desub Documentation

Release 1.0

Kumar McMillan and contributors

September 01, 2012

CONTENTS

desub is a Python module to work with a detached subprocess.

This is useful for making a web interface (or whatever) to monitor a process running in the background. You can start a process from a web request and serve subsequent requests to report status on and show output for the process you started earlier.

Think of it as a way to monitor a long running command without having to manage PIDs yourself. You can either start or join a command (same executable, same args). Desub is not designed for running multiple instances of the same command. You can just use a regular subprocess for that.

CHAPTER

ONE

INSTALLATION

```
pip install desub
```

This pulls in psutil as a requirement.

CHAPTER

TWO

API

desub.**join**(*cmd_args*, ***kw*)

Join a subprocess or start one if it's not running.

The return value is a desub.Desub object.

Start by joining a command that may or may not already be running.

```
>>> proc = desub.join(['python', 'tests/cmd/loop.py'])
```

There will only ever be one instance of that command running unless you specify a custom *root*. Check to see if the command is running:

```
>>> proc.is_running()
False
```

OK. It's not running, so start it up:

```
>>> proc.start()
>>> proc.is_running()
True
>>> type(proc.pid)
<type 'int'>
```

Fetch the output:

```
>>> proc.stdout.read()
'stdout'
>>> proc.stderr.read()
'stderr'
```

Stop the command, if you want to:

```
>>> proc.stop()
>>> proc.is_running()
False
```

When you read the output of a stopped command, that's the output from a subsequent run.

```
>>> proc.stdout.read()
'stdout'
```

class desub.**Desub**(*cmd_args*, ***kw*)

A detached subprocess.

This object represents a command with **cmd_args** that may or may not be running yet. Two processes can be running at the same time but only if their args are different or if their **roots** are different.

The interface is similar to subprocess.Popen when appropriate. All extra keyword arguments are passed into the Popen constructor.

Keyword arguments

root The root directory to store subprocess artifacts like a PID file and stdout/stderr logs. By default this is `~/.desub`.

Members

pid
The integer PID of the subprocess or None.

stdout
An open read-only file to stdout.

stderr
An open read-only file to stderr.

start()
Start the subprocess.

stop()
Stop the subprocess.

is_running()
True if the subprocess is running.

CHAPTER

THREE

DEVELOPERS

Hello! To work on this module, check out the source from git and be sure you have the tox tool. To run the test suite, cd into the root and type:

```
tox
```

This will run all tests in a virtualenv using the supported versions of Python.

To the build the docs run:

```
pip install -r docs/requirements.txt
make -C docs html
```

The issue tracker can be found on github.

CHAPTER

FOUR

INDICES AND TABLES

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