

Environment

Editor

The following environment variables make sense when using *Emacs*:

```
export EDITOR="/usr/bin/emacsclient -a /usr/bin/emacs"
export ALTERNATE_EDITOR="/usr/bin/emacs"
```

Put it into files like `~/.bashrc`, `~/.zshrc` and/or `~/.xsession` (see [Environmental Settings](#) or [MacBook Software Installation](#) Wiki page).

Shell Colors

To have shell coloring in *Shell Mode* the following code might be helpful in `~/.bashrc`, `~/.zshrc` or equivalent.

```
if [ "${INSIDE_EMACS/*comint*/comint}" == "comint" -o \
    "${INSIDE_EMACS/*tramp*/tramp}" == "tramp" ] ; then
    export GIT_PAGER=""
    export TERM=ansi
fi
```

SUDO Shell Commands (Linux)

If you want to use *sudo* with *Emacs Lisp* function *shell-command* then it possibly does **not** work in a graphical environment like *GTK+*.¹⁾ For this case i created a shell-script `askpass.sh`, with the following content:

```
#!/bin/sh
gksudo --print-pass --preserve-env --description "sudo password" 2>/dev/null
```

Then set environment variable

```
export SUDO_ASKPASS="$HOME/bin/askpass.sh"
```

and implement your shell command:

```
(shell-command (format "sudo -E -A my-sudo-cmd %s" my-sudo-execdir))
```



Another source of proposals and solutions might be the [EmacsWiki](#).

SUDO Shell Commands (macOS)

On macOS the same principles hold true. But one proposed solution is to use `pinentry-mac` (installed via *Homebrew* together with this [ss-askpass](#) wrapper).

¹⁾

Normally *Emacs* handles password prompt/input via `comint-watch-for-password-prompt` (see the [Shell Mode manual](#)).

From:

<https://wiki.rho62.de/> - rho62 Wiki

Permanent link:

<https://wiki.rho62.de/doku.php?id=development:emacs:shell-env&rev=1771256608>

Last update: 2026/02/16 16:43

