Python svlib

1 What is python-svlib ?

`svlib` is a python[1] library of utility code that is not application-specific. `svlib` contains code that a lot of my programs use.

Most of the code re-uses existing UNIX or python standard library code, but makes it easier to use.

2 Features

Some of the things implemented are:

- Abstraction of an "atomic file", i.e a file whose content is either in one state or another, but never in between.

- Lock shared by every single running program

- Reading/writing various kind of special-purpose files, like

  - Files containing only a timestamp
  - Files containing only a number
  - Files containing a list of file system paths
  - Files containing key/value pairs

It might be that other libraries do this in a much more efficient, straightforward, or just in the same way. Maybe I am not good at reading manuals or searching the web. You should really tell me if you think such or such piece of code is really duplicated and is much better elsewhere.

But then sometimes it takes much more time to read a manual, download and install a library than to just implement a feature in 10 lines. Moreover, it can be a good programming exercise.

3 Download

Python-svlib is available both as a Pacman[2] package (x86_64 only), or in source form. For source or Pacman installation, please refer to the Generic download and install instructions[3].

---

[1] http://www.python.org/

[2] http://www.archlinux.org/pacman/

[3] http://svasey.org/projects/download-install-doc_en.html

- Latest source tarball[4]

- PGP signature[5]

4   Usage

You should really dive into the source. I have tried to document every function, and there are plenty of usage examples in my programs.

`python-svlib` also installs one very simple end-user program, `switchlink`. This atomically changes where a symbolic link points to. Its first argument is the symbolic link to change (or create), and the second is where it must point to.

The program creates some temporary files during the operation, and removes them afterward. By default, those are created in the symlink's directory. The `--tempdir=DIR` option can be used in the rare case where you want to change this. Note that the temporary files must be created on the same filesystem (partition) as the symbolic link.

---

[4]`http://svasey.org/public-repo.svasey.org/src/python-svlib.tar.gz`
[5]`http://svasey.org/public-repo.svasey.org/src/python-svlib.tar.gz.sig`