
skaff Documentation

Release 0.9

Jiahui Xie

June 07, 2016

1	skaff	3
1.1	skaff package	3
1.1.1	Submodules	3
1.1.2	skaff.cli module	3
1.1.3	skaff.clitools module	3
1.1.4	skaff.config module	4
1.1.5	skaff.core module	5
1.1.6	Module contents	6
2	Indices and tables	7
	Python Module Index	9

Contents:

1.1 skaff package

1.1.1 Submodules

1.1.2 skaff.cli module

Main command line driver program for skaff.

```
class skaff.cli.SmartFormatter (*args, **kw)
    Bases: argparse.HelpFormatter
```

You can only specify one formatter in standard argparse, so you cannot both have pre-formatted description (RawDescriptionHelpFormatter) and ArgumentDefaultsHelpFormatter. The SmartFormatter has sensible defaults (RawDescriptionFormatter) and the individual help text can be marked (`help="Rl"`) for variations in formatting. Version string is formatted using `_split_lines` and preserves any line breaks in the version string.

```
skaff.cli.main()
    Parses and validates command line option flags, then invoke 'skaff()'.
```

1.1.3 skaff.clitools module

A suite of command line based tools.

```
class skaff.clitools.ANSIColor
    Bases: object

    BLUE = '\x1b[34m'
    BOLD = '\x1b[1m'
    CYAN = '\x1b[36m'
    GREEN = '\x1b[32m'
    KHAKI = '\x1b[1;33m'
    MAGENTA = '\x1b[35m'
    PURPLE = '\x1b[1;35m'
    RED = '\x1b[31m'
    RESET = '\x1b[0m'
```

```
YELLOW = '\x1b[33m'
```

```
exception skaff.clitools.TimeoutError
```

```
    Bases: Exception
```

```
skaff.clitools.single_keypress_read()
```

```
    Waits for a single keypress on stdin.
```

This is a silly function to call if you need to do it a lot because it has to store stdin's current setup, setup stdin for reading single keystrokes then read the single keystroke then revert stdin back after reading the keystroke.

Returns the character of the key that was pressed (zero on KeyboardInterrupt which can happen when a signal gets handled)

```
skaff.clitools.timeout (seconds, error_message='Timer expired')
```

```
    Sets a timer on a function; should be used as a decorator.
```

```
    Raises 'TimeoutError' upon expiration.
```

1.1.4 skaff.config module

Custom configuration type for the 'core' skaff module.

```
class skaff.config.SkaffConfig (directories, **kwargs)
```

```
    Bases: object
```

```
    Configuration type used for the argument of 'skaff' function.
```

```
    author_add (author)
```

```
        Adds 'author' to the internal 'database' if the name does not exist; otherwise do nothing.
```

```
    author_discard (author)
```

```
        Discards 'author' from the internal 'database' if the name exists; otherwise do nothing.
```

```
    static author_fetch ()
```

```
        Gets the current logged-in username from GECOS or name field. This member function is called by the constructor by default.
```

```
        Raises RuntimeError if both attempts fail.
```

```
        Note this 'classmethod' may be automatically called from 'authors_get' member function under certain scenarios.
```

```
    authors_get ()
```

```
        Gets the author(s) of the project(s).
```

```
    authors_set (authors=None)
```

```
        Sets the author(s) of the project(s). 'authors' must be an iterable type containing 'str'(s). This member function is called by the constructor by default.
```

```
        Sets the single author to be the GECOS or name field of current logged-in user if 'authors' is left as default or 'None'.
```

```
    directories_get ()
```

```
        Gets the name(s) of the outputting project-directory(ies).
```

```
    directories_set (directories=None)
```

```
        Sets the name(s) of the outputting project-directory(ies). 'directories' must be an iterable type containing 'str'(s). This member function is called by the constructor by default.
```

```
        'directories' argument must be of 'collections.Iterable' type containing instance of 'str'(s).
```


directory_add (*directory*)

Adds 'directory' to the internal 'database' if the name does not exist; otherwise do nothing.

directory_discard (*directory*)

Discards 'directory' from the internal 'database' if the name exists; otherwise do nothing.

language_get ()

Gets the major programming language used.

language_set (*language=None*)

Sets the major programming language used. Defaults to 'c' language if left as empty or 'None'. This member function is called by the constructor by default.

'language' argument must be the ones listed in 'languages_list'.

languages_list ()

Lists the supported primary programming languages. By default they are the following: {"c", "cpp"}.

license_get ()

Gets the type of license.

license_set (*license=None*)

Sets the type of license. Defaults to 'bsd2' license if left as empty or 'None'. This member function is called by the constructor by default.

'license' argument must be the ones listed in 'licenses_list'.

licenses_list ()

Lists the supported licenses. By default they are the following: {"bsd2", "bsd3", "gpl2", "gpl3", "mit"}.

quiet_get ()

Gets whether there is interactive CMakeLists.txt and Doxyfile editing.

quiet_set (*quiet=None*)

Sets whether there is interactive CMakeLists.txt and Doxyfile editing. 'True' to turn off the interactive editing. Defaults to 'False' if left as empty or 'None'. This member function is called by the constructor by default.

'quiet' argument must be of 'bool' type.

subdirectories_get ()

Gets the name(s) of the subdirectory(ies) within the project(s)' base directory(ies).

subdirectories_set (*subdirectories*)

Sets the name(s) of the subdirectory(ies) within the project(s)' base directory(ies). This member function is called by the constructor by default.

'subdirectories' argument must be of 'collections.Iterable' type containing instance of 'str'(s).

subdirectory_add (*subdirectory*)

Adds 'subdirectory' to the internal 'database' if the name does not exist; otherwise do nothing.

subdirectory_discard (*subdirectory*)

Discards 'subdirectory' from the internal 'database' if the name exists; otherwise do nothing.

1.1.5 skaff.core module

Main module of skaff.

skaff.core.skaff (*author, directories, language, license, quiet*)

Creates all the necessary subdirectories in addition to the project root.

All arguments can be 'None' except 'directories', which must be a list of strings of length ≥ 1 .

If the above is the case, ‘author’ will be guessed by GECOS field and login name of the current user; ‘language’ will default to ‘c’, ‘license’ will be set to ‘bsd2’, ‘quiet’ will be unchanged (equivalent to False).

`skaff.core.skaff_version_get()`

Returns the version information string of the skaff program.

1.1.6 Module contents

Indices and tables

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

S

skaff, 6
skaff.cli, 3
skaff.clitools, 3
skaff.config, 4
skaff.core, 5

A

ANSIColor (class in skaff.clitools), 3
author_add() (skaff.config.SkaffConfig method), 4
author_discard() (skaff.config.SkaffConfig method), 4
author_fetch() (skaff.config.SkaffConfig static method), 4
authors_get() (skaff.config.SkaffConfig method), 4
authors_set() (skaff.config.SkaffConfig method), 4

B

BLUE (skaff.clitools.ANSIColor attribute), 3
BOLD (skaff.clitools.ANSIColor attribute), 3

C

CYAN (skaff.clitools.ANSIColor attribute), 3

D

directories_get() (skaff.config.SkaffConfig method), 4
directories_set() (skaff.config.SkaffConfig method), 4
directory_add() (skaff.config.SkaffConfig method), 4
directory_discard() (skaff.config.SkaffConfig method), 5

G

GREEN (skaff.clitools.ANSIColor attribute), 3

K

KHAKI (skaff.clitools.ANSIColor attribute), 3

L

language_get() (skaff.config.SkaffConfig method), 5
language_set() (skaff.config.SkaffConfig method), 5
languages_list() (skaff.config.SkaffConfig method), 5
license_get() (skaff.config.SkaffConfig method), 5
license_set() (skaff.config.SkaffConfig method), 5
licenses_list() (skaff.config.SkaffConfig method), 5

M

MAGENTA (skaff.clitools.ANSIColor attribute), 3
main() (in module skaff.cli), 3

P

PURPLE (skaff.clitools.ANSIColor attribute), 3

Q

quiet_get() (skaff.config.SkaffConfig method), 5
quiet_set() (skaff.config.SkaffConfig method), 5

R

RED (skaff.clitools.ANSIColor attribute), 3
RESET (skaff.clitools.ANSIColor attribute), 3

S

single_keypress_read() (in module skaff.clitools), 4
skaff (module), 6
skaff() (in module skaff.core), 5
skaff.cli (module), 3
skaff.clitools (module), 3
skaff.config (module), 4
skaff.core (module), 5
skaff_version_get() (in module skaff.core), 6
SkaffConfig (class in skaff.config), 4
SmartFormatter (class in skaff.cli), 3
subdirectories_get() (skaff.config.SkaffConfig method), 5
subdirectories_set() (skaff.config.SkaffConfig method), 5
subdirectory_add() (skaff.config.SkaffConfig method), 5
subdirectory_discard() (skaff.config.SkaffConfig method), 5

T

timeout() (in module skaff.clitools), 4
TimeoutError, 4

Y

YELLOW (skaff.clitools.ANSIColor attribute), 3