

---

# **notify Documentation**

***Release***

**Alisue**

February 22, 2016



<b>1</b>	<b>notify</b>	<b>1</b>
1.1	Installation . . . . .	1
1.2	Usage . . . . .	1
<b>2</b>	<b>API documentation</b>	<b>3</b>
2.1	notify Package . . . . .	3
2.1.1	notify Package . . . . .	3
2.1.2	arguments Module . . . . .	3
2.1.3	compat Module . . . . .	3
2.1.4	conf Module . . . . .	4
2.1.5	console Module . . . . .	4
2.1.6	executor Module . . . . .	4
2.1.7	mailer Module . . . . .	5
2.1.8	notifier Module . . . . .	5
2.1.9	wizard Module . . . . .	6
2.1.10	Subpackages . . . . .	6
	tests Package . . . . .	6
	utils Package . . . . .	6
<b>3</b>	<b>Indices and tables</b>	<b>9</b>
	<b>Python Module Index</b>	<b>11</b>



---

## notify

---

Notify process termination via email. It support Python 2.4 and later (Python 2.4 and Python 2.5 was manually tested, so it might not work).

### 1.1 Installation

Use `pip` like:

```
$ pip install notify
```

Additionally, if you would like to use a mail user agent which requires authentication, installing `keyring` is strongly recommended:

```
$ pip install keyring
```

### 1.2 Usage

1. Running `notify` with following command will start setup wizard first time.

```
$ notify
```

2. Follow the setup wizard instruction
3. Check with the following command (Only for Unix system)

```
$ notify --check
```

4. Use `notify` like

```
$ notify really_havy_process -a -b --options  
$ notify -t different@address.com really_havy_process -a -b --options
```



---

## API documentation

---

### 2.1 notify Package

#### 2.1.1 notify Package

#### 2.1.2 arguments Module

`notify.arguments.parse_arguments` (*args*, *config*)

Parse specified arguments via config

**Parameters** *args* : list

Command line arguments

**config** : object

ConfigParser instance which values are used as default values of options

**Returns** *list* : arguments, options

options indicate the return value of ArgumentParser and arguments indicate the execution command/arguments

`notify.arguments.split_arguments` (*args*)

Split specified arguments to two list.

This is used to distinguish the options of the program and execution command/arguments.

**Parameters** *args* : list

Command line arguments

**Returns** *list* : options, arguments

options indicate the optional arguments for the program and arguments indicate the execution command/arguments

#### 2.1.3 compat Module

Compatibility module

## 2.1.4 conf Module

`notify.conf.config_to_options` (*config*)  
Convert ConfigParser instance to argparse.Namespace

**Parameters** `config` : object  
A ConfigParser instance

**Returns** object  
An argparse.Namespace instance

`notify.conf.create_default_config` ()  
Create default ConfigParser instance

`notify.conf.get_user_config_filename` (*appname='notify'*)  
Get user config filename.

It will return operating system dependent config filename.

**Parameters** `appname` : string  
An application name used for filename

**Returns** string  
A filename of user configuration.

## 2.1.5 console Module

`notify.console.main` (*args=None*)

## 2.1.6 executor Module

A terminal command executor module

`notify.executor.call` (*args*)  
Call terminal command and return `exit_code` and `stdout`

**Parameters** `args` : list  
A command and arguments list

**Returns** list : [`exit_code`, `stdout`]  
`exit_code` indicate the exit code of the command and `stdout` indicate the output of the command

`notify.executor.from_unicode` (*x, e*)

`notify.executor.get_command_str` (*args*)  
Get terminal command string from list of command and arguments

**Parameters** `args` : list  
A command and arguments list (unicode list)

**Returns** str  
A string indicate terminal command



## 2.1.7 mailer Module

Mailer module

`notify.mailer.create_message` (*from\_addr, to\_addr, subject, body, encoding=None*)  
Create message object for sending email

**Parameters** `from_addr` : string

An email address used for 'From' attribute

`to_addr` : string

An email address used for 'To' attribute

`subject` : string

An email subject string

`body` : string

An email body string

`encoding` : string

An email encoding string (Default: utf8)

**Returns** object

An instance of `email.mime.text.MIMEText`

`notify.mailer.send_email` (*msg, host='localhost', port=25, username=None, password=None*)  
Send an email (via TLS when username and password are specified)

**Parameters** `msg` : object

An instance of `MIMEText`. Create this with `create_message()` function.

`host` : string

A mail user agent host name (Default: localhost)

`port` : int

A mail user agent port number (Default: 25)

`username` : string

A username string used to login MUA via TLS authentication

`password` : string

A password string used to login MUA via TLS authentication

`debug` : boolean

True for displaying debug messages

## 2.1.8 notifier Module

`notify.notifier.call_and_notificate` (*args, opts*)  
Execute specified arguments and send notification email

**Parameters** `args` : list

A execution command/arguments list

**opts** : object  
A option instance

## 2.1.9 wizard Module

A notify setup wizard module

**exception** `notify.wizard.ValidationError`  
Bases: `exceptions.Exception`

`notify.wizard.console_input` (*default, validation=None, allow\_empty=False*)  
Get user input value from stdin

**Parameters default** : string

A default value. It will be used when user input nothing.

**validation** : callable

A validation function. The validation function must raise an error when validation has failed.

**Returns** string or any

A user input string or validated value

`notify.wizard.setup_wizard` (*config*)

`notify.wizard.validate_email` (*x*)

`notify.wizard.validate_int` (*x*)

## 2.1.10 Subpackages

### tests Package

`test_conf` Module

`test_executor` Module

### utils Package

`plaintext_keyring` Module

Plaintext keyring module

**class** `notify.utils.plaintext_keyring.PlaintextKeyring` (*filename=None*)  
Bases: `object`

#### Methods

`delete_password` (*service\_name, username*)

`get_password` (*service\_name, username*)

`set_password` (*service\_name, username, password*)

### **terminalmsg Module**

`notify.utils.terminalmsg.terminalmsg(paragraphs, wallchar='*')`

### **terminalsize Module**

`notify.utils.terminalsize.get_terminal_size()`

`getTerminalSize()` - get width and height of console - works on linux,os x,windows,cygwin(windows) originally retrieved from: <http://stackoverflow.com/questions/566746/how-to-get-console-window-width-in-python>



---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`



## n

- notify, 3
- notify.arguments, 3
- notify.compat, 3
- notify.conf, 4
- notify.console, 4
- notify.executor, 4
- notify.mailer, 5
- notify.notifier, 5
- notify.utils.plaintext\_keyring, 6
- notify.utils.terminalmsg, 7
- notify.utils.terminalsize, 7
- notify.wizard, 6





**C**

call() (in module notify.executor), 4  
call\_and\_notificate() (in module notify.notifier), 5  
config\_to\_options() (in module notify.conf), 4  
console\_input() (in module notify.wizard), 6  
create\_default\_config() (in module notify.conf), 4  
create\_message() (in module notify.mailer), 5

**D**

delete\_password() (notify.utils.plaintext\_keyring.PlaintextKeyring  
method), 6

**F**

from\_unicode() (in module notify.executor), 4

**G**

get\_command\_str() (in module notify.executor), 4  
get\_password() (notify.utils.plaintext\_keyring.PlaintextKeyring  
method), 6  
get\_terminal\_size() (in module notify.utils.terminalsizesize),  
7  
get\_user\_config\_filename() (in module notify.conf), 4

**M**

main() (in module notify.console), 4

**N**

notify (module), 3  
notify.arguments (module), 3  
notify.compat (module), 3  
notify.conf (module), 4  
notify.console (module), 4  
notify.executor (module), 4  
notify.mailer (module), 5  
notify.notifier (module), 5  
notify.utils.plaintext\_keyring (module), 6  
notify.utils.terminalsizesize (module), 7  
notify.utils.terminalsizesize (module), 7  
notify.wizard (module), 6

**P**

parse\_arguments() (in module notify.arguments), 3  
PlaintextKeyring (class in notify.utils.plaintext\_keyring),  
6

**S**

send\_email() (in module notify.mailer), 5  
set\_password() (notify.utils.plaintext\_keyring.PlaintextKeyring  
method), 6  
split\_arguments() (in module notify.wizard), 6  
split\_arguments() (in module notify.arguments), 3

**T**

termianlmsg() (in module notify.utils.terminalsizesize), 7

**V**

validate\_email() (in module notify.wizard), 6  
validate\_int() (in module notify.wizard), 6  
ValidationError, 6