PyREM Documentation

Release 0.1.0

Ellis Michael

November 12, 2015

Contents

Contents:

CHAPTER 1

PyREM package

1.1 pyrem.task module

task.py: Contains the main unit of execution in PyREM, the task.

class `pyrem.task.`**`Task`**
Bases: `object`

Abstract class, the main unit of execution in PyREM.

If you would like to define your own type of `Task`, you should at least implement the `_start`, `_wait`, `_stop`, and `_reset` methods.

Every task that gets started will be stopped on Python exit, as long as that exit can be caught by the `atexit` module (e.g. pressing *Ctrl+C* will be caught, but sending *SIGKILL* will not be caught).

`return_values`
dict

Subclasses of `Task` should store all of their results in this field and document what the possible return values are.

`start`(*wait=False*)
Start a task.

This function depends on the underlying implementation of _start, which any subclass of `Task` should implement.

Parameters **`wait`** (*bool*) – Whether or not to wait on the task to finish before returning from this function. Default *False*.

Raises `RuntimeError` – If the task has already been started without a subsequent call to `reset()`.

`wait`()
Wait on a task to finish and stop it when it has finished.

Raises `RuntimeError` – If the task hasn't been started or has already been stopped.

Returns The `return_values` of the task.

`stop`()
Stop a task immediately.

Raises `RuntimeError` – If the task hasn't been started or has already been stopped.

reset()
Reset a task.

Allows a task to be started again, clears the `return_values`.

Raises `RuntimeError` – If the task has not been stopped.

class pyrem.task.**SubprocessTask**(*command*, *quiet=False*, *return_output=False*, *shell=False*, *require_success=False*)
Bases: *pyrem.task.Task*

A task to run a command as a subprocess on the local host.

This process will be killed when this task is stopped. The return code of the process will be stored in `return_values['retcode']`.

Parameters

- **command** (*list of str*) – The command to execute.
- **quiet** (*bool*) – If *True*, the output of this command is not printed. Default *False*.
- **return_output** (*bool*) – If *True*, the output of this command will be saved in `return_values['stdout']` and `return_values['stderr']` when the subprocess is allowed to finish (i.e. when it is waited on instead of being stopped). Default *False*.

  quiet and **return_output** shouldn't both be true.
- **shell** (*bool*) – If *True*, allocate a shell to execute the process. See: `subprocess.Popen`. Default *False*.
- **require_success** (*bool*) – If *True* and if this task is waited on instead of being stopped, raises a `RuntimeError` if the subprocess has a return code other than *0*. Default *False*.

class pyrem.task.**RemoteTask**(*host*, *command*, *quiet=False*, *return_output=False*, *kill_remote=True*)
Bases: *pyrem.task.SubprocessTask*

A task to run a command on a remote host over ssh.

Any processes started on the remote host will be killed when this task is stopped (unless *kill_remote=False* is specified).

`return_values['retcode']` will contain the return code of the ssh command, which should currently be ignored.

Parameters

- **host** (*str*) – The host to run on.
- **command** (*list of str*) – The command to execute.
- **quiet** (*bool*) – See `SubprocessTask`.
- **return_output** (*bool*) – See `SubprocessTask`.
- **kill_remote** (*bool*) – If *True*, all processes started on the remote server will be killed when this task is stopped.

class pyrem.task.**Parallel**(*tasks*)
Bases: *pyrem.task.Task*

A task that executes several given tasks in parallel.

Currently does not capture the return_values of the underlying tasks, this will be fixed in the future.

Parameters **tasks** (list of `Task`) – Tasks to execute.

class pyrem.task.**Sequential**(*tasks*)
Bases: *pyrem.task.Task*

A tasks that executes several given tasks in sequence.

Currently does not capture the return_values of the underlying tasks, this will be fixed in the future.

Parameters **tasks** (list of Task) – Tasks to execute.

1.2 pyrem.host module

host.py: Contains classes for managing remote hosts.

The Host object is a simple wrapper around various sorts of Tasks.

class pyrem.host.**Host**(*hostname*)
Bases: object

Abstract class, an object representing some host.

hostname
str

The name of the host.

run(*command*, ***kwargs*)
Build a task to run the command on a remote host.

Parameters

- **command** (*list of str*) – The command to execute.
- ****kwargs** – Keyword args to be passed to the underlying Task's init method.

Returns The resulting task.

Return type pyrem.task.Task

class pyrem.host.**RemoteHost**(*hostname*)
Bases: *pyrem.host.Host*

A remote host.

Parameters **hostname** (*str*) – The hostname of the remote host.

run(*command*, ***kwargs*)
Run a command on the remote host.

This is just a wrapper around RemoteTask(self.hostname, ...)

send_file(*file_name*, *remote_destination=None*, ***kwargs*)
Send a file to a remote host with rsync.

Parameters

- **file_name** (*str*) – The relative location of the file on the local host.
- **remote_destination** (*str*) – The destination for the file on the remote host. If *None*, will be assumed to be the same as **file_name**. Default *None*.
- ****kwargs** – Passed to SubprocessTask's init method.

Returns The resulting task.

Return type pyrem.task.SubprocessTask

get_file (*file_name*, *local_destination=None*, ***kwargs*)
Get a file from a remote host with rsync.

Parameters

- **file_name** (*str*) – The relative location of the file on the remote host.
- **local_destination** (*str*) – The destination for the file on the local host. If *None*, will be assumed to be the same as **file_name**. Default *None*.
- ****kwargs** – Passed to `SubprocessTask`'s init method.

Returns The resulting task.

Return type `pyrem.task.SubprocessTask`

class `pyrem.host.`**LocalHost**
Bases: *pyrem.host.Host*

The local host.

run (*command*, ***kwargs*)

move_file (*file_name*, *destination*, ***kwargs*)
Move a file on the local host.

Parameters

- **file_name** (*str*) – The relative location of the file.
- **destination** (*str*) – The relative destination of the file.
- ****kwargs** – Passed to `SubprocessTask`'s init method.

Returns The resulting task.

Return type `pyrem.task.SubprocessTask`

1.3 pyrem.utils module

utils.py: Contains useful utilities to be used in other modules.

`pyrem.utils.`**synchronized** (*func*)
Function decorator to make function synchronized on `self._lock`.

If the first argument to the function (hopefully self) does not have a _lock attribute, then this decorator does nothing.

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

p

G

H

L

M

P

R

S

T

W