stdio-mgr

Release 2.0.dev1

Brian Skinn

Sep 13, 2019

CONTENTS

1 User's Guide 3

2 API 5

3 Inheritance Diagram 9

4 Indices and tables 11

Python Module Index 13

Index 15

Intro stuff to go here

CHAPTER

ONE

USER'S GUIDE

There will be things here…

CHAPTER

TWO

API

API page.

Definitely need to explain here what the public API is.

Anticipate having some of the private API presented here, but have to make it clear what's private vs public.

`stdio_mgr` *code module*.

`stdio_mgr` provides a context manager for convenient mocking and/or wrapping of `stdin`/`stdout`/`stderr` interactions.

Author Brian Skinn (bskinn@alum.mit.edu)

File Created 24 Mar 2018

Copyright (c) Brian Skinn 2018-2019

Source Repository http://www.github.com/bskinn/stdio-mgr

Documentation See README.rst at the GitHub repository

License The MIT License; see LICENSE.txt for full license terms

Members

class `stdio_mgr.stdio_mgr.`**`RandomTextIO`**

Class to capture writes to a buffer even when detached.

Subclass of `TextIOWrapper` that utilises an internal buffer defaulting to utf-8 encoding.

As a subclass of `TextIOWrapper`, it is not thread-safe.

All writes are immediately flushed to the buffer.

This class provides *`getvalue()`* which emulates the behavior of `StringIO.getvalue()`, decoding the buffer using the `encoding`. The value is available even if the stream is detached or closed.

`getvalue`()

Obtain buffer of text sent to the stream.

`write`(**args*, ***kwargs*)

Flush after each write.

class `stdio_mgr.stdio_mgr.`**`SafeCloseRandomTextIO`**

Class to capture writes to a buffer even when detached, and safely close.

Subclass of *`_SafeCloseIOBase`* and *`RandomTextIO`*.

class `stdio_mgr.stdio_mgr.`**`SafeCloseTeeStdin`**(*tee*, **args*, ***kwargs*)

Class to tee contents to a side buffer on read, and safely close.

Subclass of *`_SafeCloseIOBase`* and *`TeeStdin`*.

class stdio_mgr.stdio_mgr.**SimulateStdin**(*init_text=''*, *encoding='utf-8'*)

Class to simulate content appearing on stdin.

Subclass of `TextIOWrapper` that provides *`getvalue()`* which emulates the behavior of `StringIO.getvalue()`, decoding the buffer using the `encoding`.

This class also provides the method *`append()`*, which is not available for the base `TextIOWrapper` type. This method adds new content to the end of the stream while leaving the read position unchanged.

As a subclass of `TextIOWrapper`, it is not thread-safe.

Instantiation takes two arguments:

init_text

`str` *(optional)* – Text to use as the initial contents of the stream to be teed. Default is an empty `str`.

encoding

`str` *(optional)* – Encoding for the underlying `TextIOWrapper`. Default is "utf-8".

append(*text*)

Write to end of stream while maintaining seek position.

Actually stores the current position; seeks to end; writes *text*; and seeks to prior position.

Parameters **text** – `str` – Text to append to the current stream contents.

getvalue()

Obtain pending buffer of text for stdin.

class stdio_mgr.stdio_mgr.**StdioManager**

Substitute temporary text buffers for *stdio* in a managed context.

Context manager.

Substitutes empty *`RandomTextIO`*s for `sys.stdout` and `sys.stderr`, and a *`TeeStdin`* for `sys.stdin` within the managed context.

Upon exiting the context, the original stream objects are restored within `sys`, and the temporary streams are closed.

Parameters **in_str** – `str` *(optional)* – Initialization text for the *`TeeStdin`* substitution for *stdin*. Default is an empty string.

Yields

- *in_* – *`TeeStdin`* – Temporary stream for *stdin*.
- *out_* – *`RandomTextIO`* – Temporary stream for *stdout*, initially empty.
- *err_* – *`RandomTextIO`* – Temporary stream for *stderr*, initially empty.

class stdio_mgr.stdio_mgr.**TeeStdin**(*tee*, **args*, ***kwargs*)

Class to tee simulated contents to a side buffer on read.

Subclass of *`SimulateStdin`* and *`_Tee`* that simulates a stdin stream while teeing all content *read* from the stream to *tee*.

To emphasize: teeing occurs on content *read*, **not write**..

As a subclass of `TextIOWrapper`, it is not thread-safe.

class stdio_mgr.stdio_mgr.**_PersistedBytesIO**(*closure_callback*)

Class to persist the value after close.

A copy of the bytes value is given to a callback prior to the `close()`.

close()
Send buffer to callback and close.

class stdio_mgr.stdio_mgr.**_SafeCloseIOBase**
Class to ignore ValueError when exiting the context.

Subclass of `TextIOBase` that disregards ValueError, which can occur if the file has already been closed, when exiting the context.

class stdio_mgr.stdio_mgr.**_Tee**(*tee*, **args*, ***kwargs*)
Class to tee contents to a side buffer on read.

Subclass of `TextIOWrapper` that overrides `read()` and `readline()` to tee all content *read* from the stream to *tee*.

To emphasize: teeing occurs on content *read*, **not write**.

As a subclass of `TextIOWrapper`, it is not thread-safe.

Instantiation takes an additional argument:

tee

`TextIOBase` – Text stream to write content of each read

read(*size=None*)
Tee text to side buffer when read.

Overrides `TextIOWrapper.read()` to implement the teeing.

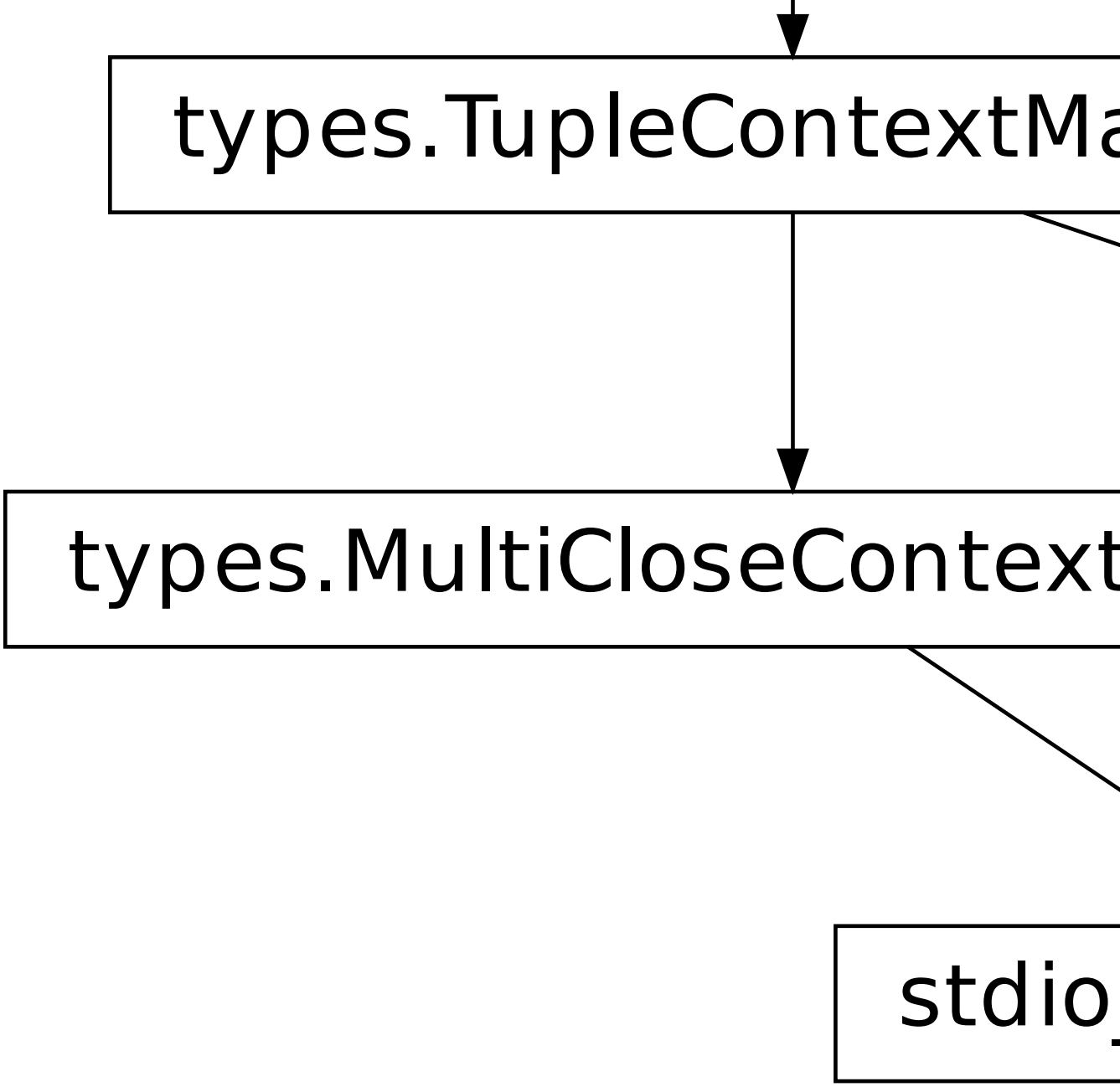
Parameters **size** – `int` or `None` *(optional)* – Number of characters to return; a negative or `None` value reads to EOF.

readline(*size=-1*)
Tee text to side buffer when read.

Overrides `TextIOWrapper.readline()` to implement the teeing.

Parameters **size** – `int` *(optional)* – Number of characters to return; a negative value reads an entire line, regardless of length

stdio_mgr.stdio_mgr.**stdio_mgr**
alias of *stdio_mgr.stdio_mgr.StdioManager*


types.TupleContextMa
types.MultiCloseContext
stdio

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

S

stdio_mgr.stdio_mgr, 5

INDEX

Symbols

`_PersistedBytesIO` (*class in stdio_mgr.stdio_mgr*), 6
`_SafeCloseIOBase` (*class in stdio_mgr.stdio_mgr*), 7
`_Tee` (*class in stdio_mgr.stdio_mgr*), 7

A

`append()` (*stdio_mgr.stdio_mgr.SimulateStdin method*), 6

C

`close()` (*stdio_mgr.stdio_mgr._PersistedBytesIO method*), 6

G

`getvalue()` (*stdio_mgr.stdio_mgr.RandomTextIO method*), 5
`getvalue()` (*stdio_mgr.stdio_mgr.SimulateStdin method*), 6

R

`RandomTextIO` (*class in stdio_mgr.stdio_mgr*), 5
`read()` (*stdio_mgr.stdio_mgr._Tee method*), 7
`readline()` (*stdio_mgr.stdio_mgr._Tee method*), 7

S

`SafeCloseRandomTextIO` (*class in stdio_mgr.stdio_mgr*), 5
`SafeCloseTeeStdin` (*class in stdio_mgr.stdio_mgr*), 5
`SimulateStdin` (*class in stdio_mgr.stdio_mgr*), 5
`stdio_mgr` (*in module stdio_mgr.stdio_mgr*), 7
`stdio_mgr.stdio_mgr` (*module*), 5
`StdioManager` (*class in stdio_mgr.stdio_mgr*), 6

T

`TeeStdin` (*class in stdio_mgr.stdio_mgr*), 6

W

`write()` (*stdio_mgr.stdio_mgr.RandomTextIO method*), 5