
Emacs Python Environment Documentation

Release 0.0.2alpha0

Takafumi Arakaki

June 23, 2016

1	End user configuration	3
2	API functions	5
3	Indices and tables	7

Links:

- [Repository](#) (at GitHub)
- [Issue tracker](#) (at GitHub)
- [Travis CI](#)
- [MELPA](#)
- [MELPA Stable](#)

Emacs integrates well with external tools written in languages other than Emacs Lisp and thus use of these tools should be encouraged. However, many people try to avoid using non-Emacs Lisp software tools since it makes installation of their Emacs plugin hard. `python-environment.el` solves this problem (only for the case the tool is written in Python) by providing `virtualenv` API in Emacs Lisp so that you can automate installation of tools written in Python.

Let's say you write Emacs plugin which uses Python script. Then you would want to setup Python modules for that script easily from Emacs. This is how to do that using `python-environment.el`

```
(require 'python-environment)

(defun YOUR-PLUGIN-install-python-dependencies ()
  (interactive)
  (python-environment-run "pip" "install" "epc"))
```

In your plugin, if you want the path to command installed in the `virtualenv`, do something like this

```
(start-process NAME BUFFER
              (python-environment-bin "COMMAND")
              ...)
```

Path to access `COMMAND` may be different in `*nix` and in Windows, but `python-environment.el` finds the right path.

End user configuration

The following variables are set by end users. Emacs plugin authors *must not* set or let-bind these variables.

Variable `python-environment-directory`

Path to directory to store all Python virtual environments. A string.

If you want to change the location to, say ```~/python-environments```, then set it like this in your Emacs setup file::

```
(setq python-environment-directory ``~/python-environments'')
```

Variable `python-environment-default-root-name`

Default Python virtual environment name. A string.

This is a name of directory relative to `python-environment-directory` where default virtual environment locates.

Thus, typically the default virtual environment path is ```~/emacs.d/python-environments/default```.

Variable `python-environment-virtualenv`

`virtualenv` command to use, including command options. List of strings.

For example, if you want to use specific Python executable (to specify Python version), append ```--python``` option like this::

```
(setq python-environment-virtualenv
      (append python-environment-virtualenv
              (``--python'' ``PATH/TO/bin/python'')))
```

I added ```--system-site-packages``` as default, but this is not mandatory. If you dont like it, removing does not break anything (well, theoretically). For reason why it is default, see discussion here:

<https://github.com/tkf/emacs-python-environment/issues/3>

API functions

Function `python-environment-make` (&optional root, virtualenv)

Make virtual environment at ROOT asynchronously.

This function does not wait until `virtualenv` finishes. Instead, it returns a deferred object `[#]_`. So, if you want to do some operation after the `virtualenv` command finishes, do something like this::

```
(deferred:$
  (python-environment-make)
  (deferred:nextc it (lambda (output) DO-SOMETHING-HERE)))
```

If ROOT is specified, it is used instead of `python-environment-default-root-name`. ROOT can be a relative path from `python-environment-virtualenv` or an absolute path.

If VIRTUALENV (list of string) is specified, it is used instead of `python-environment-virtualenv`.

.. [#] <https://github.com/kiwanami/emacs-deferred>

Function `python-environment-make-block` (&optional root, virtualenv)

Blocking version of `python-environment-make`.

I recommend NOT to use this function in interactive commands.

For reason, see `python-environment-run-block`

Function `python-environment-run` (command &optional , root, virtualenv)

Run COMMAND installed in Python virtualenv located at ROOT asynchronously.

Instead of waiting for COMMAND to finish, a deferred object `[#]_` is returned so that you can chain operations.

See `python-environment-make` for how ROOT and VIRTUALENV are interpreted and how to work with deferred object.

Use `python-environment-run-block` if you want to wait until the command exit (NOT recommended in interactive command).

.. [#] <https://github.com/kiwanami/emacs-deferred>

Function `python-environment-run-block` (command **&optional** , root, virtualenv)

Blocking version of `python-environment-run`.

I recommend NOT to use this function in interactive commands.

Emacs users have more important things to than waiting for some command to finish.

Function `python-environment-exists-p` (**&optional** root)

Return non-`nil` if virtualenv at ROOT exists.

See `python-environment-make` for how ROOT is interpreted.

Function `python-environment-bin` (path **&optional** , root)

Return full path to ```ROOT/bin/PATH``` or ```ROOT/Scripts/PATH``` if exists.

```Scripts``` is used instead of ```bin``` in typical Windows case.

In Windows, path with extension `".ext"` may be returned.

See `python-environment-make` for how ROOT is interpreted.

**Function `python-environment-lib`** (path **&optional** , root)

Return full path to ```ROOT/lib/PATH``` or ```ROOT/Lib/PATH``` if exists.

```Lib``` is used instead of ```lib``` in typical Windows case.

See `python-environment-make` for how ROOT is interpreted.

Indices and tables

- `genindex`
- `modindex`
- `search`

E

Emacs Lisp function

- python-environment-bin, 6
- python-environment-exists-p, 6
- python-environment-lib, 6
- python-environment-make, 5
- python-environment-make-block, 5
- python-environment-run, 5
- python-environment-run-block, 5

Emacs Lisp variable

- python-environment-default-root-name, 3
- python-environment-directory, 3
- python-environment-virtualenv, 3

P

python-environment-bin

- Emacs Lisp function, 6

python-environment-default-root-name

- Emacs Lisp variable, 3

python-environment-directory

- Emacs Lisp variable, 3

python-environment-exists-p

- Emacs Lisp function, 6

python-environment-lib

- Emacs Lisp function, 6

python-environment-make

- Emacs Lisp function, 5

python-environment-make-block

- Emacs Lisp function, 5

python-environment-run

- Emacs Lisp function, 5

python-environment-run-block

- Emacs Lisp function, 5

python-environment-virtualenv

- Emacs Lisp variable, 3