

---

# RepoBuddy Documentation

*Release 0.3beta*

**Ash <tuxdude.github@gmail.com>**

**Apr 28, 2017**



---

## Contents

---

<b>1</b>	<b>Introduction to RepoBuddy</b>	<b>3</b>
<b>2</b>	<b>Code Documentation</b>	<b>5</b>
2.1	repobuddy – Root package . . . . .	5
2.2	repobuddy.arg_parser - Argument Parser . . . . .	5
2.3	repobuddy.client_info – Client Info . . . . .	6
2.4	repobuddy.command_handler - Command Handler . . . . .	7
2.5	repobuddy.git_wrapper - Git Wrapper . . . . .	8
2.6	repobuddy.globals - Global Definitions . . . . .	10
2.7	repobuddy.main - Program’s Main Routine . . . . .	10
2.8	repobuddy.manifest_parser - Manifest Parser . . . . .	10
2.9	repobuddy.utils - Utility classes and functions . . . . .	12
<b>3</b>	<b>Test Code Documentation</b>	<b>15</b>
3.1	repobuddy.tests – Root package for tests . . . . .	15
3.2	repobuddy.tests.arg_parser – Argument Parser tests . . . . .	15
3.3	repobuddy.tests.client_info – Client Info tests . . . . .	16
3.4	repobuddy.tests.command_handler – Command Handler tests . . . . .	16
3.5	repobuddy.tests.common – Test Framework . . . . .	17
3.6	repobuddy.tests.git_wrapper – Git Wrapper tests . . . . .	18
3.7	repobuddy.tests.main – Main routine for the tests . . . . .	19
3.8	repobuddy.tests.manifest_parser – Manifest Parser tests . . . . .	19
3.9	repobuddy.tests.utils – Utilities tests . . . . .	20
<b>4</b>	<b>Indices and tables</b>	<b>21</b>
	<b>Python Module Index</b>	<b>23</b>



Contents:



# CHAPTER 1

---

## Introduction to RepoBuddy

---

Multi-repository management tool for GIT

An easy alternative to git sub-modules, to configure and maintain multiple Git repositories.





## repobuddy – Root package

repobuddy module.

## repobuddy.arg\_parser - Argument Parser

**exception** repobuddy.arg\_parser.ArgParserError (*error\_str=None, exit\_prog\_without\_error=False*)  
Bases: *repobuddy.utils.RepoBuddyBaseException*

Exception raised by *ArgParser*.

**Variables** **exit\_prog\_without\_error** – Set to True if *ArgParser* completed parsing the command line arguments without any errors, otherwise False.

**\_\_init\_\_** (*error\_str=None, exit\_prog\_without\_error=False*)  
Initializer.

### Parameters

- **error\_str** (*str*) – The error string to store in the exception.
- **exit\_prog\_without\_error** (*Boolean*) – If True indicates that the program exited without any errors, otherwise not. The instance variable *exit\_prog\_without\_error* stores this value.

**class** repobuddy.arg\_parser.ArgParser (*handlers*)  
Bases: *object*

Parses command line arguments for repobuddy.

**\_\_init\_\_** (*handlers*)  
Initializer.

**Parameters** `handlers` (*dict*) – A dictionary with command names as keys and the handler functions as values.

**Returns** None

**parse** (*args*)

Parse the command line arguments to repobuddy.

**Parameters** `args` (*list of strings*) – List of command line arguments.

**Returns** None

**Raises** `ArgParserError` on parsing errors.

## repobuddy.client\_info – Client Info

**exception** repobuddy.client\_info.**ClientInfoError** (*error\_str*)

Bases: `repobuddy.utils.RepoBuddyBaseException`

Exception raised by `ClientInfo`.

**\_\_init\_\_** (*error\_str*)

Initializer.

**Parameters** `error_str` (*str*) – The error string to store in the exception.

**class** repobuddy.client\_info.**ClientInfo** (*config\_file\_name=None*)

Bases: `object`

Parses/Stores/Retrieves the client configuration.

**\_\_init\_\_** (*config\_file\_name=None*)

Initializer.

**Parameters** `config_file_name` (*str*) – The name of the config file. If `config_file_name` is set to None, the configuration is just stored in-memory until `write()` is invoked. Instead if `config_file_name` is specified, the config file is opened, parsed and the instance represents the state of the config file.

**Raises** `ClientInfoError` when `config_file_name` is not None and any of the following conditions are met:

- Failed to open the config file.
- Parsing errors have been detected.
- Validating the config failed.

**set\_client\_spec** (*client\_spec\_name*)

Set the client\_spec in the config.

**Parameters** `client_spec_name` (*str*) – The value for client\_spec in the config.

**Returns** None

**Raises** `ClientInfoError` if the config does not already have the `RepoBuddyClientInfo` section.

**set\_manifest** (*manifest\_xml*)

Set the manifest in the config.

**Parameters** `manifest_xml` (*str*) – The value for manifest in the config.

**Returns** None

**Raises** *ClientInfoError* if the config does not have the RepoBuddyClientInfo section.

**get\_client\_spec()**

Get the value of `client_spec` in the config.

**Returns** The value of `client_spec` in the config.

**Return type** *str*

**Raises** *ClientInfoError* if the config does not have the `client_spec` option.

**get\_manifest()**

Get the value of `manifest` in the config.

**Returns** The value of `manifest` in the config.

**Return type** *str*

**Raises** *ClientInfoError* if the config does not have the `manifest` option.

**write(file\_name=None)**

Write the config to a file.

If `file_name` is set to `None`, the filename passed during the class initialization is used instead. If there was a file name specified during both initialization, as well as in the parameter `file_name`, this method's parameter takes precedence, and the file name specified during initialization remains unmodified.

**Parameters** `file_name` (*str*) – The name of the file to write the config into.

**Returns** `None`

**Raises** *ClientInfoError* when any of the following conditions are met:

- `file_name` parameter is `None` and no file name was provided during initialization.
- validation of the config failed.
- writing the config to the file failed.

## repopuddy.command\_handler - Command Handler

**exception** `repopuddy.command_handler.CommandHandlerError` (*error\_str*)

Bases: *repopuddy.utils.RepoBuddyBaseException*

Exception raised by *CommandHandler*.

**\_\_init\_\_(error\_str)**

**class** `repopuddy.command_handler.CommandHandler`

Bases: *object*

Provides handlers for the `repopuddy` commands.

**\_\_init\_\_()**

Initializer.

**get\_handlers()**

Get the command handlers.

**Returns** Dictionary with command names as keys and the methods as values.

**Return type** *dict*

**init\_command\_handler**(args)  
Handler for the init command.

**Returns** None

**Raises** *CommandHandlerError* on errors.

**status\_command\_handler**(args)  
Handler for the status command.

**Returns** None

**Raises** *CommandHandlerError* on errors.

## repobuddy.git\_wrapper - Git Wrapper

**exception** repobuddy.git\_wrapper.**GitWrapperError**(error\_str, is\_git\_error, git\_error\_msg='')  
Bases: *repobuddy.utils.RepoBuddyBaseException*

Exception raised by *GitWrapper*.

**Variables** **is\_git\_error** – Set to True if *GitWrapper* got back a non-zero status after executing of any of the git commands, otherwise False.

**\_\_init\_\_**(error\_str, is\_git\_error, git\_error\_msg='')  
Initializer.

**Parameters** **error\_str** (*str*) – The error string to store in the exception.

**class** repobuddy.git\_wrapper.**GitWrapper**(base\_dir)  
Bases: *object*

Helper for invoking git.

Provides a way to access and/or control the state of a git repository. Internally, it executes git commands in the work-tree for various operations.

**\_\_init\_\_**(base\_dir)  
Initializer.

**Parameters** **base\_dir** (*str*) – Absolute path of the git repository work-tree.

**Returns** None

**Raises** *GitWrapperError* if base\_dir is not an absolute path.

**clone**(remote\_url, branch, dest\_dir)  
Clone a repo.

Executes git clone -b branch remote\_url dest\_dir. At the end of the clone operation, the working directory is changed to dest\_dir.

**Parameters**

- **remote\_url** (*str*) – URL of the repository.
- **branch** (*str*) – Branch to checkout after the clone.

**Dest\_dir** Destination path to store the cloned repository.

**Returns** None

**Raises** *GitWrapperError* if the git clone command fails.

**update\_index()**

Refresh the index.

Executes `git update-index -q --ignore-submodules --refresh`.

**Returns** None

**Raises** *GitWrapperError* if the `git update-index` command fails.

**get\_untracked\_files()**

Get a list of all untracked files in the repository.

Returns the files in `git ls-files --exclude-standard --others --` as a list.

**Returns** List of untracked files.

**Return type** list of str

**Raises** *GitWrapperError* if the `git ls-files` command fails.

**get\_unstaged\_files()**

Get a list of all unstaged files in the repository.

Returns the files in `git diff-files --name-status -r --ignore-submodules --` as a list.

**Returns** List of unstaged files.

**Return type** list of str

**Raises** *GitWrapperError* if the `git diff-files` command fails.

**get\_uncommitted\_staged\_files()**

Get a list of all uncommitted but staged files.

Returns the files in `git diff-index --cached --name-status -r --ignore-submodules`.

**Returns** List of uncommitted files in the staging area.

**Return type** list of str

**Raises** *GitWrapperError* if the `git diff-index` command fails.

**get\_current\_branch()**

Get the currently checked out branch.

**Returns** Currently checked out Branch name if HEAD points to a branch, otherwise None

**Return type** str

**Raises** *GitWrapperError* on errors.

**get\_current\_tag()**

Get the currently checked out tag.

**Returns** The tag name which is currently checked out, None otherwise. If the commit pointed by HEAD contains more than one tag, the returned tag name could be any one of those tags.

**Return type** str

**Raises** *GitWrapperError* on errors.

## repobuddy.globals - Global Definitions

**exception** repobuddy.globals.**HelpStringsError** (*error\_str*)

Bases: *repobuddy.utils.RepoBuddyBaseException*

Exception raised by *HelpStrings*.

**\_\_init\_\_** (*error\_str*)

Initializer.

**Parameters** **error\_str** (*str*) – The error string to store in the exception.

**class** repobuddy.globals.**HelpStrings**

Definition of various strings used by the rest of repobuddy.

**PROGRAM\_DESCRIPTION** = 'Multi-repo manager for Git'

**PROGRAM\_NAME** = 'repobuddy'

**PROGRAM\_VERSION** = '%(prog)s 0.3beta'

**MASTER\_PARSER\_ARG\_HELP** = 'Command to invoke'

**MASTER\_PARSER\_ARG\_TITLE** = 'Available Commands'

**INIT\_COMMAND\_HELP** = 'Init the current directory to set up the repos'

**INIT\_MANIFEST\_ARG** = 'The Manifest file to use for this client'

**INIT\_CLIENT\_SPEC\_ARG** = 'The Client Spec in the Manifest to use for this client'

**HELP\_COMMAND\_HELP** = 'Show usage details for a command'

**HELP\_COMMAND\_ARG** = 'Command to see the help message for'

**STATUS\_COMMAND** = 'Show status of the current client config'

## repobuddy.main - Program's Main Routine

repobuddy.main.**run\_repobuddy** ()

Invoke repobuddy with the command line arguments.

The application exits with status 1 on errors, 0 otherwise.

returns: None

## repobuddy.manifest\_parser - Manifest Parser

**exception** repobuddy.manifest\_parser.**ManifestParserError** (*error\_str*)

Bases: *repobuddy.utils.RepoBuddyBaseException*

Exception raised by *ManifestParser*.

**\_\_init\_\_** (*error\_str*)

**class** repobuddy.manifest\_parser.**Repo** (*url=None, branch=None, dest=None*)

Bases: *repobuddy.utils.EqualityBase*

Represents the Repository in the manifest.

**\_\_init\_\_** (*url=None, branch=None, dest=None*)  
Initializer.

**Parameters**

- **url** (*str*) – URL of the repository.
- **branch** (*str*) – Name of the branch to checkout.

**Dest** Destination directory.

**class** `repobuddy.manifest_parser.ClientSpec` (*name=None, repo\_list=None*)  
Bases: `repobuddy.utils.EqualityBase`

Represents the Client Spec in the manifest.

**\_\_init\_\_** (*name=None, repo\_list=None*)  
Initializer.

**Parameters**

- **name** (*str*) – Name of the client spec.
- **repo\_list** – List of Repositories in the manifest.

**Type** list of `Repo`

**class** `repobuddy.manifest_parser.Manifest` (*default\_client\_spec=None, client\_spec\_list=None*)  
Bases: `repobuddy.utils.EqualityBase`

Represents the manifest.

**\_\_init\_\_** (*default\_client\_spec=None, client\_spec\_list=None*)  
Initializer.

**Parameters**

- **default\_client\_spec** (*str*) – Default client spec.
- **client\_spec\_list** (list of `ClientSpec`) – List of client specs.

**class** `repobuddy.manifest_parser.ManifestParser`  
Bases: `object`

Helper class for parsing the manifest XML.

**\_\_init\_\_** ()  
Initializer.

**parse** (*file\_handle*)  
Parse the manifest from the stream.

**Parameters** **file\_handle** (*File object.*) – The stream to parse the manifest from.

**Returns** None

**Raises** `ManifestParserError` on errors.

**get\_manifest** ()  
Get the manifest.

**Returns** The parsed manifest.

**Return type** `Manifest`.

## repobuddy.utils - Utility classes and functions

**exception** repobuddy.utils.RepoBuddyBaseException (error\_str)

Bases: `exceptions.Exception`

Base class of all exceptions in repobuddy.

**\_\_init\_\_** (error\_str)

Initializer.

**Parameters** **error\_str** (*str*) – The error string to store in the exception.

**exception** repobuddy.utils.FileLockError (error\_str, is\_time\_out=False)

Bases: `repobuddy.utils.RepoBuddyBaseException`

Exception raised by `FileLock`.

**Variables** **is\_time\_out** – Set to `True` if a timeout occurred when trying to acquire the lock, `False` otherwise.

**\_\_init\_\_** (error\_str, is\_time\_out=False)

Initializer.

### Parameters

- **error\_str** (*str*) – The error string to store in the exception.
- **is\_time\_out** – If `True`, the error is because of a timeout in acquiring the lock. The `is_time_out` instance variable in the exception object is set to this value.

**class** repobuddy.utils.FileLock (file\_name, timeout=1, delay=0.1)

Bases: `object`

A mutual exclusion primitive using lock files.

**\_\_init\_\_** (file\_name, timeout=1, delay=0.1)

Initializer.

### Parameters

- **file\_name** (*str*) – Name of the lock file to be created. Filename can be either an absolute or a relative file path.
- **timeout** (*float*) – Maximum time in seconds until `acquire()` blocks in trying to acquire the lock. If timeout seconds have elapsed without successfully acquiring the lock, `FileLockError` is raised.
- **delay** (*float*) – Time interval in seconds between 2 successive lock attempts.

**acquire** ()

Acquire the lock.

Acquires the lock within the designated timeout, failing which it raises `FileLockError` with `is_time_out` set to `True`.

**Returns** `None`

**Raises** `FileLockError` on errors. `is_time_out` is set to `True` only if the designated timeout has elapsed.

**release** ()

Release the lock.

**Returns** `None`



**Raises** `FileLockError` on errors. If the lock file has already been deleted, no exception is raised.

**exception** `repobuddy.utils.ResourceHelperError` (`error_str`)

Bases: `repobuddy.utils.RepoBuddyBaseException`

Exception raised by `ResourceHelper`.

**\_\_init\_\_** (`error_str`)

Initializer.

**Parameters** `error_str` (`str`) – The error string to store in the exception.

**class** `repobuddy.utils.ResourceHelper`

A helper class for loading resources.

**classmethod** `open_data_file` (`package_name`, `file_name`)

Get a stream handle to the resource.

**Parameters**

- **package\_name** (`str`) – Package name to fetch the resource from.
- **file\_name** (`str`) – Filename of the resource.

**Returns** A stream object representing the resource file.

**Raises** `ResourceHelperError` if unable to locate the resource `file_name` in `package_name`.

**class** `repobuddy.utils.EqualityBase`

Bases: `object`

Provides equality comparison operations.

A base class which provides support for performing equality comparison on the instance. The type and the instance dictionary are used for comparison.

**exception** `repobuddy.utils.LoggerError` (`error_str`)

Bases: `exceptions.Exception`

Exception raised by `Logger`.

**\_\_init\_\_** (`error_str`)

Initializer.

**Parameters** `error_str` (`str`) – The error string to store in the exception.

**class** `repobuddy.utils.Logger`

Provides logging support for the rest of `repobuddy`.

Currently supported log levels are:

- DEBUG
- MESSAGE
- ERROR

**disable\_debug** = True

**debug\_stream** = <open file '<stdout>', mode 'w'>

**msg\_stream** = <open file '<stdout>', mode 'w'>

**error\_stream** = <open file '<stdout>', mode 'w'>

**classmethod `msg`** (*msg*, *append\_new\_line=True*)

Add a log entry of level MESSAGE.

**Parameters**

- **`msg`** (*str*) – The message to log.
- **`append_new_line`** (*Boolean*) – Appends a new line after the log message when set to True.

**Returns** None

**Raises** *LoggerError* on errors.

**classmethod `debug`** (*msg*, *append\_new\_line=True*)

Add a log entry of level DEBUG.

**Parameters**

- **`msg`** (*str*) – The message to log.
- **`append_new_line`** (*Boolean*) – Appends a new line after the log message when set to True.

**Returns** None

**Raises** *LoggerError* on errors.

**classmethod `error`** (*msg*, *append\_new\_line=True*)

Add a log entry of level ERROR.

**Parameters**

- **`msg`** (*str*) – The message to log.
- **`append_new_line`** (*Boolean*) – Appends a new line after the log message when set to True.

**Returns** None

**Raises** *LoggerError* on errors.

## repopuddy.tests – Root package for tests

### repopuddy.tests.arg\_parser – Argument Parser tests

```
class repobuddy.tests.arg_parser.ArgParserTestCase (methodName='runTest')
    Bases: repobuddy.tests.common.TestCaseBase
        classmethod setUpClass ()
        classmethod tearDownClass ()
        __init__ (methodName='runTest')
        setUp ()
        test_help ()
        test_version ()
        test_init_help ()
        test_status_help ()
        test_help_unsupported_command ()
        test_unsupported_command ()
        test_handlers ()
class repobuddy.tests.arg_parser.ArgParserTestSuite
    classmethod get_test_suite ()
```

## repobuddy.tests.client\_info – Client Info tests

```
class repobuddy.tests.client_info.ClientInfoTestCase (methodName='runTest')
    Bases: repobuddy.tests.common.TestCaseBase

    classmethod setUpClass ()
    classmethod tearDownClass ()
    __init__ (methodName='runTest')
    test_read_nonexistent_file ()
    test_read_without_permissions ()
    test_read_malformed_file ()
    test_read_empty_file ()
    test_read_without_section_header ()
    test_read_just_section ()
    test_read_just_options ()
    test_read_no_client_spec ()
    test_read_no_manifest ()
    test_read_valid ()
    test_read_valid_writeback ()
    test_create_new_config ()

class repobuddy.tests.client_info.ClientInfoTestSuite

    classmethod get_test_suite ()
```

## repobuddy.tests.command\_handler – Command Handler tests

```
class repobuddy.tests.command_handler.CommandHandlerTestCase (methodName='runTest')
    Bases: repobuddy.tests.common.TestCaseBase

    classmethod setUpClass ()
    classmethod tearDownClass ()
    __init__ (methodName='runTest')
    setUp ()
    test_verify_handlers ()
    test_init_client_valid ()

class repobuddy.tests.command_handler.CommandHandlerTestSuite

    classmethod get_test_suite ()
```

## repopuddy.tests.common – Test Framework

**exception** repobuddy.tests.common.**ShellError** (*error\_str*)

Bases: *repopuddy.utils.RepoBuddyBaseException*

Exception raised by *ShellHelper*.

**\_\_init\_\_** (*error\_str*)

Initializer.

**Parameters** *error\_str* (*str*) – The error string to store in the exception.

**class** repobuddy.tests.common.**ShellHelper**

Helper for performing shell operations.

**classmethod** **exec\_command** (*command*, *base\_dir*, *debug\_output=True*)

Execute a command.

**Parameters**

- **command** (*str*) – The command to execute.
- **base\_dir** (*str*) – The base directory path.
- **debug\_output** – If set to *False*, the output of the command is **NOT** printed on *stdout* and *stderr*.

**Returns** *None*

**Raises** *ShellError* if the program's execution returned a non-zero return or any other errors.

**classmethod** **read\_file\_as\_string** (*filename*)

Read the contents of the file as a string and return it.

**Parameters** *filename* (*str*) – The name of the file to read.

**Returns** Contents of the file.

**Return type** *str*

**Raises** *ShellError* on errors.

**classmethod** **append\_text\_to\_file** (*text*, *filename*, *base\_dir*)

**classmethod** **remove\_file** (*filename*)

**classmethod** **make\_dir** (*dirname*, *create\_parent\_dirs=False*, *only\_if\_not\_exists=False*)

**classmethod** **remove\_dir** (*dirname*)

**class** repobuddy.tests.common.**TestCommon**

**classmethod** **get\_string\_stream** ()

**classmethod** **setup\_test\_repos** (*base\_dir*)

**class** repobuddy.tests.common.**TestCaseBase** (*methodName='runTest'*)

Bases: *unittest.case.TestCase*

**\_\_init\_\_** (*methodName='runTest'*)

**setUp** ()

**tearDown** ()

```
class repobuddy.tests.common.TestResult
    Bases: unittest.result.TestResult

    PASSED = 0
    ERROR = 1
    FAILED = 2
    SKIPPED = 3
    EXPECTED_FAILURE = 4
    UNEXPECTED_SUCCESS = 5

    __init__()

    classmethod get_result_str(result)

    addError(test, err)

    addFailure(test, err)

    addSuccess(test)

    addSkip(test, reason)

    addExpectedFailure(test, err)

    addUnexpectedSuccess(test)

class repobuddy.tests.common.TestRunner(stream, descriptions=True, verbosity=1)
    Bases: unittest.runner.TextTestRunner

    __init__(stream, descriptions=True, verbosity=1)

    get_test_result()

class repobuddy.tests.common.TestSuiteManager(base_dir)
    Bases: object

    classmethod get_base_dir()

    __init__(base_dir)

    add_test_suite(test_suite)

    run()

    show_results()

    was_successful()
```

## repobuddy.tests.git\_wrapper – Git Wrapper tests

```
class repobuddy.tests.git_wrapper.GitWrapperTestCase(methodName='runTest')
    Bases: repobuddy.tests.common.TestCaseBase

    classmethod setUpClass()

    classmethod tearDownClass()

    __init__(methodName='runTest')

    test_clone_valid_repo()

    test_clone_invalid_url()
```

```
test_clone_invalid_branch()
test_clone_no_write_permissions()
test_update_index_valid_repo()
test_update_index_invalid_repo()
test_untracked_no_files()
test_untracked_with_files()
test_unstaged_no_files()
test_unstaged_with_files()
test_uncommitted_no_changes()
test_uncommitted_with_changes()
test_current_branch_valid_repo()
test_current_branch_invalid_repo()
test_current_branch_detached_head()
test_current_tag_lightweight_tag()
test_current_tag_annotated_tag()
test_current_tag_no_tag()

class repobuddy.tests.git_wrapper.GitWrapperTestSuite

    classmethod get_test_suite()
```

## repobuddy.tests.main – Main routine for the tests

```
repobuddy.tests.main.run_tests()
```

## repobuddy.tests.manifest\_parser – Manifest Parser tests

```
class repobuddy.tests.manifest_parser.ManifestParserTestCase (methodName='runTest')
    Bases: repobuddy.tests.common.TestCaseBase
    __init__ (methodName='runTest')
    test_repo_str_repr()
    test_client_spec_str_repr()
    test_manifest_str_repr()
    test_parse_invalid_file_handle()
    test_valid_manifest()
    test_malformed()
    test_no_client_spec()
    test_empty_client_spec()
```

```
test_client_spec_no_name()
test_empty_repo()
test_repo_no_url()
test_repo_empty_url()
test_repo_no_branch()
test_repo_empty_branch()
test_repo_no_dest()
test_repo_empty_dest()
test_no_default_client_spec()
test_empty_default_client_spec()
test_nonexistent_default_client_spec()
test_duplicate_client_spec()
class repobuddy.tests.manifest_parser.ManifestParserTestSuite

    classmethod get_test_suite()
```

## repobuddy.tests.utils – Utilities tests

```
class repobuddy.tests.utils.UtilsTestCase(methodName='runTest')
    Bases: repobuddy.tests.common.TestCaseBase
    classmethod setUpClass()
    classmethod tearDownClass()
    __init__(methodName='runTest')
    test_file_lock_basic()
    test_file_lock_multiple_times()
    test_file_lock_multiple_threads()
    test_file_lock_delete_with_acquire()
    test_file_lock_dir_without_permissions()
class repobuddy.tests.utils.UtilsTestSuite

    classmethod get_test_suite()
```



## CHAPTER 4

---

### Indices and tables

---

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



### r

- repobuddy, [5](#)
- repobuddy.arg\_parser, [5](#)
- repobuddy.client\_info, [6](#)
- repobuddy.command\_handler, [7](#)
- repobuddy.git\_wrapper, [8](#)
- repobuddy.globals, [10](#)
- repobuddy.main, [10](#)
- repobuddy.manifest\_parser, [10](#)
- repobuddy.tests, [15](#)
- repobuddy.tests.arg\_parser, [15](#)
- repobuddy.tests.client\_info, [16](#)
- repobuddy.tests.command\_handler, [16](#)
- repobuddy.tests.common, [17](#)
- repobuddy.tests.git\_wrapper, [18](#)
- repobuddy.tests.main, [19](#)
- repobuddy.tests.manifest\_parser, [19](#)
- repobuddy.tests.utils, [20](#)
- repobuddy.utils, [12](#)



## Symbols

[\\_\\_init\\_\\_\(\) \(repobuddy.arg\\_parser.ArgParser method\), 5](#)  
[\\_\\_init\\_\\_\(\) \(repobuddy.arg\\_parser.ArgParserError method\), 5](#)  
[\\_\\_init\\_\\_\(\) \(repobuddy.client\\_info.ClientInfo method\), 6](#)  
[\\_\\_init\\_\\_\(\) \(repobuddy.client\\_info.ClientInfoError method\), 6](#)  
[\\_\\_init\\_\\_\(\) \(repobuddy.command\\_handler.CommandHandler method\), 7](#)  
[\\_\\_init\\_\\_\(\) \(repobuddy.command\\_handler.CommandHandlerError method\), 7](#)  
[\\_\\_init\\_\\_\(\) \(repobuddy.git\\_wrapper.GitWrapper method\), 8](#)  
[\\_\\_init\\_\\_\(\) \(repobuddy.git\\_wrapper.GitWrapperError method\), 8](#)  
[\\_\\_init\\_\\_\(\) \(repobuddy.globals.HelpStringsError method\), 10](#)  
[\\_\\_init\\_\\_\(\) \(repobuddy.manifest\\_parser.ClientSpec method\), 11](#)  
[\\_\\_init\\_\\_\(\) \(repobuddy.manifest\\_parser.Manifest method\), 11](#)  
[\\_\\_init\\_\\_\(\) \(repobuddy.manifest\\_parser.ManifestParser method\), 11](#)  
[\\_\\_init\\_\\_\(\) \(repobuddy.manifest\\_parser.ManifestParserError method\), 10](#)  
[\\_\\_init\\_\\_\(\) \(repobuddy.manifest\\_parser.Repo method\), 10](#)  
[\\_\\_init\\_\\_\(\) \(repobuddy.tests.arg\\_parser.ArgParserTestCase method\), 15](#)  
[\\_\\_init\\_\\_\(\) \(repobuddy.tests.client\\_info.ClientInfoTestCase method\), 16](#)  
[\\_\\_init\\_\\_\(\) \(repobuddy.tests.command\\_handler.CommandHandlerTestCase method\), 16](#)  
[\\_\\_init\\_\\_\(\) \(repobuddy.tests.common.ShellError method\), 17](#)  
[\\_\\_init\\_\\_\(\) \(repobuddy.tests.common.TestCaseBase method\), 17](#)  
[\\_\\_init\\_\\_\(\) \(repobuddy.tests.common.TestResult method\), 18](#)  
[\\_\\_init\\_\\_\(\) \(repobuddy.tests.common.TestRunner method\), 18](#)  
[\\_\\_init\\_\\_\(\) \(repobuddy.tests.common.TestSuiteManager method\), 18](#)  
[\\_\\_init\\_\\_\(\) \(repobuddy.tests.git\\_wrapper.GitWrapperTestCase method\), 18](#)  
[\\_\\_init\\_\\_\(\) \(repobuddy.tests.manifest\\_parser.ManifestParserTestCase method\), 19](#)  
[\\_\\_init\\_\\_\(\) \(repobuddy.tests.utils.UtilsTestCase method\), 20](#)  
[\\_\\_init\\_\\_\(\) \(repobuddy.utils.FileLock method\), 12](#)  
[\\_\\_init\\_\\_\(\) \(repobuddy.utils.FileLockError method\), 12](#)  
[\\_\\_init\\_\\_\(\) \(repobuddy.utils.LoggerError method\), 13](#)  
[\\_\\_init\\_\\_\(\) \(repobuddy.utils.RepoBuddyBaseException method\), 12](#)  
[\\_\\_init\\_\\_\(\) \(repobuddy.utils.ResourceHelperError method\), 13](#)

## A

[acquire\(\) \(repobuddy.utils.FileLock method\), 12](#)  
[add\\_test\\_suite\(\) \(repobuddy.tests.common.TestSuiteManager method\), 18](#)  
[addError\(\) \(repobuddy.tests.common.TestResult method\), 18](#)  
[addExpectedFailure\(\) \(repobuddy.tests.common.TestResult method\), 18](#)  
[addFailure\(\) \(repobuddy.tests.common.TestResult method\), 18](#)  
[addSkip\(\) \(repobuddy.tests.common.TestResult method\), 18](#)  
[addSuccess\(\) \(repobuddy.tests.common.TestResult method\), 18](#)  
[addUnexpectedSuccess\(\) \(repobuddy.tests.common.TestResult method\), 18](#)  
[append\\_text\\_to\\_file\(\) \(repobuddy.tests.common.ShellHelper class method\), 17](#)  
[ArgParser \(class in repobuddy.arg\\_parser\), 5](#)  
[ArgParserError, 5](#)

ArgParserTestCase (class in repobuddy.tests.arg\_parser),  
15

ArgParserTestSuite (class in repobuddy.tests.arg\_parser),  
15

## C

ClientInfo (class in repobuddy.client\_info), 6

ClientInfoError, 6

ClientInfoTestCase (class in repobuddy.tests.client\_info),  
16

ClientInfoTestSuite (class in repobuddy.tests.client\_info),  
16

ClientSpec (class in repobuddy.manifest\_parser), 11

clone() (repobuddy.git\_wrapper.GitWrapper method), 8

CommandHandler (class in re-  
pobuddy.command\_handler), 7

CommandHandlerError, 7

CommandHandlerTestCase (class in re-  
pobuddy.tests.command\_handler), 16

CommandHandlerTestSuite (class in re-  
pobuddy.tests.command\_handler), 16

## D

debug() (repobuddy.utils.Logger class method), 14

debug\_stream (repobuddy.utils.Logger attribute), 13

disable\_debug (repobuddy.utils.Logger attribute), 13

## E

EqualityBase (class in repobuddy.utils), 13

ERROR (repobuddy.tests.common.TestResult attribute),  
18

error() (repobuddy.utils.Logger class method), 14

error\_stream (repobuddy.utils.Logger attribute), 13

exec\_command() (repobuddy.tests.common.ShellHelper  
class method), 17

EXPECTED\_FAILURE (re-  
pobuddy.tests.common.TestResult attribute),  
18

## F

FAILED (repobuddy.tests.common.TestResult attribute),  
18

FileLock (class in repobuddy.utils), 12

FileLockError, 12

## G

get\_base\_dir() (repobuddy.tests.common.TestSuiteManager  
class method), 18

get\_client\_spec() (repobuddy.client\_info.ClientInfo  
method), 7

get\_current\_branch() (re-  
pobuddy.git\_wrapper.GitWrapper method),  
9

get\_current\_tag() (repobuddy.git\_wrapper.GitWrapper  
method), 9

get\_handlers() (repobuddy.command\_handler.CommandHandler  
method), 7

get\_manifest() (repobuddy.client\_info.ClientInfo  
method), 7

get\_manifest() (repobuddy.manifest\_parser.ManifestParser  
method), 11

get\_result\_str() (repobuddy.tests.common.TestResult  
class method), 18

get\_string\_stream() (re-  
pobuddy.tests.common.TestCommon class  
method), 17

get\_test\_result() (repobuddy.tests.common.TestRunner  
method), 18

get\_test\_suite() (repobuddy.tests.arg\_parser.ArgParserTestSuite  
class method), 15

get\_test\_suite() (repobuddy.tests.client\_info.ClientInfoTestSuite  
class method), 16

get\_test\_suite() (repobuddy.tests.command\_handler.CommandHandlerTestSuite  
class method), 16

get\_test\_suite() (repobuddy.tests.git\_wrapper.GitWrapperTestSuite  
class method), 19

get\_test\_suite() (repobuddy.tests.manifest\_parser.ManifestParserTestSuite  
class method), 20

get\_test\_suite() (repobuddy.tests.utils.UtilsTestSuite class  
method), 20

get\_uncommitted\_staged\_files() (re-  
pobuddy.git\_wrapper.GitWrapper method),  
9

get\_unstaged\_files() (repobuddy.git\_wrapper.GitWrapper  
method), 9

get\_untracked\_files() (re-  
pobuddy.git\_wrapper.GitWrapper method),  
9

GitWrapper (class in repobuddy.git\_wrapper), 8

GitWrapperError, 8

GitWrapperTestCase (class in re-  
pobuddy.tests.git\_wrapper), 18

GitWrapperTestSuite (class in re-  
pobuddy.tests.git\_wrapper), 19

## H

HELP\_COMMAND\_ARG (re-  
pobuddy.globals.HelpStrings attribute), 10

HELP\_COMMAND\_HELP (re-  
pobuddy.globals.HelpStrings attribute), 10

HelpStrings (class in repobuddy.globals), 10

HelpStringsError, 10

## I

INIT\_CLIENT\_SPEC\_ARG (re-  
pobuddy.globals.HelpStrings attribute), 10

- init\_command\_handler() (repobuddy.command\_handler.CommandHandler method), 7
- INIT\_COMMAND\_HELP (repobuddy.globals.HelpStrings attribute), 10
- INIT\_MANIFEST\_ARG (repobuddy.globals.HelpStrings attribute), 10
- ## L
- Logger (class in repobuddy.utils), 13
- LoggerError, 13
- ## M
- make\_dir() (repobuddy.tests.common.ShellHelper class method), 17
- Manifest (class in repobuddy.manifest\_parser), 11
- ManifestParser (class in repobuddy.manifest\_parser), 11
- ManifestParserError, 10
- ManifestParserTestCase (class in repobuddy.tests.manifest\_parser), 19
- ManifestParserTestSuite (class in repobuddy.tests.manifest\_parser), 20
- MASTER\_PARSER\_ARG\_HELP (repobuddy.globals.HelpStrings attribute), 10
- MASTER\_PARSER\_ARG\_TITLE (repobuddy.globals.HelpStrings attribute), 10
- msg() (repobuddy.utils.Logger class method), 13
- msg\_stream (repobuddy.utils.Logger attribute), 13
- ## O
- open\_data\_file() (repobuddy.utils.ResourceHelper class method), 13
- ## P
- parse() (repobuddy.arg\_parser.ArgParser method), 6
- parse() (repobuddy.manifest\_parser.ManifestParser method), 11
- PASSED (repobuddy.tests.common.TestResult attribute), 18
- PROGRAM\_DESCRIPTION (repobuddy.globals.HelpStrings attribute), 10
- PROGRAM\_NAME (repobuddy.globals.HelpStrings attribute), 10
- PROGRAM\_VERSION (repobuddy.globals.HelpStrings attribute), 10
- ## R
- read\_file\_as\_string() (repobuddy.tests.common.ShellHelper class method), 17
- release() (repobuddy.utils.FileLock method), 12
- remove\_dir() (repobuddy.tests.common.ShellHelper class method), 17
- remove\_file() (repobuddy.tests.common.ShellHelper class method), 17
- Repo (class in repobuddy.manifest\_parser), 10
- repobuddy (module), 5
- repobuddy.arg\_parser (module), 5
- repobuddy.client\_info (module), 6
- repobuddy.command\_handler (module), 7
- repobuddy.git\_wrapper (module), 8
- repobuddy.globals (module), 10
- repobuddy.main (module), 10
- repobuddy.manifest\_parser (module), 10
- repobuddy.tests (module), 15
- repobuddy.tests.arg\_parser (module), 15
- repobuddy.tests.client\_info (module), 16
- repobuddy.tests.command\_handler (module), 16
- repobuddy.tests.common (module), 17
- repobuddy.tests.git\_wrapper (module), 18
- repobuddy.tests.main (module), 19
- repobuddy.tests.manifest\_parser (module), 19
- repobuddy.tests.utils (module), 20
- repobuddy.utils (module), 12
- RepoBuddyBaseException, 12
- ResourceHelper (class in repobuddy.utils), 13
- ResourceHelperError, 13
- run() (repobuddy.tests.common.TestSuiteManager method), 18
- run\_repobuddy() (in module repobuddy.main), 10
- run\_tests() (in module repobuddy.tests.main), 19
- ## S
- set\_client\_spec() (repobuddy.client\_info.ClientInfo method), 6
- set\_manifest() (repobuddy.client\_info.ClientInfo method), 6
- setUp() (repobuddy.tests.arg\_parser.ArgParserTestCase method), 15
- setUp() (repobuddy.tests.command\_handler.CommandHandlerTestCase method), 16
- setUp() (repobuddy.tests.common.TestCaseBase method), 17
- setup\_test\_repos() (repobuddy.tests.common.TestCommon class method), 17
- setUpClass() (repobuddy.tests.arg\_parser.ArgParserTestCase class method), 15
- setUpClass() (repobuddy.tests.client\_info.ClientInfoTestCase class method), 16
- setUpClass() (repobuddy.tests.command\_handler.CommandHandlerTestCase class method), 16
- setUpClass() (repobuddy.tests.git\_wrapper.GitWrapperTestCase class method), 18
- setUpClass() (repobuddy.tests.utils.UtilsTestCase class method), 20
- ShellError, 17
- ShellHelper (class in repobuddy.tests.common), 17

show\_results() (repopbuddy.tests.common.TestSuiteManager test\_current\_tag\_annotated\_tag() (re-  
method), 18 popbuddy.tests.git\_wrapper.GitWrapperTestCase  
method), 19  
SKIPPED (repopbuddy.tests.common.TestResult at-  
tribute), 18 test\_current\_tag\_lightweight\_tag() (re-  
popbuddy.tests.git\_wrapper.GitWrapperTestCase  
method), 19  
STATUS\_COMMAND (repopbuddy.globals.HelpStrings  
attribute), 10 test\_current\_tag\_no\_tag() (re-  
popbuddy.command\_handler.CommandHandler  
method), 8 test\_duplicate\_client\_spec() (re-  
popbuddy.tests.manifest\_parser.ManifestParserTestCase  
method), 20  
test\_empty\_client\_spec() (re-  
popbuddy.tests.manifest\_parser.ManifestParserTestCase  
method), 19  
test\_empty\_default\_client\_spec() (re-  
popbuddy.tests.manifest\_parser.ManifestParserTestCase  
method), 20  
test\_empty\_repo() (repopbuddy.tests.manifest\_parser.ManifestParserTestCase  
method), 20  
test\_file\_lock\_basic() (re-  
popbuddy.tests.utils.UtilsTestCase method),  
20  
test\_file\_lock\_delete\_with\_acquire() (re-  
popbuddy.tests.utils.UtilsTestCase method),  
20  
test\_file\_lock\_dir\_without\_permissions() (re-  
popbuddy.tests.utils.UtilsTestCase method),  
20  
test\_file\_lock\_multiple\_threads() (re-  
popbuddy.tests.utils.UtilsTestCase method),  
20  
test\_file\_lock\_multiple\_times() (re-  
popbuddy.tests.utils.UtilsTestCase method),  
20  
test\_handlers() (repopbuddy.tests.arg\_parser.ArgParserTestCase  
method), 15  
test\_help() (repopbuddy.tests.arg\_parser.ArgParserTestCase  
method), 15  
test\_help\_unsupported\_command() (re-  
popbuddy.tests.arg\_parser.ArgParserTestCase  
method), 15  
test\_init\_client\_valid() (re-  
popbuddy.tests.command\_handler.CommandHandlerTestCase  
method), 16  
test\_init\_help() (repopbuddy.tests.arg\_parser.ArgParserTestCase  
method), 15  
test\_malformed() (repopbuddy.tests.manifest\_parser.ManifestParserTestCase  
method), 19  
test\_manifest\_str\_repr() (re-  
popbuddy.tests.manifest\_parser.ManifestParserTestCase  
method), 19  
test\_no\_client\_spec() (re-  
popbuddy.tests.manifest\_parser.ManifestParserTestCase

## T

tearDown() (repopbuddy.tests.common.TestCaseBase  
method), 17  
tearDownClass() (repopbuddy.tests.arg\_parser.ArgParserTestCase  
class method), 15  
tearDownClass() (repopbuddy.tests.client\_info.ClientInfoTestCase  
class method), 16  
tearDownClass() (repopbuddy.tests.command\_handler.CommandHandlerTestCase  
class method), 16  
tearDownClass() (repopbuddy.tests.git\_wrapper.GitWrapperTestCase  
class method), 18  
tearDownClass() (repopbuddy.tests.utils.UtilsTestCase  
class method), 20  
test\_client\_spec\_no\_name() (re-  
popbuddy.tests.manifest\_parser.ManifestParserTestCase  
method), 19  
test\_client\_spec\_str\_repr() (re-  
popbuddy.tests.manifest\_parser.ManifestParserTestCase  
method), 19  
test\_clone\_invalid\_branch() (re-  
popbuddy.tests.git\_wrapper.GitWrapperTestCase  
method), 18  
test\_clone\_invalid\_url() (re-  
popbuddy.tests.git\_wrapper.GitWrapperTestCase  
method), 18  
test\_clone\_no\_write\_permissions() (re-  
popbuddy.tests.git\_wrapper.GitWrapperTestCase  
method), 19  
test\_clone\_valid\_repo() (re-  
popbuddy.tests.git\_wrapper.GitWrapperTestCase  
method), 18  
test\_create\_new\_config() (re-  
popbuddy.tests.client\_info.ClientInfoTestCase  
method), 16  
test\_current\_branch\_detached\_head() (re-  
popbuddy.tests.git\_wrapper.GitWrapperTestCase  
method), 19  
test\_current\_branch\_invalid\_repo() (re-  
popbuddy.tests.git\_wrapper.GitWrapperTestCase  
method), 19  
test\_current\_branch\_valid\_repo() (re-  
popbuddy.tests.git\_wrapper.GitWrapperTestCase  
method), 19



method), 19	test_repo_no_dest() (re-	
test_no_default_client_spec() (re-	pobuddy.tests.manifest_parser.ManifestParserTestCase	
method), 20	method), 20	
test_nonexistent_default_client_spec() (re-	test_repo_no_url() (repobuddy.tests.manifest_parser.ManifestParserTestCase	
method), 20	method), 20	
test_parse_invalid_file_handle() (re-	test_repo_str_repr() (re-	
method), 20	pobuddy.tests.manifest_parser.ManifestParserTestCase	
test_parse_invalid_file_handle() (re-	method), 19	
method), 19	test_status_help() (repobuddy.tests.arg_parser.ArgParserTestCase	
test_read_empty_file() (re-	method), 15	
pobuddy.tests.client_info.ClientInfoTestCase	test_uncommitted_no_changes() (re-	
method), 16	pobuddy.tests.git_wrapper.GitWrapperTestCase	
test_read_just_options() (re-	method), 19	
pobuddy.tests.client_info.ClientInfoTestCase	test_uncommitted_with_changes() (re-	
method), 16	pobuddy.tests.git_wrapper.GitWrapperTestCase	
test_read_just_section() (re-	method), 19	
pobuddy.tests.client_info.ClientInfoTestCase	test_unstaged_no_files() (re-	
method), 16	pobuddy.tests.git_wrapper.GitWrapperTestCase	
test_read_malformed_file() (re-	method), 19	
pobuddy.tests.client_info.ClientInfoTestCase	test_unstaged_with_files() (re-	
method), 16	pobuddy.tests.git_wrapper.GitWrapperTestCase	
test_read_no_client_spec() (re-	method), 19	
pobuddy.tests.client_info.ClientInfoTestCase	test_unsupported_command() (re-	
method), 16	pobuddy.tests.arg_parser.ArgParserTestCase	
test_read_no_manifest() (re-	method), 15	
pobuddy.tests.client_info.ClientInfoTestCase	test_untracked_no_files() (re-	
method), 16	pobuddy.tests.git_wrapper.GitWrapperTestCase	
test_read_nonexistent_file() (re-	method), 19	
pobuddy.tests.client_info.ClientInfoTestCase	test_untracked_with_files() (re-	
method), 16	pobuddy.tests.git_wrapper.GitWrapperTestCase	
test_read_valid() (repobuddy.tests.client_info.ClientInfoTestCase	method), 19	
method), 16	test_update_index_invalid_repo() (re-	
test_read_valid_writeback() (re-	pobuddy.tests.git_wrapper.GitWrapperTestCase	
pobuddy.tests.client_info.ClientInfoTestCase	method), 19	
method), 16	test_update_index_valid_repo() (re-	
test_read_without_permissions() (re-	pobuddy.tests.git_wrapper.GitWrapperTestCase	
pobuddy.tests.client_info.ClientInfoTestCase	method), 19	
method), 16	test_valid_manifest() (re-	
test_read_without_section_header() (re-	pobuddy.tests.manifest_parser.ManifestParserTestCase	
pobuddy.tests.client_info.ClientInfoTestCase	method), 19	
method), 16	test_verify_handlers() (re-	
test_repo_empty_branch() (re-	pobuddy.tests.command_handler.CommandHandlerTestCase	
pobuddy.tests.manifest_parser.ManifestParserTestCase	method), 16	
method), 20	test_version() (repobuddy.tests.arg_parser.ArgParserTestCase	
test_repo_empty_dest() (re-	method), 15	
pobuddy.tests.manifest_parser.ManifestParserTestCase	TestCaseBase (class in repobuddy.tests.common), 17	
method), 20	TestCommon (class in repobuddy.tests.common), 17	
test_repo_empty_url() (re-	TestResult (class in repobuddy.tests.common), 17	
pobuddy.tests.manifest_parser.ManifestParserTestCase	TestRunner (class in repobuddy.tests.common), 18	
method), 20	TestSuiteManager (class in repobuddy.tests.common), 18	
test_repo_no_branch() (re-	U	
pobuddy.tests.manifest_parser.ManifestParserTestCase	UNEXPECTED_SUCCESS (re-	
method), 20	pobuddy.tests.common.TestResult attribute),	

[18](#)

`update_index()` (`repobuddy.git_wrapper.GitWrapper`  
method), [8](#)

`UtilsTestCase` (class in `repobuddy.tests.utils`), [20](#)

`UtilsTestSuite` (class in `repobuddy.tests.utils`), [20](#)

## W

`was_successful()` (`repobuddy.tests.common.TestSuiteManager`  
method), [18](#)

`write()` (`repobuddy.client_info.ClientInfo` method), [7](#)