Cookiecutter Context V2 Fork Documentation Documentation

Release 2.0.0-alpha.u+1.6.0

E.R. Uber

Nov 04, 2017

Contents:

CHAPTER 1

Introduction

This documentation serves as a set of notes for the developer and a record of testing & implementation for those individuals who wish to better understand the technical details of forking the Cookiecutter Repo on GitHub in order to implement a new **cookeicutter.json** format spec as proposed by hackebrot's cookiecutter pull request #848.

1.1 Cookiecutter References

The following are useful Cookiecutter references:

- ReadTheDocs
- Cookiecutter Repo on GitHub
- hackebrot's cookiecutter pull request #848 - the catalyst for this implementation effort
- Cookiecutter v2 Fork

1.2 Timeline

The initial implementation & test of this work was done over the course of 8 days between 20 Oct and 28 Oct 2017. After 100% test coverage was attained, the code was re-visited due to an implementation defect in the handling of **extra context** overwrites – it needed to be more sophisticated for v2 contexts. The defect resolution was attained on 29 Oct 2017 with 100% test coverage; however, more tests were added on 30 Oct 2017 to cover all **extra context** overwrites for each type of field supported in the v2 context.

A couple days were consumed implementing and testing a utility to allow easy conversion of a version 1 **cookiecutter** template file to version 2 file. This effort is a separate GitHub project.

CHAPTER 2

Dev/Test Environment Setup

The following sections detail the steps taken to get a development and test environment configured to establish a testing baseline.

Primary development and testing was done using:

```
Python 3.6.2 (v3.6.2:5fd33b5, Jul  8 2017, 04:14:34)
[MSC v.1900 32 bit (Intel)] on win32
```

Some **Posix** testing was also done as described in the next section.

2.1 Notes Regarding Development Console Sessions

Note that two different consoles are used on **Windows 10** (which is the source of most documented console sessions in this document):

1. The git-bash console uses a **$** prompt
2. The ConEmu console session uses a **>** prompt

On **Posix** a Virtual Box system running Ubuntu Desktop **16.04.3 LTS** on a **Windows 10** host was used to run unit tests and do adhoc command line testing.

2.2 Development/Test Environment Setup

The following steps were taken to setup the initial development & test environment:

1. Fork the Cookiecutter Repo on GitHub to the Cookiecutter Version 2 Project on GitHub
2. Clone the forked repo to local development machine, create a development branch and check it out:

```
$ cd /d/Devel/_python/__eru/___forks/
$ git clone https://github.com/eruber/cookiecutter.git
$ cd cookiecutter
$ git checkout -b new-2.0-context
```

3. Create a virtual environment for development and test:

```
> cd D:\Devel\_python\__eru\___forks\cookiecutter
> python -m venv .env
```

   We chose **.env** because that directory is defined in the **.gitignore** file.

4. Activate the virtual environment & list pre-installed packages:

```
>.env\Scripts\activate.bat
(.env) >pip list

Package    Version
---------- -------
pip        9.0.1
setuptools 28.8.0
```

5. Install **cookicutter** in development (editable) mode & list cookiecutter's installed dependencies:

```
(.env) >pip install -e .

(.env) >pip list

Package         Version     Location
--------------- ----------- --------------------------------------------
arrow           0.10.0
binaryornot     0.4.4
certifi         2017.7.27.1
chardet         3.0.4
click           6.7
cookiecutter    1.6.0       d:\devel\_python\__eru\___forks\cookiecutter
future          0.16.0
idna            2.6
Jinja2          2.9.6
jinja2-time     0.2.0
MarkupSafe      1.0
pip             9.0.1
poyo            0.4.1
python-dateutil 2.6.1
requests        2.18.4
setuptools      28.8.0
six             1.11.0
urllib3         1.22
whichcraft      0.4.1
```

6. Install test requirements & list installed packages:

```
(.env) >pip install -r test_requirements.txt

(.env) >type test_requirements.txt
pytest
pytest-cov
pytest-mock==1.1
```

```
pytest-catchlog
freezegun

(.env) >pip list
Package         Version     Location
--------------- ----------- --------------------------------------------
arrow           0.10.0
binaryornot     0.4.4
certifi         2017.7.27.1
chardet         3.0.4
click           6.7
colorama        0.3.9
cookiecutter    1.6.0       d:\devel\_python\__eru\___forks\cookiecutter
coverage        4.4.1
freezegun       0.3.9
future          0.16.0
idna            2.6
Jinja2          2.9.6
jinja2-time     0.2.0
MarkupSafe      1.0
pip             9.0.1
poyo            0.4.1
py              1.4.34
pytest          3.2.3
pytest-catchlog 1.2.2
pytest-cov      2.5.1
pytest-mock     1.1
python-dateutil 2.6.1
requests        2.18.4
setuptools      28.8.0
six             1.11.0
urllib3         1.22
whichcraft      0.4.1
```

7. Run pytest (the **setup.cfg** file configures pytest to run from the project root):

```
(.env) D:\Devel\_python\__eru\___forks\cookiecutter>pytest
============================= test session starts ========================
platform win32 -- Python 3.6.2, pytest-3.2.3, py-1.4.34, pluggy-0.4.0
rootdir: D:\Devel\_python\__eru\___forks\cookiecutter, inifile: setup.cfg
plugins: mock-1.1, cov-2.5.1, catchlog-1.2.2
collected 261 items

tests\test_abort_generate_on_hook_error.py ..
tests\test_cli.py ...........................
tests\test_cookiecutter_invocation.py ..
tests\test_cookiecutter_local_no_input.py .......
tests\test_cookiecutter_local_with_input.py ..
tests\test_custom_extensions_in_hooks.py ..
tests\test_default_extensions.py .
tests\test_environment.py ..
tests\test_exceptions.py .
tests\test_find.py ..
tests\test_generate_context.py ..........
tests\test_generate_copy_without_render.py .
tests\test_generate_file.py .....
tests\test_generate_files.py ..................
tests\test_generate_hooks.py ...s....
```

```
tests\test_get_config.py .....
tests\test_get_user_config.py .........
tests\test_hooks.py ..........
tests\test_log.py ...
tests\test_main.py ..
tests\test_output_folder.py ..
tests\test_preferred_encoding.py .
tests\test_prompt.py .........................
tests\test_read_repo_password.py .
tests\test_read_user_choice.py .....
tests\test_read_user_dict.py .......
tests\test_read_user_variable.py .
tests\test_read_user_yes_no.py .
tests\test_repo_not_found.py .
tests\test_specify_output_dir.py ..
tests\test_utils.py .........
tests\replay\test_dump.py .....
tests\replay\test_load.py ....
tests\replay\test_replay.py ......
tests\repository\test_abbreviation_expansion.py .......
tests\repository\test_determine_repo_dir_clones_repo.py .....
tests\repository\test_determine_repo_dir_finds_existing_cookiecutter.py .
tests\repository\test_determine_repository_should_use_local_repo.py ...
tests\repository\test_is_repo_url.py .............
tests\repository\test_repository_has_cookiecutter_json.py ...
tests\vcs\test_clone.py ..........
tests\vcs\test_identify_repo.py .............
tests\vcs\test_is_vcs_installed.py ....
tests\zipfile\test_unzip.py .............


----------- coverage: platform win32, python 3.6.2-final-0 -----------
Name                          Stmts   Miss  Cover
-------------------------------------------------
cookiecutter\__init__.py          2      0   100%
cookiecutter\__main__.py          3      0   100%
cookiecutter\cli.py              49      0   100%
cookiecutter\config.py           51      0   100%
cookiecutter\environment.py      21      0   100%
cookiecutter\exceptions.py       24      0   100%
cookiecutter\extensions.py        9      0   100%
cookiecutter\find.py             18      0   100%
cookiecutter\generate.py        166      0   100%
cookiecutter\hooks.py            61      1    98%
cookiecutter\log.py              21      0   100%
cookiecutter\main.py             31      0   100%
cookiecutter\prompt.py           90      0   100%
cookiecutter\replay.py           30      0   100%
cookiecutter\repository.py       39      0   100%
cookiecutter\utils.py            50      0   100%
cookiecutter\vcs.py              54      0   100%
cookiecutter\zipfile.py          61      2    97%
-------------------------------------------------
TOTAL                           780      3    99%


=================== 260 passed, 1 skipped in 25.54 seconds ===============
```

At this point, we have a working test environment based on Cookiecutter v1.6.0; therefore test development and implementation work can begin.

To compare these baseline test numbers to the v2.0.0 test numbers go to the *Testing* section.

2.3 Additional Setup to Build Documentation

In order to build the Cookiecutter docs, the following additional packages need to be installed:

```
pip install sphinx
pip install sphinx_rtd_theme
```

Then change directory into the repo's top-level **cookiecutter/docs** directory and do:

```
make html
```

My experience with the docs build is that it succeeded but generated 26 warnings.

To view the docs, point your web browser at:

```
cookiecutter/docs/_build/html/index.html
```

CHAPTER 3

The Implementation

Before implementation began a general survey of the **Cookiecutter v1.6.0** codebase was performed and a set of implementation goals were developed to help shape any implementation decisions that might have to be made.

3.1 Implementation Goals

The line in the sand here is that **Cookiecutter v1.6.0** is compatible with the current **cookiecutter.json** context format – lets call that context version v1; and **Cookiecutter v2** adds *additional* support for the new **cookiecutter.json** context format as initially defined by hackebrot's cookiecutter pull request #848 – lets call that context v2.

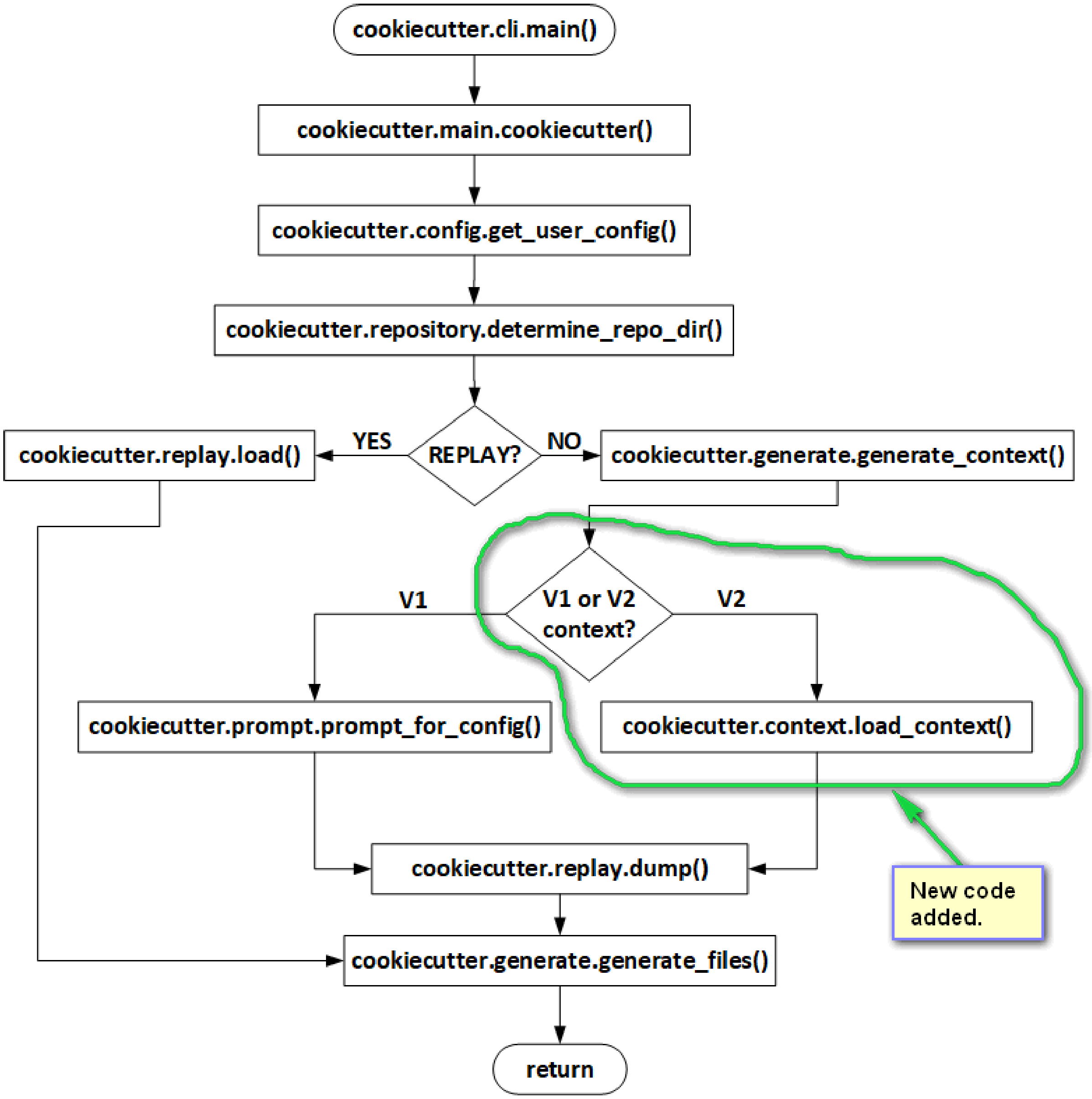
The following goals served to guide the implementation of v2 context support:

1. **Cookiecutter v2.0.0** supports *both* context v1 and context v2 formats.
2. Disturb as little of the existing v1.6.0 codebase as possible - this implementation is not a re-write, or re-factoring of the code that is already in place for v1.6.0; but a graft of support for v2 context onto the v1.6.0 codebase.
3. The **Cookiecutter** user interface does not change.
4. Cookiecutter's external package dependencies do not change.
5. Use the context module code referenced by hackebrot's cookiecutter pull request #848 and located in the new-context-format branch of this hackebrot repo.
6. All existing tests for v1 context (**Cookiecutter 1.6.0**) still pass as originally written when a v1 context is provided via the **cookiecutter.json** file – in other words, there is *no* contamination of the 1.6.0 codebase by any v2 changes.
7. The new code paths and features implemented that support context v2 have tests that follow the testing structure already defined for v1.6.0; and of course, these new tests achieve 100% code coverage of any new code paths.

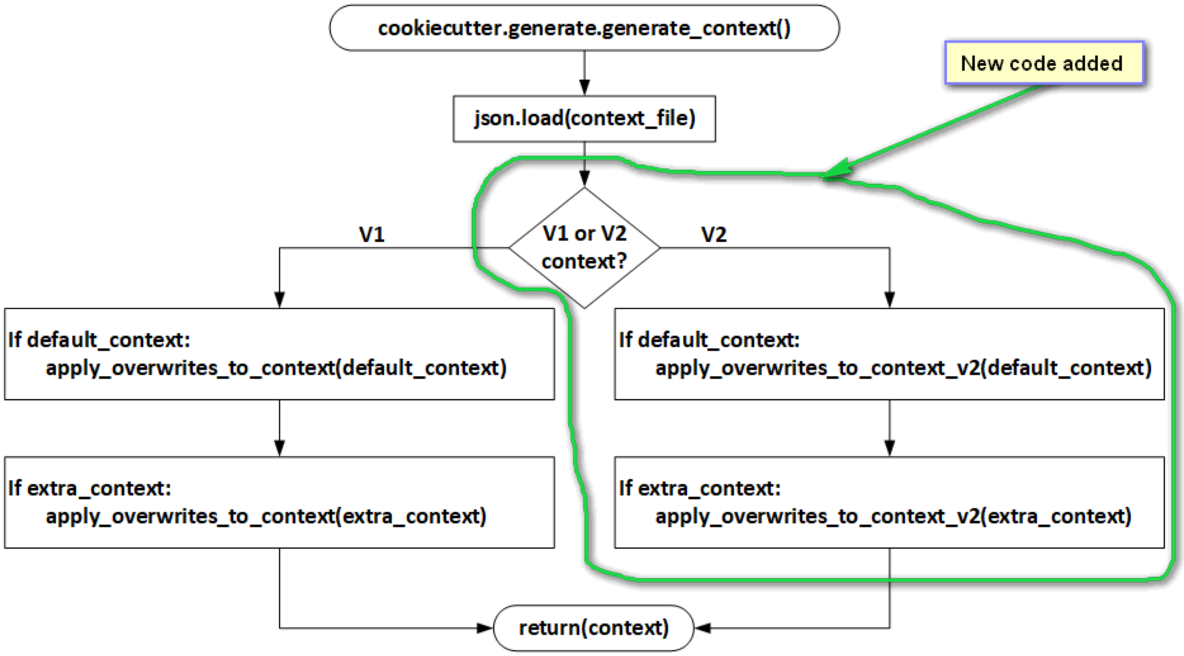
3.2 Implementation Overview

In keeping with **Implementation Goal #2** (disturb little) only two source files were updated.

The first being **cookiecutter/main.py** as shown in the diagram below:


cookiecutter.cli.main()
cookiecutter.main.cookiecutter()
cookiecutter.config.get_user_config()
cookiecutter.repository.determine_repo_dir()
cookiecutter.replay.load()
YES
REPLAY?
NO
cookiecutter.generate.generate_context()
V1
V1 or V2 context?
V2
cookiecutter.prompt.prompt_for_config()
cookiecutter.context.load_context()
cookiecutter.replay.dump()
New code added.
cookiecutter.generate.generate_files()
return

The second source file updated was **cookiecutter/generate.py** as shown in the diagram below:


cookiecutter.generate.generate_context()
New code added
json.load(context_file)
V1
V1 or V2 context?
V2
If default_context:
apply_overwrites_to_context(default_context)
If default_context:
apply_overwrites_to_context_v2(default_context)
If extra_context:
apply_overwrites_to_context(extra_context)
If extra_context:
apply_overwrites_to_context_v2(extra_context)
return(context)

The only new source file added to the project was **cookiecutter/context.py** which implements the v2 context support. This is the code referenced in **Implementation Goal #5** above.

The actual repo containing the original version of **context.py** is located located in the new-context-format branch of this hackebrot repo.

3.3 Implementation Statistics

Comparing statement coverage reports between **Cookiecutter** v1.6.0 and v2.0.0 provides us with some indication of how much code was added to the v1.6.0 codebase (a total of 222 statements)

```
                               v1.6.0  v2.0.0   Delta   Delta
   File Name                    Stmts   Stmts   Stmts     %      Notes
---------------------------------------------------------------------------
cookiecutter\__init__.py          2       2       0       0% <-- version bump
cookiecutter\__main__.py          3       3       0
cookiecutter\cli.py              49      49       0
cookiecutter\config.py           51      51       0
cookiecutter\context.py           -     163    +163    +100% <-- v2 new context
cookiecutter\environment.py      21      21       0
cookiecutter\exceptions.py       24      24       0
cookiecutter\extensions.py        9       9       0
cookiecutter\find.py             18      18       0
cookiecutter\generate.py        166     222     +56   +33.7% <-- v2 overwrites
cookiecutter\hooks.py            61      61       0
cookiecutter\log.py              21      21       0
cookiecutter\main.py             31      34      +3    +9.7% <-- v2 context load
cookiecutter\prompt.py           90      90       0
cookiecutter\replay.py           30      30       0
cookiecutter\repository.py       39      39       0
cookiecutter\utils.py            50      50       0
cookiecutter\vcs.py              54      54       0
```

```
cookiecutter\zipfile.py           61       61        0
---------------------------------------------------------------------------
                    Stmt Totals    780     1002      222     +28.5%
```

As you can see from the table above, aside from the version bump, only two original v1.6.0 files were modified and only one new file was added.

3.4 Modified Files

This section provides a bit more detail on what changes were made to the files modified:

```
cookiecutter/__init__.py (version bump no further details given herein)

cookiecutter/main.py

cookiecutter/generate.py
```

3.4.1 cookiecutter/main.py

The **cookiecutter/context.py** file contains a function that determines if a given context is v1 or v2 – this function is called in **main.py** to differentiate between processing the original v1 context via a call to **prompt.prompt_for_config()** or calling **context.load_context()** to process the new v2 context.

3.4.2 cookiecutter/generate.py

The changes made to **cookiecutter/generate.py** all have to do with supporting default and extra context overwrites for version 2 templates.

Added **generate.apply_default_overwrites_to_context_v2()** to mirror for v2 context what **generate.apply_overwrites_to_context()** does for v1 context.

Also added **generate.apply_overwrites_to_context_v2()** to handle v2 context overwrites via the extra context.

Modifed **generate.generate_context()** to check for v1 or v2 context and call the previously mentioned v2 overwrite functions.

Overwrite Considerations Regarding ‘default’ & ‘choices’ Fields

When a variable is defined that has both the ‘default’ and the ‘choices’ fields, these two fields influence each other. If one of these fields is updated, but not the other field, then the other field will be automatically updated by the overwrite logic.

If both fields are updated, then the ‘default’ value will be moved to the first location of the ‘choices’ field if it exists elsewhere in the list; if default value is not in the list, it will be added to the first location in the choices list. The overwrite logic will take care of this even though the **extra context** choices list does not explicitly specify this behavior.

Special Overwrite Syntax

A couple of special syntax tokens were introduced in the **extra context** processing contained in **generate.apply_overwrites_to_context_v2()** that allow the following overwrite capabilities:

1. Change ‘name’ field in a variable
2. Remove a field from a variable

Changing the ‘name’ Field in a Variable

Because the algorithm chosen to find a variable’s dictionary entry in the variables list of OrderDicts uses the variable’s ‘name’ field; it could not be used to simultaneously hold a new ‘name’ field value.

Therefore the following **extra context** dictionary entry snytax was introduced to allow the ‘name’ field of a variable to be changed:

```
{
   'name': 'CURRENT_VARIABLE_NAME::NEW_VARIABLE_NAME',
}
```

The variable’s current name is post-fixed with a double colon (::) followed by the new name of the variable.

So, for example, to change a variable’s ‘name’ field from ‘director_credit’ to ‘producer_credit’, would require:

```
{
   'name': 'director_credit::producer_credit',
}
```

Removing a Field from a Variable

It is possible that a previous **extra context** overwrite requires that a subsequent variable entry be removed.

In order to accomplish this a **remove field token** is used in the **extra context** as follows:

```
{
   'name': 'director_cut',
   'skip_if': '<<REMOVE::FIELD>>',
}
```

In the example above, the **extra context** overwrite results in the variable named ‘director_cut’ having it’s ‘skip_if’ field removed.

3.5 New Files

The only new file added to the implementation is:

```
cookiecutter/context.py
```

3.5.1 cookiecutter/context.py

This file takes care of processing the new v2 context – the base code in this file was written by **@hackebrot** and available at the new-context-format branch of this hackebrot repo and is referenced and discussed heavily in hackebrot’s cookiecutter pull request #848.

The following features were added to the base code:

- added support for **float** type since **click.prompt** supports that type
- added support for **UUID** type since **click.prompt** supports that type
- added context v2 check function – used in **main.py** & **generate.py**
- type checking on context field injection (inbound data from the JSON file)

- implemented validation of variable's default or user input value
- added validation flags to allow controlling the validation – ignoring case, etc
- added docstring comments to document parameters for classes **Variable** and **CookiecutterTemplate**
- added docstring comments to document parameters for function **load_context()**
- insured CLI option *–no-input* also suppresses v2 context user prompts
- added method **Variable.__str__()** to help with debugging

3.6 Implementation TODOs

The following sections attempt to document what additional implementation features could be realized in the future.

3.6.1 The Cookiecutter Command Line

Add a dump context option that emits the context variable list, but does not render any project files.

I actually have implemented this is a stand-alone command line tool, after implementing it, I realized it would be trivial to just add a flag option to the cookiecutter CLI to do this. But this was not implemented herein because it would violate **Implementation Goal #3**.

3.6.2 cookiecutter/context.py

1. Could add a **CookiecutterTemplate.__str__()** that uses the **Variable.__str__()** to realize a complete string dump of the **CookecutterTemplate** object for debugging. This would be very easy to do.
2. A better specification of what differentiates a v1 context from a v2 context should be considered in the future. Right now a v2 context must define the following fields:

   ```
   name
   cookiecutter_version
   variables
   ```

   At the very least, we could verify that 'variables' is a list of **OrderedDict** objects. But at the moment, we just check for the existence of the three fields.
3. Perhaps adding a variable field named *include* of type **click.File** that specifies a path to a file that can be included/injected into the template. Potential file formats supported could be JSON, INI, YAML, TOML, etc. The idea here being that the file contents could be loaded (included) in the context namespace – includes should probably be done prior to any of the current context processing. This would allow the context namespace to be aware of the contents of other configuration files in the project. Of course this whole idea borders on making Cookiecutter a compiler of configuration files which in and of itself is probably too heavy of a lift.

3.7 Documentation Updates

Documentation that has been updated:

1. The API documentation has been updated

Documentation that has **not** yet been updated (mostly because of a desire for some concensus moving forward with the user docs):

1. The user manual portion of the docs has not been updated, but has been reviewed and the best approach I think would be to add a new third tutorial covering the version 2 template format. Also any place in the user docs where a version 1 format is shown, we could add the analogous version 2 format as well.
2. If this pull request gains momentum, and a concensus is reached on what approach is best for the user docs, I will be happy to do the initial implementation.

CHAPTER 4

Testing

The baseline v1.6.0 test run in the *Setup section* showed totals of:

```
=================== 260 passed, 1 skipped in 25.54 seconds ===============
```

The v2.0.0 test run below shows the following totals:

```
=================== 308 passed, 1 skipped in 38.97 seconds =================
```

So thus far, 48 tests have been added:

```
              v1.6.0  v2.0.0    Delta   Delta
               Tests   Tests    Tests     %
Test Totals     261     309      48    +18.4%
```

The was made to keep all version 2 tests in their own files – this makes them more obivous and does not require any edits to existing version 1 test files (no contamination of version 1 tests).

The following new test files & test support directories were added:

> **tests/test_generate_context_v2.py** - Tests v2 changes made to **generate.py**.
>
> **tests/test-generate-context-v2/** - Contains test support files for **tests/test_generate_context_v2.py**.
>
> **tests/test_context.py** - Tests the new v2 file **context.py**.
>
> **tests/test-context/** - Contains test support files for **tests/test_context.py**.

Test coverage for v2.0.0 looks like this:

```
----------- coverage: platform win32, python 3.6.2-final-0 -----------
Name                          Stmts   Miss  Cover   Missing
-----------------------------------------------------------
cookiecutter\__init__.py          2      0   100%
cookiecutter\__main__.py          3      0   100%
cookiecutter\cli.py              49      0   100%
cookiecutter\config.py           51      0   100%
cookiecutter\context.py         163      0   100%       <--- new file
```

```
cookiecutter\environment.py      21      0   100%
cookiecutter\exceptions.py       24      0   100%
cookiecutter\extensions.py        9      0   100%
cookiecutter\find.py             18      0   100%
cookiecutter\generate.py        222      0   100%       <--- modified file
cookiecutter\hooks.py            61      1    98%   95
cookiecutter\log.py              21      0   100%
cookiecutter\main.py             34      0   100%       <--- modified file
cookiecutter\prompt.py           90      0   100%
cookiecutter\replay.py           30      0   100%
cookiecutter\repository.py       39      0   100%
cookiecutter\utils.py            50      0   100%
cookiecutter\vcs.py              54      0   100%
cookiecutter\zipfile.py          61      2    97%   10-11
-----------------------------------------------------------
TOTAL                          1002      3    99%

=================== 308 passed, 1 skipped in 42.20 seconds =================
```

Also to improve the test coverage report, an additional option was added to the **pytest** section of the **set.cfg** file.

The git diff looks like this:

```
$ git diff setup.cfg
diff --git a/setup.cfg b/setup.cfg
index 83901b4..805cc52 100644
--- a/setup.cfg
+++ b/setup.cfg
@@ -16,5 +16,5 @@ universal = 1

 [tool:pytest]
 testpaths = tests
-addopts = --cov=cookiecutter
+addopts = --cov-report term-missing --cov=cookiecutter
```

This additional **‘–cov-report term-missing’** option adds the *Missing* column to the test coverage report so you can easily determine what lines of code are missing coverage.

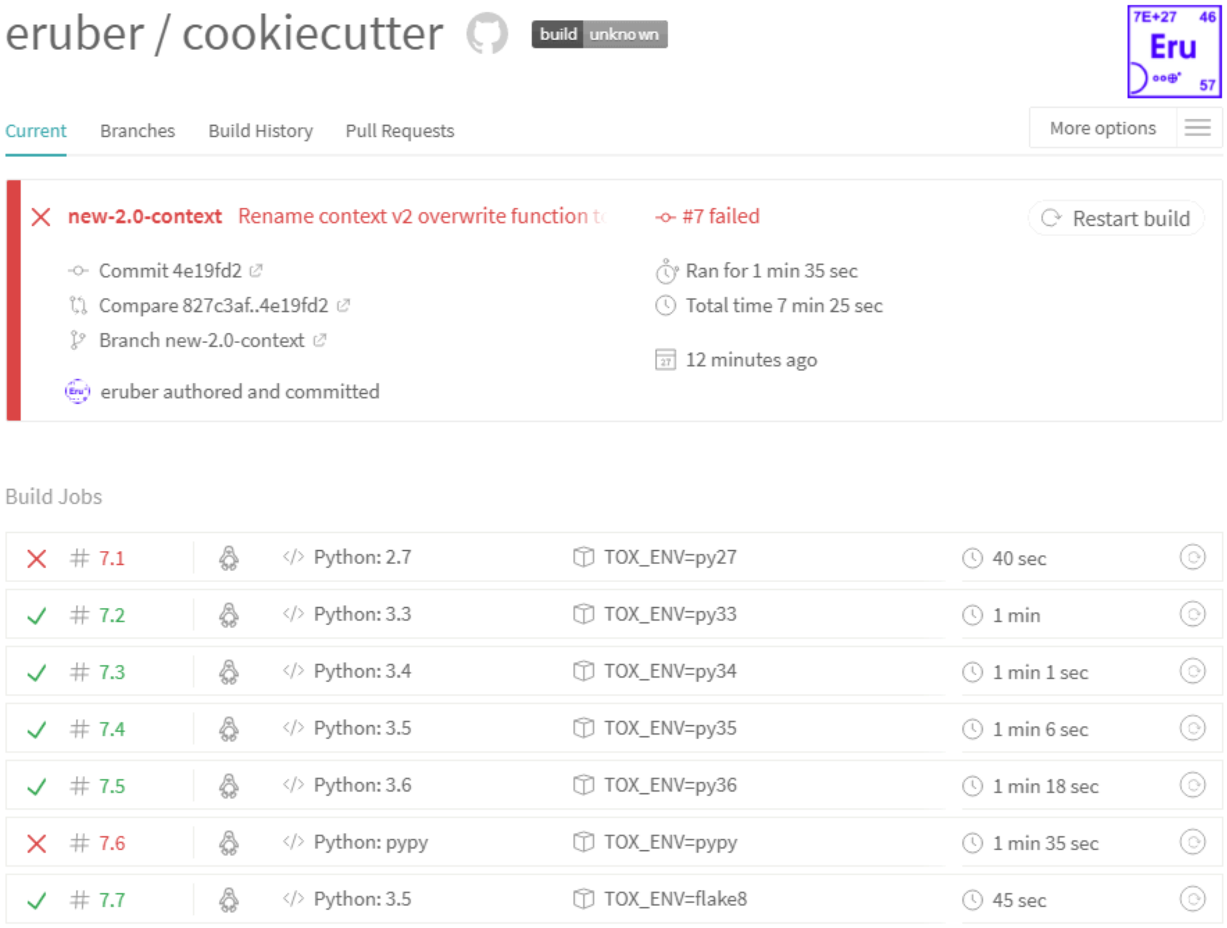
CHAPTER 5

CI Travis Status

The Travis screen shot below illustrates the current status of the project measured against the official **Cookiecutter Contributor Guidelines**:

- Support for Python versions 3.6, 3.5, 3.4, 3.3
- **Not supporting** Python 2.7 and PyPy (which is based on Python 2.7.13)

For the latest up-to-date Travis status see Cookiecutter v2 @ Travis.



✗ new-2.0-context Rename context v2 overwrite function t

#7 failed

Commit 4e19fd2

Compare 827c3af..4e19fd2

Branch new-2.0-context

eruber authored and committed

Ran for 1 min 35 sec

Total time 7 min 25 sec

12 minutes ago

Build Jobs

✗	# 7.1	Python: 2.7	TOX_ENV=py27	40 sec	
✓	# 7.2	Python: 3.3	TOX_ENV=py33	1 min	
✓	# 7.3	Python: 3.4	TOX_ENV=py34	1 min 1 sec	
✓	# 7.4	Python: 3.5	TOX_ENV=py35	1 min 6 sec	
✓	# 7.5	Python: 3.6	TOX_ENV=py36	1 min 18 sec	
✗	# 7.6	Python: pypy	TOX_ENV=pypy	1 min 35 sec	
✓	# 7.7	Python: 3.5	TOX_ENV=flake8	45 sec	

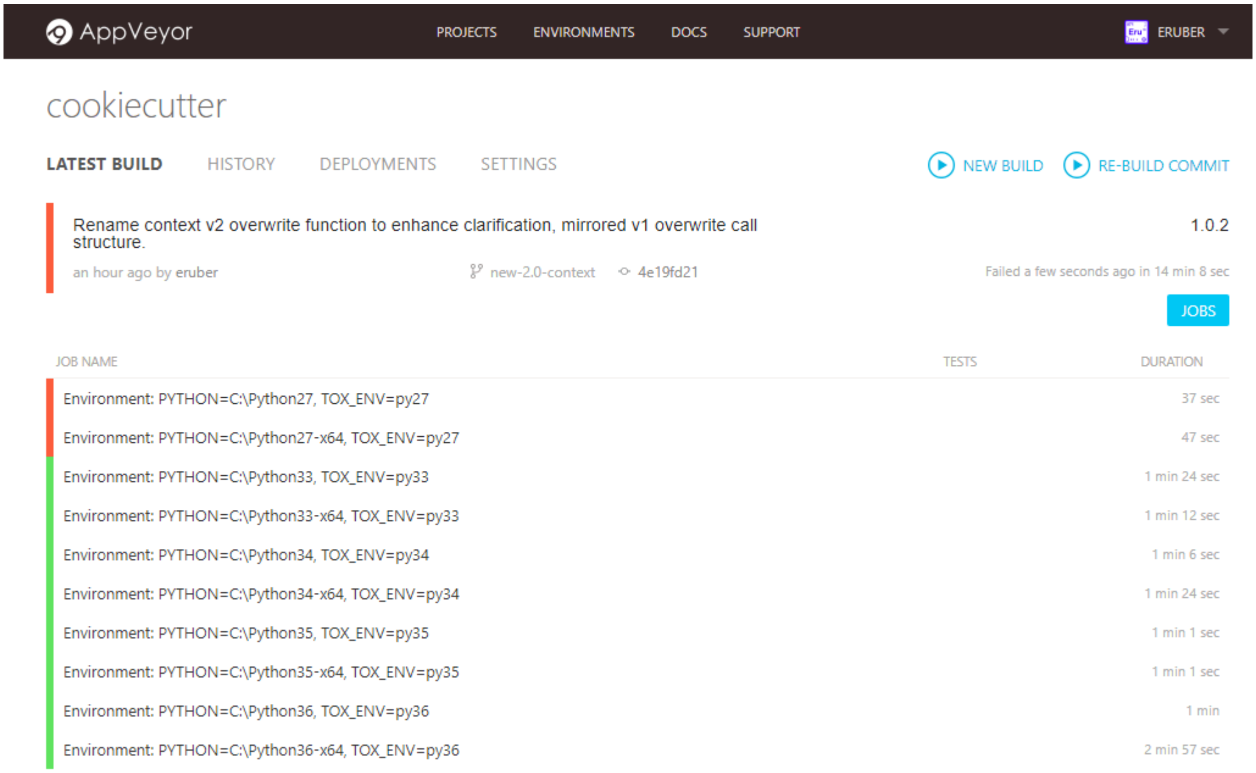
CHAPTER 6

CI AppVeyor Status

The AppVeyor screen shot below illustrates the current status of the project measured against the official **Cookiecutter Contributor Guidelines**:

- Support for Python versions 3.6, 3.5, 3.4, 3.3 (both 32 & 64 bit)
- **Not supporting** Python 2.7

For the latest up-to-date Travis status see Cookiecutter v2 @ AppVeyor .



🟥 Rename context v2 overwrite function to enhance clarification, mirrored v1 overwrite call structure. — 1.0.2

an hour ago by eruber — new-2.0-context — 4e19fd21 — Failed a few seconds ago in 14 min 8 sec

	JOB NAME	TESTS	DURATION
🟥	Environment: PYTHON=C:\Python27, TOX_ENV=py27		37 sec
🟥	Environment: PYTHON=C:\Python27-x64, TOX_ENV=py27		47 sec
🟩	Environment: PYTHON=C:\Python33, TOX_ENV=py33		1 min 24 sec
🟩	Environment: PYTHON=C:\Python33-x64, TOX_ENV=py33		1 min 12 sec
🟩	Environment: PYTHON=C:\Python34, TOX_ENV=py34		1 min 6 sec
🟩	Environment: PYTHON=C:\Python34-x64, TOX_ENV=py34		1 min 24 sec
🟩	Environment: PYTHON=C:\Python35, TOX_ENV=py35		1 min 1 sec
🟩	Environment: PYTHON=C:\Python35-x64, TOX_ENV=py35		1 min 1 sec
🟩	Environment: PYTHON=C:\Python36, TOX_ENV=py36		1 min
🟩	Environment: PYTHON=C:\Python36-x64, TOX_ENV=py36		2 min 57 sec

CHAPTER 7

Tools

In the course of doing this implementation a couple of utitlity tools were developed and are presented herein.

7.1 Context Dump Utility

This utility allows one to play with different **cookiecutter.json** and see what context variables are produced without actually doing any jinja2 rendering of files.

This very same functionality could be easily attained using cookiecutter if the command line interface supported a –dumpcontext option.

7.1.1 contextdump.py

Usage: contextdump.py [OPTIONS]

> Load the specified json object from the –cct PATH and if –no-input is omitted, prompt the user for input. After all input is gathered, the template's context is dumped.

Options:

--cct PATH Path to a Cookiecutter template file (cct), defaults to 'cookiecutter.json'

--no-input Do not prompt for parameters and only use cookiecutter.json file content

--tdump If specified, dump the input Cookiecutter template file specified by the –cct PATH option

-h, --help Show this message and exit.

Requires an environment that includes a cookiecutter installation.

7.2 Template Transformer

This utility will convert a version 1 **cookiecutter.json** file to version 2.

The original **cookiecutter.json** file will be renamed cookiecutter_v1.json and the transformed version 2 file will become **cookiecutter.json**.

7.2.1 templatetransformer.py

CHAPTER 8

Conclusion

Apologies for the length of this document, but it helped me clarify my thoughts as I progressed through the implementation. Most things don't seem all that complicated from afar, but when you start looking at them with a microscope, you can get shoulder's deep in no time. Write ups help me deal with complexity.

It is obvious from the discussion in hackebrot's cookiecutter pull request #848 that the Cookiecutter maintainers and community feel that adding v2 context support is a fairly risky thing that will certainly increase the overall support effort. Those sentiments are definitely valid from my perspective as well.

This whole digression for me snowballed gradually and then gained momentum simply because I wanted a "skip_if" feature and I was so impressed by hachebrot's **context.py** code.

Another apology for not including Python 2.7 support as per the Contributor Guidelines – I lack the ergs to do it now – I would like to say that quite honestly, for me Python 2.7 is a blip on the horizon in my rear-view mirror. However, with some help, I am sure we could clear this hurdle.

Anyway, I do want to make the pull request available regardless, to generate some discussion and feedback. Perhaps it will ulitmately lead to something positive for the **Cookiecutter** community.

As menioned in the introduction, I have a repo here containing a command line tool that will convert a version 1 template to a version 2 template. I of course will not deploy it to PyPI since it would, no doubt, cause all sorts of confusion. But it is available for experimentation.

Anyway, I shall submit my pull request now... if anyone actually read this far, I am impressed with your fortitude.