
coccigrep Documentation

Release 1.0rc1

Eric Leblond

September 02, 2011

CONTENTS

1	Introduction	3
2	Documentation of module	5
3	Indices and tables	9
	Python Module Index	11
	Index	13

Contents:

INTRODUCTION

coccigrep provides an abstraction for running spatch in the scope of searching for information in C source code.

There is two interesting classes:

- `coccigrep.CocciGrep`: main class that run the intensive task
- `coccigrep.CocciGrepConfig`: used to parse and get configuration value

To build a request:

- Create a `coccigrep.CocciGrep` instance
- Call `coccigrep.CocciGrep.setup()` function to setup the search
- Call `coccigrep.CocciGrep.run()` function to execute the search
- Call `coccigrep.CocciGrep.display()` function to display the output of the search

DOCUMENTATION OF MODULE

exception `coccigrep.CocciConfigException` (*value*)

Exception raised when configuration parameter are not correct.

For example, it is returned if spatch command can not be found.

exception `coccigrep.CocciException` (*value*)

Generic class for coccigrep exception

class `coccigrep.CocciGrep`

Core class of the module: setup and run.

This class is the core of the module. It is responsible of initialisation and running of the request.

add_operations (*new_ops*)

Add operation to the list of supported operations

Parameters *new_ops* (*list of str*) – list of filenames (ending by .cocci)

display (*mode='raw', before=0, after=0, oformat='term'*)

Display output for complete request

Parameters

- **mode** (*str*) – display mode
- **before** (*int*) – number of lines to display before match
- **after** (*int*) – number of lines to display after match
- **oformat** (*str*) – format of output for color (term, html)

Returns the result of the search as a str

get_datadir ()

get_operation_name (*fname*)

get_operations ()

Get list of available operations

Returns list of operations in a list of str

run (*files*)

Run the search against the files given in argument

This function is doing the main job. It will run spatch with the correct parameters by using subprocess or it will use multiprocessing if a concurrency level greater than 1 has been asked.

Parameters *files* (*list of str*) – list of filenames

Raise `CocciRunException` or `CocciConfigException`

set_concurrency (*ncpus*)

Set concurrency level (number of spatch command to run in parallel)

Parameters *ncpus* (*int*) – number of process to launch in parallel

set_spatch_cmd (*cmd*)

Set path or command name for spatch

Parameters *cmd* (*str*) – Name of parth of the spatch command

set_verbose ()

Activate verbose mode

setup (*stype, attribut, operation*)

Parameters

- **stype** (*str*) – structure name, used to replace ‘\$type’ in the cocci file
- **attribut** (*str*) – basically attribut of the structure, used to replace ‘\$attribut’ in the cocci file
- **operation** (*str*) – search operation to do

Raise `CocciRunException`

class `coccigrep.CocciGrepConfig`

Configuration handling class

This class parses configuration and can be used to access to configuration item via get operations. This is mainly a wrapper around configparser.

get (*section, value*)

Get value for a configuration item

Parameters

- **section** (*str*) – name of the section in the ini file
- **value** (*str*) – name of the value under the section

Returns value of option as a str

getboolean (*section, value*)

Get value for a configuration item returned as boolean

Parameters

- **section** (*str*) – name of the section in the ini file
- **value** (*str*) – name of the value under the section

Returns value of option as a boolean

getint (*section, value*)

Get value for a configuration item returned as int

Parameters

- **section** (*str*) – name of the section in the ini file
- **value** (*str*) – name of the value under the section

Returns value of option as a int

parse_config()

Parse the hierarchy of configuration files

class coccigrep.**CocciMatch** (*mfile, mline, mcol, mlineend, mcolend*)

Store a match and take care of its display

display (*stype, mode='raw', oformat='term', before=0, after=0*)

Display output for a single match

Parameters

- **mode** (*str*) – display mode
- **oformat** (*str*) – format of output for color (term, html)
- **before** (*int*) – number of lines to display before match
- **after** (*int*) – number of lines to display after match

class coccigrep.**CocciProcess** (*cmd, verbose*)

Class used for running spatch command in the case of multiprocessing

execute (*option=''*)

join ()

recv ()

start ()

exception coccigrep.**CocciRunException** (*value*)

Exception raised when running parameters are not correct.

For example, it is returned if a required argument is missing.

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

PYTHON MODULE INDEX

C

`coccigrep`, 5

INDEX

A

`add_operations()` (coccigrep.CocciGrep method), 5

C

`CocciConfigException`, 5

`CocciException`, 5

`CocciGrep` (class in coccigrep), 5

`coccigrep` (module), 5

`CocciGrepConfig` (class in coccigrep), 6

`CocciMatch` (class in coccigrep), 7

`CocciProcess` (class in coccigrep), 7

`CocciRunException`, 7

D

`display()` (coccigrep.CocciGrep method), 5

`display()` (coccigrep.CocciMatch method), 7

E

`execute()` (coccigrep.CocciProcess method), 7

G

`get()` (coccigrep.CocciGrepConfig method), 6

`get_datadir()` (coccigrep.CocciGrep method), 5

`get_operation_name()` (coccigrep.CocciGrep method), 5

`get_operations()` (coccigrep.CocciGrep method), 5

`getboolean()` (coccigrep.CocciGrepConfig method), 6

`getint()` (coccigrep.CocciGrepConfig method), 6

J

`join()` (coccigrep.CocciProcess method), 7

P

`parse_config()` (coccigrep.CocciGrepConfig method), 6

R

`recv()` (coccigrep.CocciProcess method), 7

`run()` (coccigrep.CocciGrep method), 5

S

`set_concurrency()` (coccigrep.CocciGrep method), 6

`set_spatch_cmd()` (coccigrep.CocciGrep method), 6

`set_verbose()` (coccigrep.CocciGrep method), 6

`setup()` (coccigrep.CocciGrep method), 6

`start()` (coccigrep.CocciProcess method), 7