

Use Emacs denote package to write in Notty

John Doe · August 19, 2025

If you use Emacs, Notty is accompanied by an Emacs Lisp package providing the integration of Notty and the Denote package. The following is an example of configuration.

```
(use-package denote
  :bind (("C-c n n" . denote))
  :config
  (setq denote-directory (expand-file-name "~/Documents/notes/typ/")))

(use-package typst-ts-mode)

(use-package denote-notty
  :load-path "/path/to/notty/directory/"
  :after (denote typst-ts-mode)
  :demand t
  :bind (:map typst-ts-mode-map
    ("C-c n b" . denote-notty-backlinks)
    ("C-c n c" . denote-notty-contexts)
    ("C-c n t" . denote-notty-transclude)
    ("C-c n l" . denote-link)
    ("C-c n L" . denote-add-links)
    ("C-c n q c" . denote-query-contents-link) ; create link that
triggers a grep
    ("C-c n q f" . denote-query-filenames-link) ; create link that
triggers a dired
    ;; Note that `denote-rename-file' can work from any context, not
just
    ;; Dired buffers. That is why we bind it here to the `global-
map'.
    ("C-c n r" . denote-rename-file)
    ("C-c n R" . denote-rename-file-using-front-matter))
  :config
  (push denote-notty-file-type denote-file-types)
  (defun +denote-sluggify-signature (str)
    "Make STR an appropriate slug but allowing uppercase letters."
    (denote-slug-put-equals
      (replace-regexp-in-string "[!@#$%^&*()+'\"?;~`'\"'\"/\"-]*" ""
str)))
  (setq denote-file-name-slug-functions
    '((title . denote-sluggify-title)
```

```
(signature . +denote-sluggify-signature)
(keyword . denote-sluggify-keyword)))
```