
incor Documentation

Release 1.0.0

lakshmanaram, srivatsan-ramesh

Oct 30, 2017

Contents

1	Requirements	3
2	Getting Started	5
3	How to Use	7
4	Using Templates	9
5	Using Input files	11
6	Demo	13
7	Configuring INCOR	15
8	Contributors	17
8.1	API Reference	17
9	Indices and tables	19
	Python Module Index	21

INstant COde Runner runs the programs present in the mentioned directory instantaneously as and when changes are saved.

Version	Quality	Documentation

CHAPTER 1

Requirements

- python>=2.7
- watchdog>=0.8.2
- psutil>=4.3.0

CHAPTER 2

Getting Started

Install it using pip

```
sudo pip install incor
```

Or

Clone this repository to get started.

```
git clone https://github.com/lakshmanaram/Incor.git
```

And install it.

```
python setup.py install
```

You can also install it for development.

```
python setup.py develop
```


CHAPTER 3

How to Use

Navigate to the folder where you want the files to be compiled and executed instantly and type this in the terminal

```
incor
```

Or else you can run it directly with the path of the directory.

```
incor /path/to/the/directory
```

General template for running incor.

```
incor /optional/path/to/the/directory [<option> <value> ...]
```


CHAPTER 4

Using Templates

You can also add some template files in the directory where incor runs. The template files should be of the form ‘template.(some_extension)’ for example ‘template.c’ or ‘template.py’. Whenever a new empty file is created inside the working directory with a particular extension, Incor searches for a template file with the same extension and writes the template into the newly created file.

Example:

All the empty files created in the present directory with .cpp extension will have the contents of a.cpp, .c extension will have the contents of a.c and .py extension will have the contents of a.py

```
incor -t a
```

All the empty files created in the present directory with extension in [cpp,c,py] will have the contents of template.extension if available anywhere in the path/to/the/directory

```
incor path/to/the/directory -t
```

Using Input files

You can also add some input files in the directory where incor runs. The default input file is 'input.txt'. Whenever a program is executed and run, Incor provides the contents of the specified input file as input to the program.

Example:

All the successfully compiled and executed programs will use input.txt present in path/to/the/directory or in any of it's sub-directories as input

```
incor path/to/the/directory -i
```

All the successfully compiled and executed programs will use a.txt present in the current working directory or in any of it's sub-directories as input

```
incor -i a.txt
```


CHAPTER 6

Demo

Youtube link

Configuring INCOR

incor can be configured for a run using these options -

- | | |
|-------------|--|
| -i | To specify the input file name for the to be compiled program(with extension). |
| -t | To specify the name of template file(without extension). |
| -c | To specify the C compiler to be used. |
| -cpp | To specify the C++ compiler to be used. |
| -py | To specify the python interpreter to be used. |

Srivatsan R

Adarsh B

If you have an idea for a new feature that could be added, Go ahead! I will be happy to see a pull request from you!

Contents:

API Reference

`incor.main.main()`

The entry function for the command 'incor' in terminal

class `incor.EventHandler.EventHandler` (*path, compilers*)

The class inherited from `watchdog.events.FileSystemEventHandler`, overrides the functions `on_modified()` and `on_created()` from the base class.

on_created (*event*)

Overrides the function `on_created()` from the base class

Parameters *event* (`watchdog.events.FileModifiedEvent`) –

Specifies the type of event

on_modified (*event*)

Overrides the function `on_modified()` from the base class

Parameters *event* (`watchdog.events.FileModifiedEvent`) –

Specifies the type of event

CHAPTER 9

Indices and tables

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

i

`incor.EventHandler`, [17](#)
`incor.main`, [17](#)

E

EventHandler (class in incor.EventHandler), [17](#)

I

incor.EventHandler (module), [17](#)

incor.main (module), [17](#)

M

main() (in module incor.main), [17](#)

O

on_created() (incor.EventHandler.EventHandler method),
[17](#)

on_modified() (incor.EventHandler.EventHandler
method), [17](#)