Xenon Documentation

Release 0.5.0

Michele Lacchia

August 07, 2016

Contents

Xenon is a monitoring tool based on Radon. It monitors your code's complexity. Ideally, Xenon is run every time you commit code. Through command line options, you can set various thresholds for the **complexity** of your code. It will fail (i.e. it will exit with a non-zero exit code) when any of these requirements is not met.

CHAPTER 1

Installation

With Pip:

```
$ pip install xenon
```

Or download the source and run the setup file (requires setuptools):

```
$ python setup.py install
```

CHAPTER 2

Usage

Typically you would use Xenon in two scenarios:

1. As a `git commit` hook: to make sure that your code never exceeds some complexity values.
2. On a **continuous integration** server: as a part of your build, to keep under control, as above, your code's complexity. See Xenon's .travis.yml file for an example usage.

2.1 The command line

Everything boils down to Xenon's command line usage. To control which files are analyzed, you use the options `-e, --exclude` and `-i, --ignore`. Both accept a comma-separated list of glob patterns. The value usually needs quoting at the command line, to prevent the shell from expanding the pattern (in case there is only one). Every filename is matched against the *exclude* patterns. Every directory name is matched against the *ignore* patterns. If any of the patterns matches, Xenon won't even descend into them.

The actual threshold values are defined through these options:

- `-a, --max-average`: Threshold for the *average* complexity (across all the codebase).
- `-m, --max-modules`: Threshold for *modules* complexity.
- `-b, --max-absolute`: *Absolute* threshold for *block* complexity.

All of these options are inclusive.

2.2 An actual example

```
$ xenon --max-absolute B --max-modules A --max-average A
```

or, more succinctly:

```
$ xenon -b B -m A -a A
```

With these options Xenon will exit with a non-zero exit code if any of the following conditions is met:

- At least one block has a rank higher than `B` (i.e. `C`, `D`, `E` or `F`).
- At least one module has a rank higher than `A`.
- The average complexity (among all of the analyzed blocks) is ranked with `B` or higher.

CHAPTER 3

Other resources

For more information regarding cyclomatic complexity and static analysis in Python, please refer to Radon's documentation, the project on which Xenon is based on:

- More on cyclomatic complexity: http://radon.readthedocs.org/en/latest/intro.html
- More on Radon's ranking: http://radon.readthedocs.org/en/latest/commandline.html#the-cc-command