
laforge

Release 0.1.6

Matt VanEseltine

2019 Jun 29

CONTENTS

1	Overview	3
2	An Example Build	5
2.1	Directory Organization	5
2.2	.env	5
2.3	build.ini	5
3	Module Documentation	7
3.1	builder	7
3.2	command	8
3.3	distros	8
3.4	sql	8
3.5	tech	10
3.6	toolbox	10
4	Contributing to Development	11
4.1	Suggested Environment	11
4.2	Embedded TODOs	12
4.3	Docstring Gaps	12
5	Free Software License	15
	Python Module Index	17
	Index	19

laforge: a low-key build system for data work.

OVERVIEW

laforge is a low-key build system designed to interoperate Python, SQL, and Stata data work, originally developed internally for [IRIS, the Institute for Research on Innovation and Science](#) at the [University of Michigan's Institute for Social Research](#).

Features:

- Interoperable: Read, write, and execute Python, SQL, and Stata scripts/data.
- Straightforward: Simple build INI files designed for a one-click build.
- No lock-in: Maintain scripts independent from *laforge*.

AN EXAMPLE BUILD

2.1 Directory Organization

Because *laforge* needs to find the scripts, and because scripts will likely interact with output from other scripts, I tend to keep related project/sub-project files together:

```
project
├── input
│   ├── a.csv
│   ├── b.xlsx
│   └── c.csv
├── output
│   ├── results_d.csv
│   ├── results_e.csv
│   └── results_f.csv
├── .env
├── build.ini
├── g.py
├── h.sql
├── i.py
└── j.py
```

2.2 .env

```
distro = mssql
server = MSSQL
database = testdb
schema = laforge
```

2.3 build.ini

```
[DEFAULT]
```


MODULE DOCUMENTATION

3.1 builder

Builder reads and executes tasks and lists of tasks.

`laforge.builder.show_env (path=None)`
Show the calculated generic section environment

`laforge.builder.run_build (script_path, *, log, debug=False, dry_run=False)`
laforge's core build command

class `laforge.builder.Verb`
An enumeration.

class `laforge.builder.Target`
An enumeration.

class `laforge.builder.FileCall (method, kwargs)`

property `kwargs`
Alias for field number 1

property `method`
Alias for field number 0

exception `laforge.builder.TaskConstructionError`

exception `laforge.builder.TaskExecutionError`

class `laforge.builder.TaskList (from_string, location='.')`

Todo: Implement cache_results=False

load_section_config (section='DEFAULT')
Put together config from env, TaskList config, section config

execute ()
Execute each task in the list.

Todo: Restore quiet?

dry_run ()
List each task in the list.

```
class laforge.builder.BaseTask (*, identifier, verb, target, content, config)
    Create a task to (verb) (something)
```

Todo:

```
if ":" in self.content: previous_result_key, actual_path_content = self.content.split(":")
```

property path

For handlers where dir[verb] + content = path

```
class laforge.builder.FileReader (*, identifier, verb, target, content, config)
```

```
class laforge.builder.InternalPythonExecutor (*, identifier, verb, target, content, config)
    Execute (without importing) Python script by path
```

This will set `__main__` = 'laforge'

..todo

Allow implicit/explicit **return** of results.

```
class laforge.builder.ShellExecutor (*, identifier, verb, target, content, config)
```

```
class laforge.builder.StataExecutor (*, identifier, verb, target, content, config)
```

```
class laforge.builder.SQLQueryReader (*, identifier, verb, target, content, config)
```

```
class laforge.builder.SQLExecutor (*, identifier, verb, target, content, config)
```

```
class laforge.builder.SQLReaderWriter (*, identifier, verb, target, content, config)
```

```
class laforge.builder.FileWriter (*, identifier, verb, target, content, config)
    Handles all tasks writing to file.
```

```
class laforge.builder.ExistenceChecker (*, identifier, verb, target, content, config)
```

3.2 command

Command-line interface for laforge.

3.3 distros

```
exception laforge.distros.SQLDistroNotFound
```

3.4 sql

SQL utilities for mid-level interaction. Inspired by pathlib; powered by SQLALchemy.

Note: Supported: MSSQL, MariaDB/MySQL, PostgreSQL, SQLite. Supportable: Firebird, Oracle, Sybase.

```
exception laforge.sql.SQLTableNotFound
```

```
exception laforge.sql.SQLChannelNotFound
```

exception laforge.sql.SQLIdentifierProblem

class laforge.sql.Channel (distro, *, server=None, database=None, schema=None, ***engine_kwargs*)

Abstraction from Engine, other static details.

execute_statement (statement, fetch=False)

Execute SQL (core method)

Todo: De-messify

laforge.sql.execute (statement, fetch=False, channel=None)

Convenience method, autofetches Channel if possible

class laforge.sql.Script (query, channel=None)

SQL query string, parsable by 'go' separation and execute()able.

execute (statements=None)

Execute itself(s)

to_table ()

Executes all and tries to return a DataFrame for the result of the final query.

This is one of two ways that laforge retrieves tables.

Warning: This is limited by the capacity of Pandas to retrieve only the final result. For Microsoft SQL Server, if a lengthy set of queries is desired, the most reliable approach appears to be a single final query after a 'go' as a batch terminator.

Warning: This will rename columns that do not conform to naming standards.

class laforge.sql.Table (name, channel=None, ***kwargs*)

Represents a SQL table, featuring methods to read/write DataFrames.

Todo: Factor out to superclass to allow views

write (df, if_exists='replace')

From DataFrame, create a new table and fill it with values

read ()

Return the full table as a DataFrame

drop (ignore_existence=False)

Delete the table within SQL

class laforge.sql.Scalar (prox)

Little helper to produce clearly typed single (upper left) ResultProxy result.

class laforge.sql.Identifier (user_input, extra=None)

Single standardized variable/database/schema/table/column/anything identifier.

Todo: class InvalidIdentifierError relay_id_problem(identifier, action, reason=None, replacement=None)

3.5 tech

`laforge.tech.make_first_upper(s)`

Uppercase the first letter of s, leaving the rest alone.

3.6 toolbox

Handful of utility functions

Note: These intentionally *only* depend on builtins.

Note: Some copyright information within this file is identified per-block below.

`laforge.toolbox.round_up(n, nearest=1)`

Round up n to the nearest nearest.

Parameters

- **n** –
- **nearest** – (Default value = 1)

`laforge.toolbox.prepare_to_access(path)`

Make directory exist and verify that file would be writable

`laforge.toolbox.verify_file_is_writable(path, retry_attempts=3, retry_seconds=5)`

Check for locked file (e.g. Excel has CSV open)

`laforge.toolbox.flatten(foo)`

Take any set of nests in an iterator and reduce it into one generator.

‘Nests’ include any iterable except strings.

Parameters foo –

Note: `flatten()` was authored by [Amber Yust](https://stackoverflow.com/a/5286571) at <https://stackoverflow.com/a/5286571>. This function is not claimed under the laforge license.

CONTRIBUTING TO DEVELOPMENT

laforge supports Python 3.6+.

Process	Tool	Documentation
Automation	Nox	https://nox.readthedocs.io/
Test	pytest	https://docs.pytest.org/
Test coverage	pytest-cov	https://pytest-cov.readthedocs.io/
Format	Black	https://black.readthedocs.io/
Lint	Flake8	http://flake8.pycqa.org/
List more	Pylint	https://pylint.readthedocs.io/en/latest/
Document	Sphinx	https://www.sphinx-doc.org/

4.1 Suggested Environment

```
# Create virtual environment
python -m venv .venv

# Activate virtual environment with shell-specific script:
. .venv/bin/activate.fish      # fish
# $ source ./.venv/bin/activate # bash
# source ./.venv/bin/activate.csh # csh
# Note that Python for Windows creates ./Scripts/ rather than ./bin/
# .\.venv\Scripts\Activate.ps1  # PowerShell
# .venv\Scripts\Activate.bat    # cmd

# Install packages
python -m pip install -r requirements.txt

# Optional [packages] to include Excel and/or non-SQLite databases
python -m pip install -e .[mysql]

# Run tests
python -m pytest

# Run the gauntlet
python -m nox
```

4.2 Embedded TODOs

Todo: Implement cache_results=False

[original entry](#)

Todo: Restore quiet?

[original entry](#)

Todo:

if “:” in self.content: previous_result_key, actual_path_content = self.content.split(“:”)

[original entry](#)

Todo: De-messify

[original entry](#)

Todo: Factor out to superclass to allow views

[original entry](#)

Todo: class InvalidIdentifierError relay_id_problem(identifier, action, reason=None, replacement=None)

[original entry](#)

4.3 Docstring Gaps

```
Undocumented Python objects
=====
laforge.builder
-----
Functions:
* find_build_config_in_directory
* get_package_logger
* get_verb
* is_verb
* run_cmd
* seconds_since

Classes:
* BaseTask -- missing methods:

    - implement
```

(continues on next page)

(continued from previous page)

```

    - validate_results
* ExistenceChecker -- missing methods:

    - implement
* FileReader -- missing methods:

    - implement
* FileWriter -- missing methods:

    - implement
    - write
* InternalPythonExecutor -- missing methods:

    - implement
* SQLExecutor -- missing methods:

    - implement
* SQLQueryReader -- missing methods:

    - implement
* SQLReaderWriter -- missing methods:

    - implement
* ShellExecutor -- missing methods:

    - implement
* StataExecutor -- missing methods:

    - implement
* Task
* TaskList -- missing methods:

    - load_tasks
    - template_content

```

```
laforge.command
```

```
-----
```

```
Functions:
```

```
* user_confirms_cleartext
```

```
laforge.distros
```

```
-----
```

```
Classes:
```

```
* Distro
* MSSQL
* MySQL
* PostgreSQL
* SQLite

```

```
laforge.sql
```

```
-----
```

```
Functions:
```

```
* fix_bad_columns
```

```
Classes:
```

```
* Channel -- missing methods:

```

(continues on next page)

(continued from previous page)

```
- clean_up_statement
- find
- grab
- retrieve_engine
- save_engine
* Identifier -- missing methods:

- check
* Script -- missing methods:

- read
* Table -- missing methods:

- exists
- resolve

laforge.tech
-----
Functions:
* capitalize_sentences
* nobabble

Classes:
* ModifiableVerb
* Technobabbler

laforge.toolbox
-----
Functions:
* is_reserved_word
```

FREE SOFTWARE LICENSE

Copyright 2019 Matt VanEseltine.

laforge is free software: you can redistribute it and/or modify it under the terms of the GNU Affero General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

laforge is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Affero General Public License for more details. Copies are attached with this documentation and available online at <https://www.gnu.org/licenses/agpl.html>.

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

PYTHON MODULE INDEX

I

- `laforge.builder`, 7
- `laforge.command`, 8
- `laforge.distros`, 8
- `laforge.sql`, 8
- `laforge.tech`, 10
- `laforge.toolbox`, 10

B

BaseTask (class in *laforge.builder*), 7

C

Channel (class in *laforge.sql*), 9

D

drop() (*laforge.sql.Table* method), 9

dry_run() (*laforge.builder.TaskList* method), 7

E

execute() (in module *laforge.sql*), 9

execute() (*laforge.builder.TaskList* method), 7

execute() (*laforge.sql.Script* method), 9

execute_statement() (*laforge.sql.Channel* method), 9

ExistenceChecker (class in *laforge.builder*), 8

F

FileCall (class in *laforge.builder*), 7

FileReader (class in *laforge.builder*), 8

FileWriter (class in *laforge.builder*), 8

flatten() (in module *laforge.toolbox*), 10

I

Identifier (class in *laforge.sql*), 9

InternalPythonExecutor (class in *laforge.builder*), 8

K

kwargs() (*laforge.builder.FileCall* property), 7

L

laforge.builder (module), 7

laforge.command (module), 8

laforge.distro (module), 8

laforge.sql (module), 8

laforge.tech (module), 10

laforge.toolbox (module), 10

load_section_config() (*laforge.builder.TaskList* method), 7

M

make_first_upper() (in module *laforge.tech*), 10

method() (*laforge.builder.FileCall* property), 7

P

path() (*laforge.builder.BaseTask* property), 8

prepare_to_access() (in module *laforge.toolbox*), 10

R

read() (*laforge.sql.Table* method), 9

round_up() (in module *laforge.toolbox*), 10

run_build() (in module *laforge.builder*), 7

S

Scalar (class in *laforge.sql*), 9

Script (class in *laforge.sql*), 9

ShellExecutor (class in *laforge.builder*), 8

show_env() (in module *laforge.builder*), 7

SQLChannelNotFound, 8

SQLDistroNotFound, 8

SQLExecutor (class in *laforge.builder*), 8

SQLIdentifierProblem, 8

SQLQueryReader (class in *laforge.builder*), 8

SQLReaderWriter (class in *laforge.builder*), 8

SQLTableNotFound, 8

StataExecutor (class in *laforge.builder*), 8

T

Table (class in *laforge.sql*), 9

Target (class in *laforge.builder*), 7

TaskConstructionError, 7

TaskExecutionError, 7

TaskList (class in *laforge.builder*), 7

to_table() (*laforge.sql.Script* method), 9

V

Verb (class in *laforge.builder*), 7

verify_file_is_writable() (in module *laforge.toolbox*), 10

W

`write()` (*laforge.sql.Table* method), 9