

---

# **ConfigMe Documentation**

*Release 0.5dev*

**Michael R**

**Sep 27, 2017**



---

## Contents

---

<b>1</b>	<b>What Problem Does ConfigMe Solve For You?</b>	<b>3</b>
<b>2</b>	<b>Whats Next?</b>	<b>5</b>
<b>3</b>	<b>Narrative Documentation</b>	<b>7</b>
3.1	Tutorial . . . . .	7
3.2	Usage . . . . .	11
<b>4</b>	<b>Support and Documentation</b>	<b>13</b>
<b>5</b>	<b>How it Works</b>	<b>15</b>
<b>6</b>	<b>Current Project Status</b>	<b>17</b>
6.1	Test Coverage . . . . .	17
<b>7</b>	<b>Source Code</b>	<b>19</b>
<b>8</b>	<b>API</b>	<b>21</b>
<b>9</b>	<b>License</b>	<b>23</b>
<b>10</b>	<b>Authors</b>	<b>25</b>
<b>11</b>	<b>Indices and tables</b>	<b>27</b>



ConfigMe is a small, fast, down-to-earth, open source configuration generation framework and command line utility. It makes generation of real-world configuration files and deployment more fun, more predictable, and more productive.



---

## What Problem Does ConfigMe Solve For You?

---

Whether used in production, staging or development the majority of the configuration files are usually **95% to 99%** identical and it can prove **time consuming and error-prone** to maintain them in sync.

ConfigMe solves this problem for you.



## CHAPTER 2

---

Whats Next?

---

Learn ConfigMe by following the *Tutorial*



## Tutorial

This document provides a brief tutorial on how to use ConfigMe.

## Installation

1. Clone the ConfigMe repo
2. Setup a virtualenv environment
3. `python ./setup develop`

## Defining The Problem

To demonstrate the use of of *ConfigMe* we will generate set of development and production configuration for NginX for a project called *my-site*.

Now lets look at a typical NginX configuration setup.

```
/etc/  
|-- nginx  
  |-- nginx.conf  
  |-- sites-available  
    |-- my-site.conf
```

However for `nginx.conf` production configuration we want *gzipping on* while in development configuration we want *gzipping off*.

Additionally we want `my-site` to be served on *port 80* in production and *port 8080* in development.

## Configuration Folder Skeleton

First create the folder skeleton that will contain the templates, settings and output:

```
cd /tmp

mkdir -p /tmp/configme/templates/etc/nginx/sites-available
mkdir -p /tmp/configme/settings
mkdir -p /tmp/configme/output
```

## Create etc/nginx/nginx.conf template

```
$EDITOR /tmp/configme/templates/etc/nginx/nginx.conf
```

```
1 user www-data;
2 worker_processes 4;
3 pid /var/run/nginx.pid;
4
5 events {
6     worker_connections 768;
7 }
8
9 http {
10     sendfile on;
11     tcp_nopush on;
12     tcp_nodelay on;
13     keepalive_timeout 65;
14     types_hash_max_size 2048;
15
16     include /etc/nginx/mime.types;
17     default_type application/octet-stream;
18
19     access_log /var/log/nginx/access.log;
20     error_log /var/log/nginx/error.log;
21
22     gzip {{gzip_status}};
23     include /etc/nginx/sites-enabled/*;
24 }
```

---

**Note:** The config file is a Jinja2 template and as you can see line 22 contains **gzip\_status** variable.

---

## Create etc/nginx/sites-available/my-site.conf template

```
$ $EDITOR /tmp/configme/templates/etc/nginx/sites-available/my-site.conf
```

```
1 server {
2     listen {{site_port}};
3     root /usr/share/nginx/www;
4     index index.html index.htm;
5     server_name localhost;
6     location / {
7         try_files $uri $uri/ /index.html;
```

```

8     }
9 }

```

**Note:** The config file is also Jinja2 template and line 2 contains `site_port` variable.

## Add development Role Settings

Now lets create the `development.configme` file and set the `gzip_status` and `site_port` variables to their respective development values.

```
$EDITOR /tmp/configme/settings/development.configme
```

```

1 [etc/nginx/nginx.conf]
2
3 gzip_status = off
4
5 [etc/nginx/sites-available/my-site.conf]
6
7 site_port = 8080

```

**Note:** The `development.configme` settings file uses the INI format and contains sections for each of the configuration files that development role will have generated. **If the file is not included in here it will not generated.**

## Add production Role Settings

Now lets create the `production.configme` file and set the `gzip_status` and `site_port` variables to their respective production values.

```
$EDITOR /tmp/configme/settings/production.configme
```

```

1 [etc/nginx/nginx.conf]
2
3 gzip_status = on
4
5 [etc/nginx/sites-available/my-site.conf]
6
7 site_port = 80

```

**Note:** The `production.configme` settings file uses the INI format and contains settings for the production role. As you can see the two configurations only differ slightly.

## Resulting Configuration Skeleton

Now check the result in `/tmp/configme/`

```
/tmp/configme/  
|-- output  
|-- settings  
|   |-- development.configme  
|   `-- production.configme  
`-- templates  
    |-- etc  
    |-- nginx  
        |-- nginx.conf  
        `-- sites-available  
            |-- my-site.conf
```

## Generate development Configuration

```
configme \  
  --templates-path=/tmp/configme/templates \  
  --settings-path=/tmp/configme/settings \  
  --output-path=/tmp/configme/output \  
  --role-name development
```

## Generate production Configuration

```
configme \  
  --templates-path=/tmp/configme/templates \  
  --settings-path=/tmp/configme/settings \  
  --output-path=/tmp/configme/output \  
  --role-name production
```

## Resulting Configuration Sets

```
/tmp/configme/output/  
|-- development  
|   |-- etc  
|   |-- nginx  
|       |-- nginx.conf  
|       `-- sites-available  
|           |-- my-site.conf  
`-- production  
    |-- etc  
    |-- nginx  
        |-- nginx.conf  
        `-- sites-available  
            |-- my-site.conf
```

## What's Next?

Learn more about ConfigMe usage in the next section: *Usage*

## Usage

This document describes basic usage for ConfigMe.

ConfigMe is run using `configme` command-line utility, uses Jinja2 templates and INI files (using Python's ConfigParser)

## Terms

**Role** is a collection of configuration files generated for a specific purpose. An example of role names can be: *development*, *stage*, *production* or anything you want.

**Configuration Template** is file containing template for configuration file. The template may contain variables that will be replaced by values contained in the role's settings file. Currently only templates are written in Jinja2 templating language. Support for other templating languages is planned.

**Role Settings** is a file in a INI format containing variables for each of the configuration files

## Usage and Command-line Options

To see usage run `configme --help`

```
usage: configme [-h] -t TEMPLATES_PATH -s SETTINGS_PATH -o OUTPUT_PATH -r
              ROLE_NAME [-u ROLE_SUFFIX]
              [-b ROLE_VARIABLES [ROLE_VARIABLES ...]]

configme 0.4dev command line utility.

optional arguments:
  -h, --help            show this help message and exit
  -t TEMPLATES_PATH, --templates-path TEMPLATES_PATH
                        Path to configuration templates folder.
  -s SETTINGS_PATH, --settings-path SETTINGS_PATH
                        Path to settings folder.
  -o OUTPUT_PATH, --output-path OUTPUT_PATH
                        Path to output folder.
  -r ROLE_NAME, --role-name ROLE_NAME
                        Role name.
  -u ROLE_SUFFIX, --role-suffix ROLE_SUFFIX
                        Role suffix.
  -b ROLE_VARIABLES [ROLE_VARIABLES ...],
  --role-variables ROLE_VARIABLES [ROLE_VARIABLES ...]
                        Variables that will interpolated into the settings
                        files.
```

## File Naming Conventions

ConfigMe uses INI files and Python's ConfigParsers which places some limitations on naming of sections in the INI files. Since these sections names correspond to files located on the filesystems this limitations extends there as well.

### Role Names

The name should be have consist of characters considered valid for a file name on the operating system it is run on. It must also be a correct section name in the INI style *configme* settings file. Additionally it must be at least marginally human readable.

As such due to difficulty maintaining OS specific forbidden characters set, complying with INI file specifications, and keeping readability a set of forbidden characters have chosen.

The naming rules are:

- name cannot start with a: *space*
- name cannot contain: < > : ' " / \ | ? \* '

### File Names and Paths

The path should be have consist of characters considered valid for a file name on the operating system it is run on. It must also be a correct section name in the INI style settings file. Additionally it must be at least marginally human readable.

As such due to difficulty maintaining OS specific forbidden characters set, complying with INI file specifications, and keeping readability a set of forbidden characters have chosen.

The naming rules are:

- name cannot start with a: *space / ../ ./*
- name cannot contain: *../ ./ < > : ' " | ? \* '*

---

### Support and Documentation

---

Read the official ConfigMe documentation here: <http://configme.readthedocs.org>

Ask for help on IRC: [irc.freenode.org](http://irc.freenode.org) in #pyramid channel, mention “**configme**” in your question, to alert people who can help.

You can use the IRC WebChat here: <http://webchat.freenode.net/?channels=pyramid>

To report bugs, and obtain support please use ConfigMe issue tracker here:  
<http://github.com/goodwillcoding/configme/issues>



## CHAPTER 5

---

### How it Works

---

ConfigMe is written in Python, uses Jinja2 for templates and INI files for settings configuration. Support for Django, Mako and other templates is planned.



---

## Current Project Status

---

Project is still in development at 90% completion. That means it works but there no guarantees it won't break. The names, command-line options and code interfaces may change so backward compatibility is currently not guaranteed.

## Test Coverage

**Current unit test stats:**

<b>Line Coverage</b> (nosexcov*):	100%
<b>Number of unit tests:</b>	84

\* nosexcov: <http://pypi.python.org/pypi/nosexcov>



## CHAPTER 7

---

### Source Code

---

Source code is located on GitHub: <http://github.com/goodwillcoding/configme>



## CHAPTER 8

---

API

---

API will be available at a later date when the implementation stabilizes.



## CHAPTER 9

---

### License

---

ConfigMe is offered under the BSD-derived [Repoze Public License](#).



## CHAPTER 10

---

Authors

---

ConfigMe is produced by the Goodwill Coding.

ConfigMe is developed by *Michael R.*



# CHAPTER 11

---

## Indices and tables

---

- [Glossary](#)
- [genindex](#)
- [search](#)

### Current unit test stats:

<b>Line Coverage</b> (nosexcov*):	100%
<b>Number of unit tests:</b>	84

\* nosexcov: <http://pypi.python.org/pypi/nosexcov>



## A

About, 1  
API, 1  
Authors, 1

## B

Bugs, 1

## L

License, 1

## P

Project Status, 1

## S

Source Code, 1  
Support and Documentation, 1

## T

Tutorial, 7

## U

Usage, 10