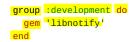


Libnotify

- Runs on Linux, FreeBSD, OpenBSD and Solaris
- Supports Libnotify

The libnotify gem supports the Gnome libnotify notification daemon, but it can be used on other window managers as well. You have to install the libnotify-bin package with your favorite package manager.

To use libnotify you have to add it to your Gemfile and run bundler:



RootFile & ChangeFileLines

If you are unable to build the libnotify gem on your system, Guard also has a built in notifier - notifysend - that shells out to the notify-send utility that comes with libnotify-bin.

Please note that NotifyOSD, which is the default notification UI on Ubuntu and possibly other systems, ignores the timeout parameter for libnotify.

Notifu

- Runs on Windows
- Supports Notifu

The rb-notifu gem supports Windows system tray notifications.

To use rb-notifu you have to add it to your Gemfile and run bundler:

```
group :development do
gem 'rb-notifu'
end
```

ChangeFileLines (Windows)

Terminal Notifier

• Runs on Mac OS X 10.8 only

The terminal-notifier-guard sends notifications to the OS X Notification Center.

To use terminal-notifier-guard you have to add it to your Gemfile and run bundler:

```
group :development do
gem 'terminal-notifier-guard'
```

ChangeFileLines (Mac)

Terminal Title

• Runs in every terminal supporting XTerm escape sequences to set the window title.

different editors

Clone this wiki locally

https://github.com/guard/gu

Emacs

 Runs on any platform with Emacs + emacsclient (http://www.emacswiki.org/emacs/ /EmacsClient)

TMux

• To use TMux notifications, you have to start Guard within a TMux session.

The TMux notifier will color the background of the left part of the status bar indicating the status of the notifications. Optionally you can set :display_message => true to display the Guard notification as 'display-message' notification.

The way these messages are formatted is configurable.

```
# Guardfile
notification :tmux,
 display_message: true,
 timeout: 5, # in seconds
 default_message_format: '%s >> %s',
 # the first %s will show the title, the second the message
 # Alternately you can also configure *success_message_format*,
 # *pending_message_format*, *failed_message_format*
 line_separator: , # since we are single line we need a separator
 color_location: 'status-left-bg', # to customize which tmux element will ch
 # Other options:
 default_message_color: 'black',
                                            DeclareVariables
 success: 'colour150',
 failure: 'colour174',
                                            RootFile
 pending: 'colour179',
 # Notify on all tmux clients
 display_on_all_clients: false,
```

The color location option can also take an array:

```
color_location: %w[status-left-bg pane-active-border-fg pane-border-fg]
```

The result will be for RSpec using example above

```
RSpec >> 15 test, 0 failures > in 0.002 sec
```

You can use nice powerline chars here if you have that configured.

You can get the message history by using Ctrl+b ~ (where Ctrl+b is your key to activate TMux).

File

You can also have Guard write notifications to a file. Each notification will overwrite
the file. This allows other commands to be run based on the status of other guard
commands.

Example:

```
# Guardfile
notification :file, path: '.guard_result'

guard :shell do
  watch '.guard_result' do
    if File.read('.guard_result').lines.first.strip == 'failed'
        # ...
    end
end
end
```

ChangeFileLines

Configuration:

```
# Guardfile
notification :file;
path: '.guard_result', # Required, no default
format: ''result: %s\ntitle: %s\nmessage: %s\n'' # Default: ''%s\n%s\n''s\n'''
```

This wiki and the Guard README document contains a lot of information, please take your time and read these instructions carefully.

If you run into any trouble, you may start by understanding how Guard works.

We provide detailed changes for each Guard release.

Be sure to read the CONTRIBUTING guidelines before reporting a new Guard issue or open a pull request.

If you have any questions about the Guard usage or want to share some information with the Guard community, please go to one of the following places:

- Google+ community
- Google group
- StackOverflow
- IRC channel #guard (irc.freenode.net) for chatting

 $@\ 2015\ GitHub,\ Inc.\quad Terms\quad Privacy\quad Security\quad Contact\\$



Status API Training Shop Blog About