RepoBuddy Documentation

Release 0.2alpha

Ash <tuxdude.github@gmail.com>

May 01, 2013

CONTENTS

Contents:

CHAPTER

ONE

INTRODUCTION TO REPOBUDDY

Multi-repository management tool for GIT

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

CHAPTER

TWO

CODE DOCUMENTATION

2.1 repobuddy – Root package

repobuddy module.

2.2 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 precedece, and the file name specififed 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.

2.3 `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*)

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.

2.4 `repobuddy.command_handler` - Command Handler

exception `repobuddy.command_handler.`**`CommandHandlerError`**(*error_str*)

Bases: `repobuddy.utils.RepoBuddyBaseException`

Exception raised by `CommandHandler`.

`__init__`(*error_str*)

class `repobuddy.command_handler.`**`CommandHandler`**

Bases: `object`

Provides handlers for the `repobuddy` 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.

2.5 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='*')

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.

2.6 repobuddy.globals - Global Definitions

exception repobuddy.globals.**HelpStringsError**(*error_str*)
Bases: `repobuddy.utils.RepoBuddyBaseException`

Exception raised by `HelpStrings`.

__init__(*error_str*)

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 0.2alpha'

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'

2.7 repobuddy.main - Program's Main Routine

repobuddy.main.**run_repobuddy**()

Invoke repobuddy with the command line arguments.

Exits with status 1 on errors, 0 otherwise.

returns: None

2.8 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`

`__init__`()

`parse`(*file_handle*)

`get_manifest`()

2.9 `repobuddy.utils` - Utility classes and functions

exception `repobuddy.utils.`**`RepoBuddyBaseException`**(*error_str*)
Bases: `exceptions.Exception`

`__init__`(*error_str*)

exception `repobuddy.utils.`**`FileLockError`**(*error_str*, *is_time_out=False*)
Bases: `repobuddy.utils.RepoBuddyBaseException`

`__init__`(*error_str*, *is_time_out=False*)

class `repobuddy.utils.`**`FileLock`**(*file_name*, *timeout=1*, *delay=0.1*)
Bases: `object`

`__init__`(*file_name*, *timeout=1*, *delay=0.1*)

`acquire`()

`release`()

exception `repobuddy.utils.`**`ResourceHelperError`**(*error_str*)
Bases: `repobuddy.utils.RepoBuddyBaseException`

`__init__`(*error_str*)

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

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

class `repobuddy.utils.`**`EqualityBase`**
Bases: `object`

exception `repobuddy.utils.`**`LoggerError`**(*error_str*)
Bases: `exceptions.Exception`

`__init__`(*error_str*)

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

disable_debug = **True**

debug_stream = **<open file '<stdout>', mode 'w' at 0x7fcafbd1e1e0>**

msg_stream = **<open file '<stdout>', mode 'w' at 0x7fcafbd1e1e0>**

error_stream = **<open file '<stdout>', mode 'w' at 0x7fcafbd1e1e0>**

classmethod msg(*msg*, *append_new_line=True*)

classmethod debug(*msg*, *append_new_line=True*)

classmethod error(*msg*, *append_new_line=True*)

CHAPTER

THREE

INDICES AND TABLES

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

PYTHON MODULE INDEX

r

repobuddy, **??**
repobuddy.arg_parser, **??**
repobuddy.client_info, **??**
repobuddy.command_handler, **??**
repobuddy.git_wrapper, **??**
repobuddy.globals, **??**
repobuddy.main, **??**
repobuddy.manifest_parser, **??**
repobuddy.utils, **??**