# mrbob Documentation

*Release 0.1a4*

**Domen Kožar, Tom Lazar**

December 11, 2012

# CONTENTS

**Author** Tom Lazar <tom@tomster.org>, Domen Kožar <domen@dev.si>

**Source code** github.com project

**Bug tracker** github.com issues

**License** BSD

**Generated** December 11, 2012

**Version** 0.1a4

**Features**

- asks questions which need to be answered to render structure
- questions can be grouped by using a namespace
- renders templates from a folder, Python egg, or zip file
- supports Python 2.6 - 3.3, pypy
- 100% test coverage
- uses Jinja2 as the default rendering engine (can be replaced)
- multiple ways to specify variables to render templates
- preserves permissions when rendering templates

**Flow of mr.bob**

**Introduction**

**mr.bob** is a tool that takes a directory skeleton, copies over its directory structure to a target folder, and can use the Jinja2 (or some other) rendering engine to dynamically generate the files. Additionally, it can ask you questions needed to render the structure, or provide a config file to answer them.
**mr.bob** is meant to deprecate previous tools such as paster (PasteScript) and templer.

CHAPTER

ONE

# USER GUIDE

## 1.1 Installation

```
$ pip install mr.bob
```

## 1.2 Usage

Once you install mr.bob, the *mrbob* command is available:

```
$ mrbob --help
usage: mrbob [-h] [-O TARGET_DIRECTORY] [-c CONFIG] [-V] [-l] [-r RENDERER]
             [template]

Filesystem template renderer

positional arguments:
  template              Template to use for rendering

optional arguments:
  -h, --help            show this help message and exit
  -O TARGET_DIRECTORY, --target-directory TARGET_DIRECTORY
                        Where to output rendered structure. Defaults to
                        current directory.
  -c CONFIG, --config CONFIG
                        Configuration file to specify either [mr.bob] or
                        [variables] sections.
  -V, --version         Display version number
  -l, --list-questions  List all questions needed for the template
  -r RENDERER, --renderer RENDERER
                        Dotted notation to a renderer function. Defaults to
                        mrbob.rendering:jinja2_renderer
```

By default, the target directory is the current folder. The most basic use case is rendering a template from a relative folder:

```
$ mrbob ../template_folder/
```

Or from a package:

```
$ mrbob some.package:template_folder/
```

Or from a zip file:

```
https://github.com/iElectric/mr.bob/zipball/master
```

Or from a relative path in a zip file:

```
https://github.com/iElectric/mr.bob/zipball/master#mrbob/template_sample
```

## 1.3 Sample template to try out

```
$ mrbob mrbob:template_sample/
Welcome to mr.bob interactive mode. Before we generate directory structure, some questions need to be

Answer with a question mark to display help.
Value in square brackets at the end of the questions present default value if there is no answer.


--> How old are you? [24]:

--> What is your name?: Foobar

--> Enter password:


Generated file structure at /current/directory/
```

## 1.4 Listing all questions needed to have corresponding variable for a template

```
$ mrbob --list-questions mrbob:template_sample/
author.age.default = 24
author.age.help = We need your age information to render the template
author.age.question = How old are you?
author.name.question = What is your name?
author.name.required = True
author.password.command_prompt = getpass:getpass
author.password.question = Enter password
```

## 1.5 Configuration

Configuration is done with *.ini* style files. There are two sections for configuration: *mr.bob* and *variables*.

Example of global config file *~/.mrbob* or command line parameter *mrbob –config foo.ini*.

```
[mr.bob]
renderer = moo.foo:render_mako

[variables]
author.name = Domen Kožar
author.email = domen@dev.si
```

### 1.5.1 Configuration inheritance

Configuration can be specified in multiple ways. See flow of mr.bob on the documentation front page to know how options are preferred.

### 1.5.2 Nesting variables into namespaces called groups

All variables can be specified in namespaces, such as *author.name*. Currently namespaces don't do anything special besides providing readability.

### 1.5.3 `mr.bob` section reference

| Parameter | Default | Explanation |
|---|---|---|
| renderer | mrbob.rendering:jinja2_renderer | Function for rendering templates |
| verbose | False | Output more information, useful for debugging |

## 1.6 Collection of community managed templates

You are encouraged to use the *bobtemplates.something* Python egg namespace to write templates and contribute them to this list by making a pull request.

- bobtemplates.ielectric

CHAPTER

TWO

# WRITING YOUR OWN TEMPLATE

## 2.1 Starting

Writing your own template is as easy as creating a *.mrbob.ini* that may contain questions. Everything else is extra. To start quickly, use the template starter that ships with *mr.bob*:

```
$ mr.bob mrbob:template_starter/
Welcome to mr.bob interactive mode. Before we generate directory structure, some questions need to be

Answer with a question mark to display help.
Value in square brackets at the end of the questions present default value if there is no answer.


--> How old are you? [24]:

--> What is your name?: Foobar

--> Enter password:


Generated file structure at /home/ielectric/code/mr.bob
```

See *.mrbob.ini* for sample questions and *sample.txt.bob* for sample rendering.

## 2.2 Templating

Files inside the structure can be just copied to destination, or they can be suffixed with *.bob* and the templating engine will be used to render them.

By default a slightly customized *Jinja2* templating is used. The big differences are that variables are referenced with *{{{ variable }}}* instead of *{{ variable }}* and blocks are *{{% if variable %}}* instead of *{% if variable %}*. To read more about templating see Jinja2 documentation.

Variables can also be used on folder and file names. Surround variables with plus signs. For example *foo/+author+/+age+.bob* given variables *author* being *Foo* and *age* being *12*, *foo/Foo/12* will be rendered.

Templating engine can be changed by specifying *renderer* in mr.bob config section in *dotted notation*. It must be a callable that expects a text source as the first parameter and a dictionary of variables as the second.

When rendering the structure, permissions will be preserved for files.

## 2.3 Writing Questions

*[question]* section in *.mrbob.ini* specifies a *schema* for how *[variables]* are validated. Example speaks for itself:

```
[questions]
author.name.question = What is your name?
author.required = True

author.age.question = How old are you?
author.age.help = We need your age information to render the template
author.age.default = 24

author.password.question = Enter password
author.password.command_prompt = getpass:getpass
```

Questions will be asked in the order written in *.mrbob.ini*.

### 2.3.1 `questions` section reference

| Parameter | Default | Explanation |
|---|---|---|
| name | | Required. Unique identifier for the question |
| question | | Required. Question given interactively to a user when generating structure |
| default | None | Default value when no answer is given. Can be a *dotted notation* |
| required | False | Specify if question must be answered |
| action | lambda x: x | Extra action to be taken except returning value to be used stored in variables |
| validator | None | Validator can raise `mrbob.configurator.ValidationError` and question will be asked again |
| com-mand_prompt | `raw_input()` | Function that accepts a question and asks user for the answer |
| help | “” | Extra help returned when user inputs a question mark |

## 2.4 Validators

Validators are functions with an answer as the only parameter. They may return a value to be used as an answer and may raise `ValidationError` for the question to be asked again.

See `mrbob.validators` for validators that ship with *mr.bob*.

CHAPTER

THREE

# DESIGN GOALS

- Cover 80% of use cases, don’t become too complex
- Ability to use templates not only from eggs, but also folders and similar
- Python 3 support
- Jinja2 renderer by default, but replaceable
- Ability to render multiple templates to the same target directory

CHAPTER

FOUR

# WHY ANOTHER TOOL

- PasteScript is a big package with lots of legacy code and noone seems to care about maintaining it (and porting it to python3)
- a tool should do one thing and that thing good, which is where PasteScript fails
- PasteScript works only with Python eggs, mr.bob can also render templates from folder and in future maybe from http links
- PasteScript uses Cheetah which doesn't work on PyPy and has C extensions that need to be compiled
- PasteScript in unmaintainable, with really dodgy code
- PasteScript doesn't preserve permissions when copying/rendering files
- mr.bob is just 200 lines of code with some extra features in mind that PasteScript cannot provide, such as a Python API for use by higher level libraries

CHAPTER

FIVE

# DEVELOPER GUIDE

## 5.1 Setup developer environment

```
$ git clone https://github.com/iElectric/mr.bob.git
$ cd mrbob
$ virtualenv .
$ source bin/activate
$ python setup.py develop
$ easy_install mr.bob[test,development]
$ mrbob --help
```

## 5.2 Running tests

Easy as:

```
$ ./bin/test
```

## 5.3 Making a Release

Using *zest.releaser*:

```
$ bin/fullrelease
```

CHAPTER

SIX

# SOURCE DOCUMENTATION

## 6.1 `mrbob` – Main package

### 6.1.1 `mrbob.configurator` – Machinery to figure out configuration

**exception** `mrbob.configurator.`**`ConfigurationError`**

Bases: `mrbob.configurator.MrBobError`

Raised during configuration phase

**class** `mrbob.configurator.`**`Configurator`**(*template*, *target_directory*, *bobconfig=None*, *variables=None*)

Bases: `object`

Controller that figures out settings and renders file structure.

**Parameters**

- **template** – Template name
- **target_directory** – Filesystem path to a output directory
- **bobconfig** – Configuration for mr.bob behaviour
- **variables** – Given variables

**`ask_questions`**()

Loops through questions and asks for input if variable is not yet set.

**`render`**()

Render file structure given instance configuration. Basically calls `mrbob.rendering.render_structure()`.

**exception** `mrbob.configurator.`**`MrBobError`**

Bases: `exceptions.Exception`

Base class for errors

**class** `mrbob.configurator.`**`Question`**(*name*, *question*, *default=None*, *required=False*, *action=<function <lambda> at 0x22e16e0>*, *validator=None*, *command_prompt=<built-in function raw_input>*, *help=''*)

Bases: `object`

Question configuration. Parameters are used to configure validation of the answer.

**`ask`**()

Eventually, ask the question.

**exception** `mrbob.configurator.`**`TemplateConfigurationError`**

Bases: `mrbob.configurator.ConfigurationError`

Raised reading template configuration

**exception** `mrbob.configurator.`**`ValidationError`**

Bases: `mrbob.configurator.MrBobError`

Raised during question validation

`mrbob.configurator.`**`parse_template`**(*template_name*)

Resolve template name into absolute path to the template and boolean if absolute path is temporary directory.

### 6.1.2 `mrbob.cli` – Command line interface

Command line interface to mr.bob

`mrbob.cli.`**`main`**(*args=['-b', 'latex', '-d', '_build/doctrees', '.', '_build/latex'], quiet=False*)

Main function called by *mrbob* command.

### 6.1.3 `mrbob.parsing` – Parsing *.ini* files

### 6.1.4 `mrbob.rendering` – Everything related to rendering templates and directory structure

`mrbob.rendering.`**`render_structure`**(*fs_source_root*, *fs_target_root*, *variables*, *verbose*, *renderer*)

Recursively copies the given filesystem path *fs_source_root_ to a target directory 'fs_target_root*.

Any files ending in *.bob* are rendered as templates using the given renderer using the variables dictionary, thereby losing the *.bob* suffix.

strings wrapped in + signs in file- or directory names will be replaced with values from the variables, i.e. a file named +*name*+.*py.bob* given a dictionary {'name': 'bar'} would be rendered as *bar.py*.

### 6.1.5 `mrbob.validators` – Useful validators for questions

`mrbob.validators.`**`boolean`**(*value*)

Converts value to Python boolean given values: y, n, yes, no, true, false, 1, 0

CHAPTER

SEVEN

# GLOSSARY

**dotted notation** Importable Python function specified with dots as importing a module separated with a column to denote a function. For example *mrbob.rendering:render_structure*

**mr.bob** configures how *mrbob* behaves

**variables** answers to the questions that will be passed to templates for rendering

CHAPTER

EIGHT

# INDICES AND TABLES

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

# PYTHON MODULE INDEX