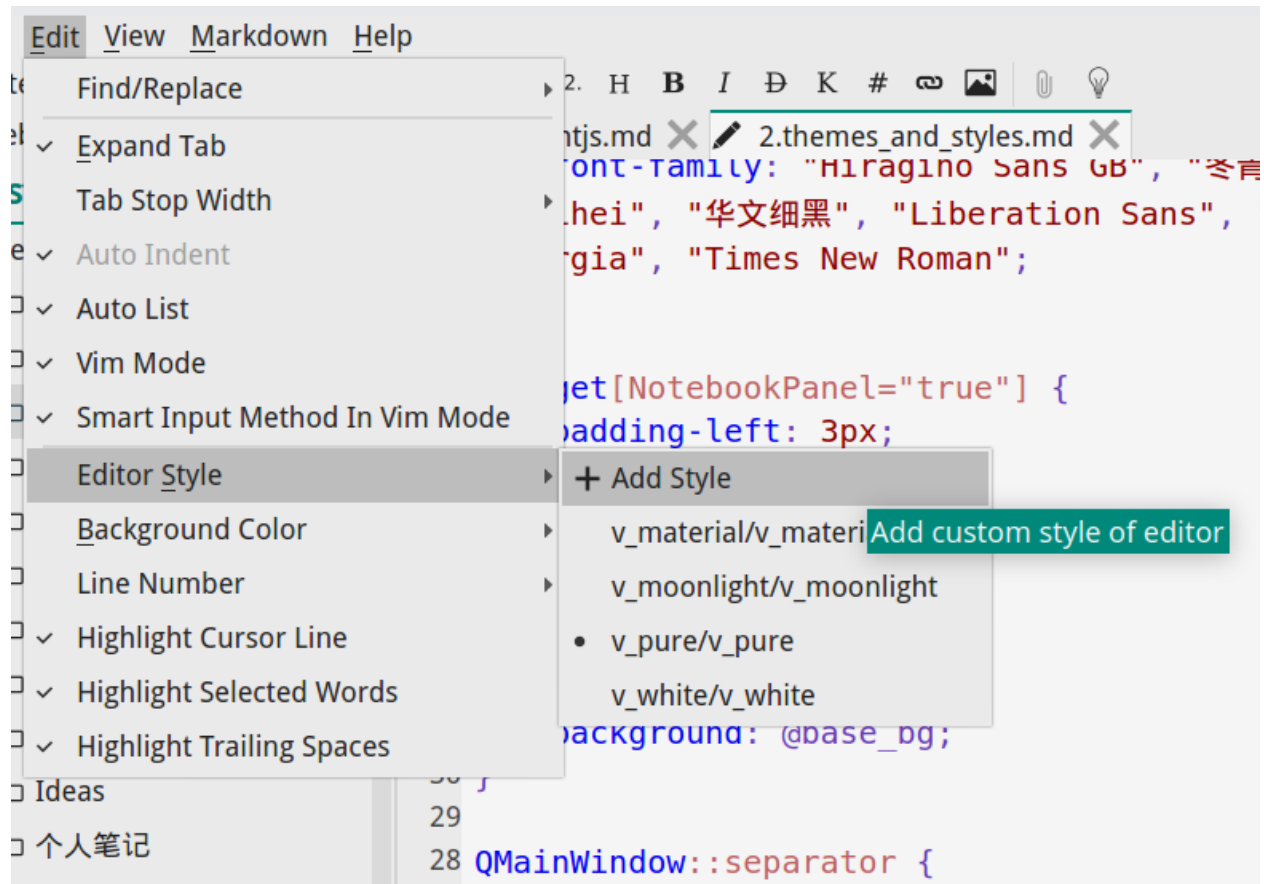
8.1.2 Custom Themes

VNote supports custom themes. Just place your theme (folder) to the `themes` folder, restart VNote and choose your theme in the `File` menu.

The best way to custom a theme is tuning a default theme. VNote will output default themes in the `themes` folder (or if not, you could download it [here](#)). Copy a default theme and rename the `palette` file. Then you could tune it to your taste. Restart of VNote is needed to let the changes take effect.

8.2 Editor Styles

Editor style is specified by a `mdhl` file. Each theme may carry a `mdhl` file. You could also apply another `mdhl` file instead of using the default one specified by the theme. Separate style file could be placed in the `styles` folder in the configuration folder. Restart of VNote is needed to scan new `mdhl` files and re-open of notes is needed to apply new style.



8.2.1 Syntax of MDHL File

The `mdhl` file adopts the [The Syntax of PEG Markdown Highlight Stylesheets](#) and expands it to support more elements.

```
# This is the default markdown styles used for Peg-Markdown-Highlight
# created by Le Tan(tamllokveer@gmail.com).
```

(continues on next page)

(continued from previous page)

```

# For a complete description of the syntax, please refer to the original
# documentation of the style parser
# [The Syntax of PEG Markdown Highlight Stylesheets](http://hasseg.org/peg-markdown-
↪highlight/docs/stylesheet_syntax.html).
# VNote adds some styles in the syntax which will be marked [VNote] in the comment.
#
# Note: Empty lines within a section is NOT allowed.
# Note: Do NOT modify this file directly. Copy it and tune your own style!

editor
# QTextEdit just choose the first available font, so specify the Chinese fonts first
# Do not use "" to quote the name
font-family: Hiragino Sans GB, , Microsoft YaHei, , Microsoft YaHei UI, WenQuanYi_
↪Micro Hei, , Dengxian, , STXihei, , Liberation Sans, Droid Sans, NSimSun, , SimSun,
↪, Helvetica, sans-serif, Tahoma, Arial, Verdana, Geneva, Georgia, Times New Roman
font-size: 12
foreground: 222222
background: f5f5f5
# [VNote] Style for trailing space
trailing-space: a8a8a8
# [VNote] Style for line number
line-number-background: eaeaea
line-number-foreground: 424242
# [VNote] Style for selected word highlight
selected-word-foreground: 222222
selected-word-background: dfdf00
# [VNote] Style for searched word highlight
searched-word-foreground: 222222
searched-word-background: 4db6ac
# [VNote] Style for searched word under cursor highlight
searched-word-cursor-foreground: 222222
searched-word-cursor-background: 66bb6a
# [VNote] Style for incremental searched word highlight
incremental-searched-word-foreground: 222222
incremental-searched-word-background: ce93d8
# [VNote] Style for color column in fenced code block
color-column-background: dd0000
color-column-foreground: ffff00
# [VNote] Style for preview image line
preview-image-line-foreground: 9575cd

editor-selection
foreground: eeeeee
background: 005fff

editor-current-line
background: c5cae9
# [VNote] Vim insert mode cursor line background
vim-insert-background: c5cae9
# [VNote] Vim normal mode cursor line background
vim-normal-background: e0e0e0
# [VNote] Vim visual mode cursor line background
vim-visual-background: bbdefb
# [VNote] Vim replace mode cursor line background
vim-replace-background: f8bbd0

H1

```

(continues on next page)

(continued from previous page)

```
foreground: 222222
font-style: bold
font-size: +8

VERBATIM
foreground: 673ab7
font-family: Consolas, Monaco, Andale Mono, Monospace, Courier New
# [VNote] Codeblock sylte from HighlightJS (bold, italic, underlined, color)
# The last occurence of the same attribute takes effect
# Could specify multiple attribute in one line
hljs-comment: 6c6c6c
hljs-keyword: 0000ee
hljs-attribute: 0000ee
hljs-selector-tag: 0000ee
hljs-meta-keyword: 0000ee
hljs-doctag: 0000ee
hljs-name: 0000ee
hljs-type: 880000
hljs-string: 880000
hljs-number: 880000
hljs-selector-id: 880000
hljs-selector-class: 880000
hljs-quote: 880000
hljs-template-tag: 880000
hljs-deletion: 880000
hljs-title: bold, 880000
hljs-section: bold, 880000
hljs-regexp: bc6060
hljs-symbol: bc6060
hljs-variable: bc6060
hljs-template-variable: bc6060
hljs-link: bc6060
hljs-selector-attr: bc6060
hljs-selector-pseudo: bc6060
hljs-literal: af00d7
hljs-built_in: 008700
hljs-bullet: 008700
hljs-code: 008700
hljs-addition: 008700
hljs-meta: 1f7199
hljs-meta-string: 4d99bf
hljs-emphasis: italic
hljs-strong: bold
```

The syntax highlight of the code blocks in edit mode is specified by the VERBATIM element.

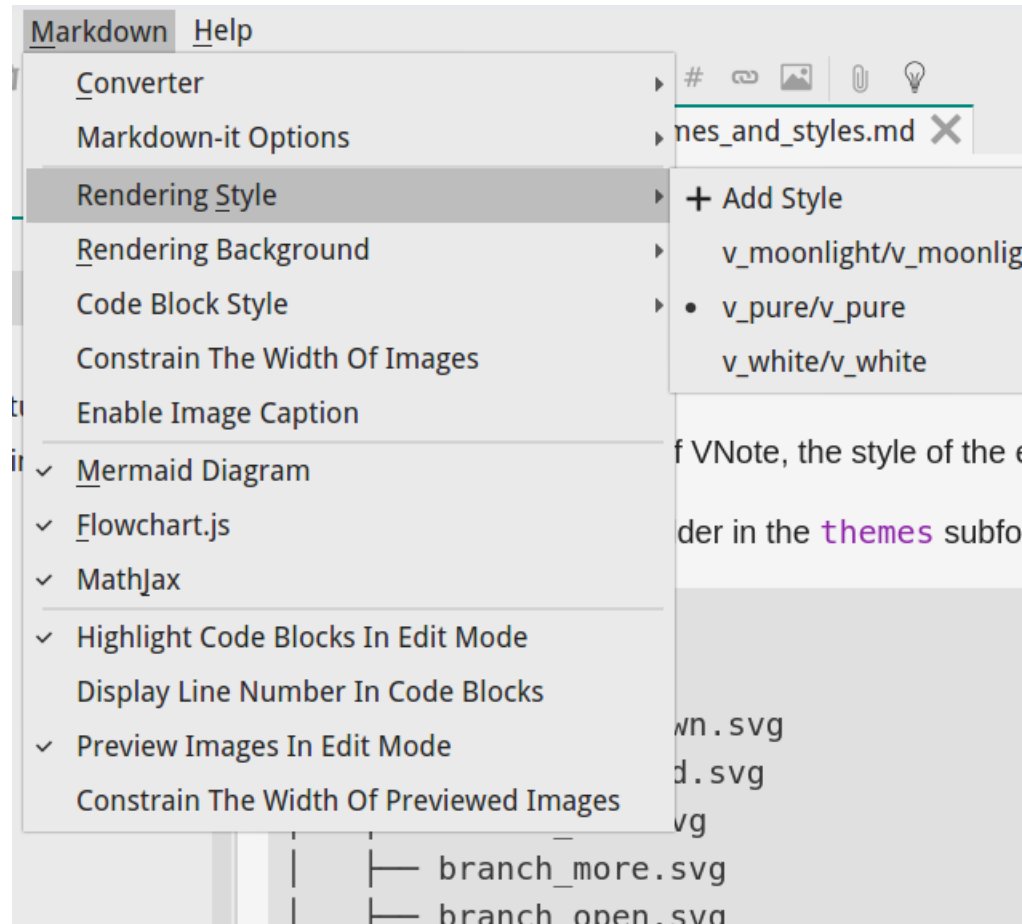
For example, if you want to change the font size of the code block in edit mode, you may need to add the following line under VERBATIM element:

```
font-size: -2
```

8.3 Rendering Styles

Rendering style in read mode is specified by a `css` file. Each theme may carry a `css` file. You could also apply another `css` file instead of using the default one specified by the theme. Separate style file could be placed in the

`styles` folder in the configuration folder. Restart of VNote is needed to scan new `css` files and re-open of notes is needed to apply new style.



8.4 Code Block Rendering Styles

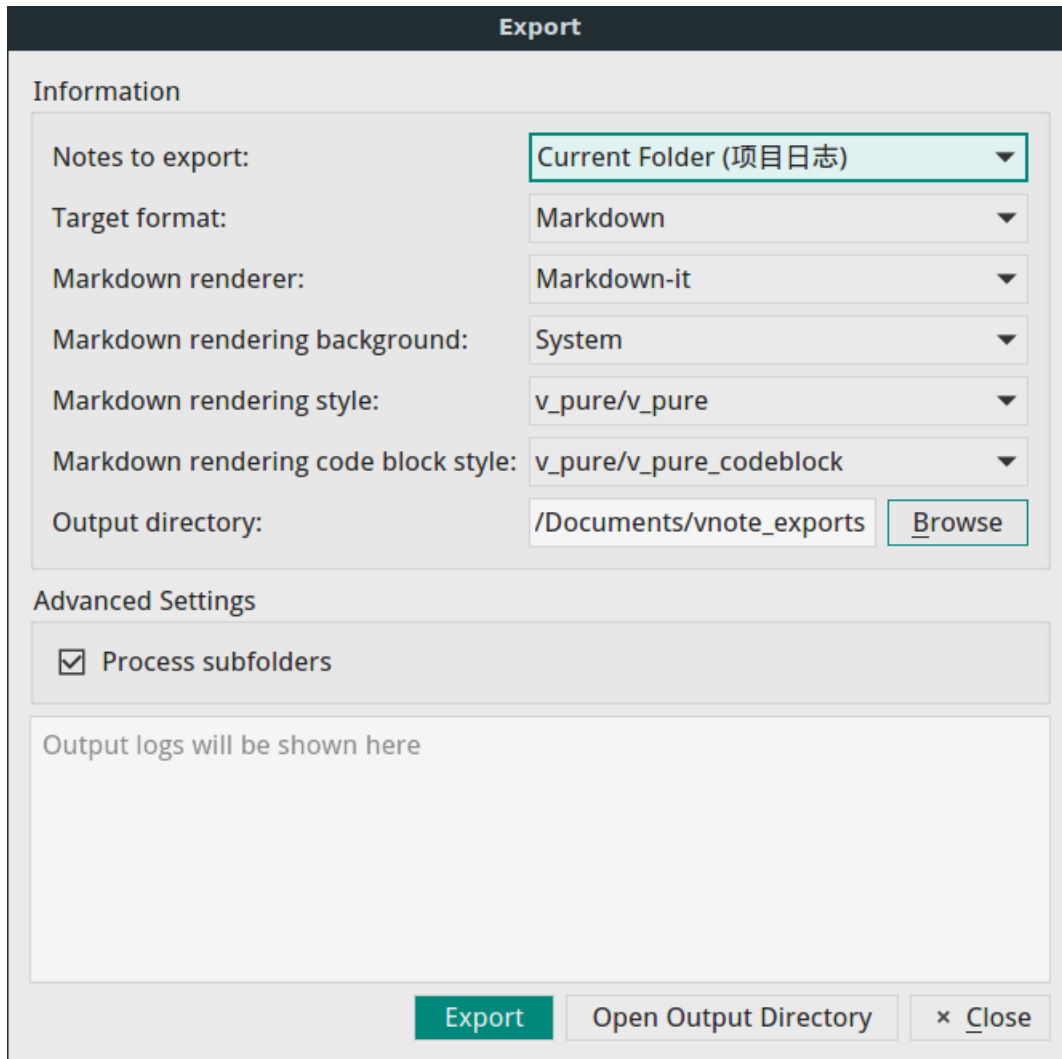
Rendering style in read mode is specified by a `css` file. Each theme may carry a `css` file. You could also apply another `css` file instead of using the default one specified by the theme. Separate style file could be placed in the `styles/codeblock_styles` folder in the configuration folder. Restart of VNote is needed to scan new `css` files and re-open of notes is needed to apply new style.

This file is used by the **highlight.js** renderer engine. You could download existing style files from [Highlight.js Github](#).

CHAPTER 9

Export

VNote supports rich and extensible export.



9.1 Sources

There are four sources to select which notes to export.

- **Current Note**
 - Export current note;
- **Current Folder**
 - Export all the notes within current folder;
 - Support export subfolders recursively;
- **Current Notebook**
 - Export all the notes of current notebook;
- **Cart**
 - Cart is a tool to collect random notes for further processing.

9.2 Options

We could specify **Markdown renderer**, **Markdown rendering background**, **Markdown rendering style**, and **Markdown rendering code block style** for the export. For example, we could specify a *transparent* background when we export PDFs.

9.3 Formats

9.3.1 Markdown

We could export a Markdown note as Markdown format. For a note named `vnote.md`, VNote will create a folder named `vnote_md`, then copy the `vnote.md` to this folder, then copy local images of this note, then copy the attachments of this note. After this, we could compress this folder and share it.

9.3.2 HTML

When exporting a note as HTML format, VNote will first render the Markdown note and then output the HTML as a file. Advanced settings of HTML format includes:

- **Embed CSS styles**
 - Embed CSS styles in the exported HTML file.
- **Complete page**
 - When checked, VNote will also export related images besides the HTML file. If not checked, you may lose local images.
- **MIME HTML**
 - Export the MIME HTML format.

9.3.3 PDF

VNote supports exporting notes as PDF files. Page size and margins are configurable. There are two engines to choose to do the export.

Built-In Engine

The built-in engine works better with the content, but lacks the bookmark in the generated PDF file.

wkhtmltopdf

`wkhtmltopdf` is a third-party tool to convert HTML to PDF. You need to install this extra tool which will be used by VNote. When using this engine, VNote will first convert the Markdown notes to HTML format, and then call `wkhtmltopdf` with the generated HTML files as input.

Advanced Settings

| | | | |
|---------------------|---|-------------------|---|
| Page layout: | A4, Portrait | Settings | <input type="checkbox"/> Enable Table Of Contents |
| | <input checked="" type="checkbox"/> Use wkhtmltopdf | | Download wkhtmltopdf |
| wkhtmltopdf path: | wkhtmltopdf | | Browse |
| Title: | Empty to use the na... | Output file name: | Empty to use the name o... |
| Page number: | None | | <input checked="" type="checkbox"/> Enable background |
| Additional options: | Use " to enclose options containing spaces | | |

Advanced settings includes:

- **wkhtmltopdf path**
 - Path of the wkhtmltopdf executable.
- **Title**
 - Title of the generated PDF. It is only valid when **All In One** is enabled.
- **Output file name**
 - The name of the generated PDF file. It is only valid when All In One is enabled.
- **Page Number**
 - Whether append page number to every page and the position of it.
- **Enable background**
 - Whether enable the background of the HTML.
- **Additional options**
 - Additional global options which will be passed to wkhtmltopdf. For available options, please reference the documentation of wkhtmltopdf.

9.3.4 PDF (All In One)

When choosing this format, VNote will utilize the wkhtmltopdf tool to convert multiple Markdown notes into **one** PDF file. To make this work well, a good practice in using VNote is to turn on **Heading Sequence** starting from level 2 and insert the title as level 1 heading for each note.

9.3.5 Custom

The **Custom** format enable users to choose whatever tool they like to process notes.

Advanced Settings

Source format: Output suffix:

Enable All In One

Output file name: Input directories separator:

%0 for the input file; **%1** for the output file; **%2** for the rendering CSS style file; **%3** for the input file directory.

```
pandoc --resource-path=.: "%3" --css="%2" -s -o "%1" "%0"
```

Advanced settings:

- **Source format**
 - VNote supports using Markdown or HTML format as the inputs to your tool.
- **Output suffix**
 - The suffix of the output file, such as docx, pdf, or ppt.
- **Enable All In One**
 - Whether put multiple notes as the input once to your tool. They are separated by spaces.
- **Output file name**
 - The file name of the output. It is only valid when All In One is enabled.
- **Input directories separator**
 - VNote could pass the directories of the input notes to your tool. You could use these directories as the resource folders. This option specifies the separator to concatenate these directories.
- **Command**
 - The command executed by VNote. Some special place holder you could use:
 - * %0: the input files;
 - * %1: the output file;
 - * %2: the rendering CSS style file to be used;
 - * %3: the input file directories;
 - * %4: the rendering code block CSS style file to be used;

An example to use `pandoc` to export notes:

```
pandoc --resource-path=.: "%3" --css="%2" --css="%4" -s -o "%1" "%0"
```

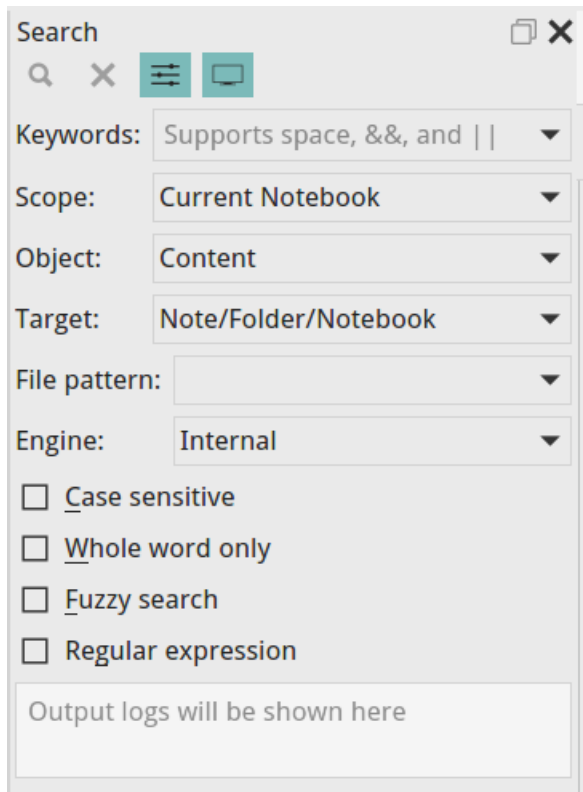
Then if the output suffix is pdf, it will generate the PDF file; if it is docx, it will generate the DOCX file.

CHAPTER 10

Full-Text Search

VNote provides powerful built-in full-text search to search your notes.

Ctrl+E C or Ctrl+Alt F to activate the **Search** dock widget. You could also activate it through the **Edit** or **View** menu.



10.1 Keywords

- Specify the keywords to search for;
- Support `&&` and `||` for AND and OR logics, such as `markdown && vnote`;
- Space-separated keywords mean AND, such as `markdown vnote`;

10.1.1 Magic Switch

VNote supports **Magic Switch** in the keywords to turn on/off some options of the search:

- `\f` or `\F`: disable or enable **Fuzzy Search**;
- `\c` or `\C`: case insensitive or sensitive;
- `\r` or `\R`: disable or enable **Regular Expression**;
- `\w` or `\W`: disable or enable **Whole Word Only**;

Example: `vnote \C \W` to search `vnote` with case-insensitive and whole-word-only.

10.2 Scope

There are four scopes to constrain the search:

- `Opened Notes`;
- `Current Folder`;
- `Current Notebook`;
- `All Notebooks`;

10.3 Object

Specify whether the search should be executed against the **Content** or the **Name**.

10.4 Target

We could search among:

- `Note`;
- `Folder`;
- `Notebook`;
- `Note/Folder/Notebook`;

10.5 File Pattern

We could specify the file pattern to filter the files we are interested in, such as `*.md` to search only Markdown files.

Frequently Asked Questions

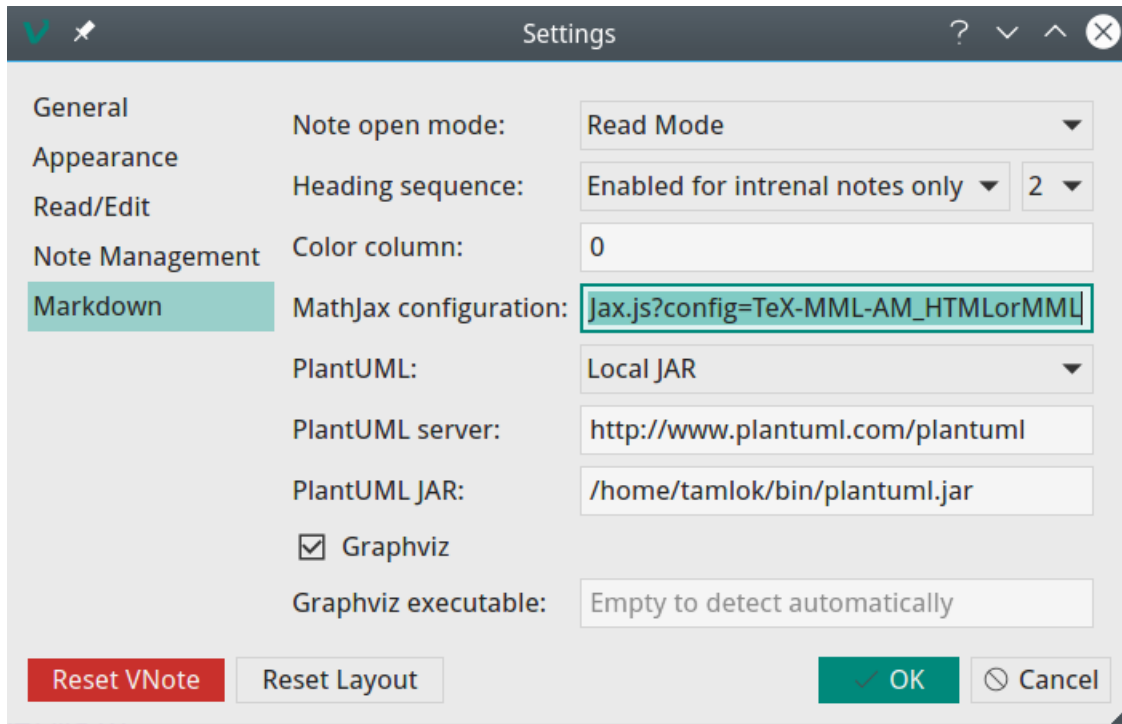
11.1 How to specify customized MathJax script?

VNote supports a configuration `[web]/mathjax_javascript` which specifies the location of the MathJax and its configuration to use. The default value may look like this:

```
[web]
; Location and configuration for Mathjax
mathjax_javascript=https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.2/MathJax.js?
↔config=TeX-MML-AM_CHTML
```

You could download the MathJax ([howto](#)) and specify a local copy of the MathJax to use, like `mathjax_javascript="file:///home/tamlok/Downloads/MathJax-2.7.2/MathJax.js?config=TeX-MML-AM_CHTML"`.

You could also specify customized script in GUI settings dialog:



Notes:

- Need to prepend `file://` to the local path;
- Using local path may get the fonts wrong. I think a better way is to run a light HTTP server locally.

11.2 How to make VNote more “portable”?

VNote will read and store configuration files in the common directory of the operating system to store application data. If you prefer to put these configuration files along with the VNote executable, you could copy or create the `vnote.ini` file in the directory containing VNote executable. VNote will first try to read `vnote.ini` from where the executable exists.

However, please notice that VNote will store the notebook’s path in the configuration file, so it may read a wrong path and complain about errors if you copy VNote to another computer.

11.3 How to draw diagrams using Mermaid or Flowchart.js?

1. Enable Mermaid Diagram and Flowchart.js in the Markdown menu;
2. Re-open opened tabs to let it take effect;
3. For Mermaid, write a fenced code block specified with `language mermaid`;
4. For Flowchart.js, write a fenced code block specified with `language flowchart` or `flow`;

Mermaid example:

```
```mermaid
graph TD;
```

(continues on next page)



(continued from previous page)

```
A-->B;
A-->C;
B-->D;
C-->D;
...

```

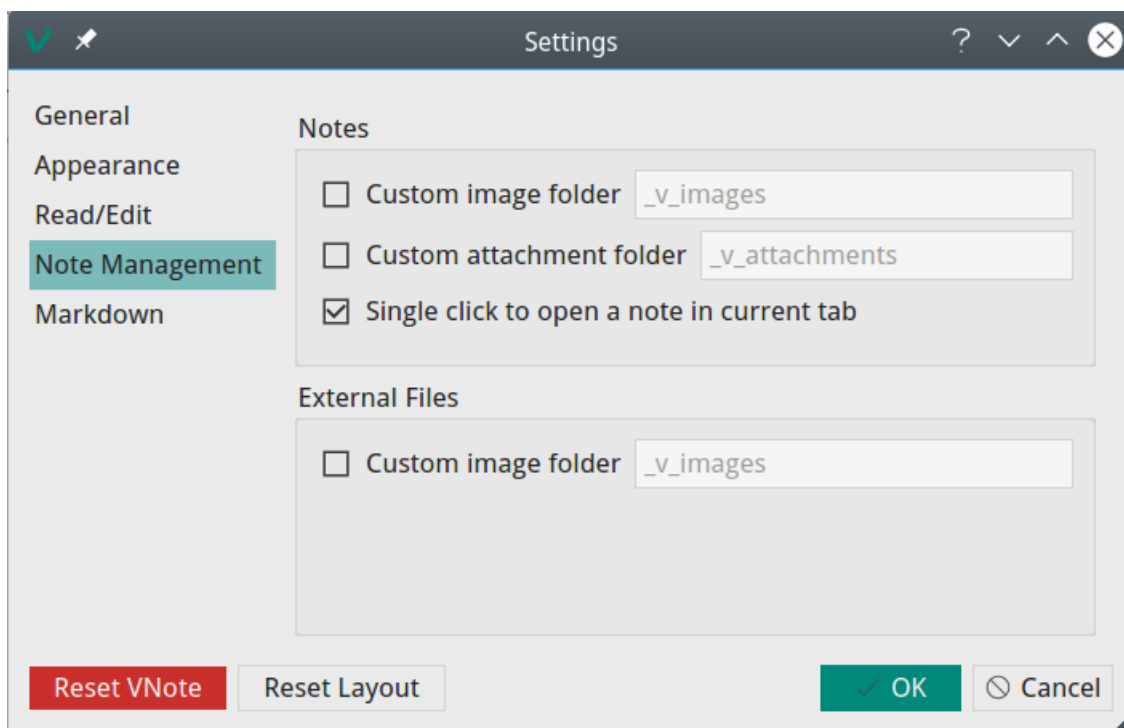
Flowchart.js example:

```
```flowchart
st=>start: Start:>http://www.google.com[blank]
e=>end:>http://www.google.com
op1=>operation: My Operation
sub1=>subroutine: My Subroutine
cond=>condition: Yes
or No?:>http://www.google.com
io=>inputoutput: catch something...

st->op1->cond
cond(yes)->io->e
cond(no)->sub1(right)->op1
...
```

```

## 11.4 How to open a note in another tab?



When Single click to open a note in current tab is enabled, single-click in the note list to open the note in current tab, while double-click to open it in a new tab.

## 11.5 Interface is freezed

## 11.6 Cursor is not visible in edit mode

## 11.7 Crash when opening a note

## 11.8 Menu does not work in fullscreen mode

All the above 4 issues may be caused by the driver of the display card. Steps to solve it:

1. On Windows, try to update the display card driver;
2. Try to schedule VNote to run with integrated display card;
3. If not works, try to declare a system environment variable named `QT_OPENGL` and set its value to `desktop`, or `angle`, or `software`.

Restart of VNote is needed after the action.