makeprojects Documentation

Release 0.13.1

Rebecca Ann Heineman <becky@burgerbecky.com>

Nov 21, 2022

CONTENTS

Makeprojects for Python

The makeprojects module makes it easy to autogenerate project files for several popular Integrated Development Enviroments (IDEs)

- Documentation is found at https://makeprojects.readthedocs.io
- Doxygen generated documentation is found at https://makeprojects.readthedocs.io/en/latest/doxygen
- Python Packing Index (PyPI): https://pypi.python.org/pypi/makeprojects
- Source code and issue tracker: https://github.com/burgerbecky/makeprojects

CHAPTER

ONE

COMPATIBILITY

- Python 2.7.1 or higher
- Python 3.4 or higher

CHAPTER

TWO

INSTALLATION

Type in `pip install -U makeprojects`. Some platforms may require the `sudo` prefix.

CHAPTER

THREE

BUGS

If you find a bug, issue or have a feature request, please submit a bug report by emailing becky@burgerbecky.com and mention python version, integer size (32 bit or 64 bit) and what platform was used (Windows / Mac OSX / Linux).

CHAPTER FOUR

TABLE OF CONTENTS

4.1 Buildme

`buildme` is a build system launcher that by using a single command, multiple build systems could be launched using a singluar command line. The tool has the ability to build all projects in a single directory, or even recursively through subdirectories with the `-r` parameter.

This is the launcher for the `makeprojects` project creation system and the `build_rules.py` file.

The tool has the ability to parse all project files to determine which version of Visual Studio, XCode, etc. to invoke so there is no needed to determine what build system to use, since it has the ability to determine the proper IDE/build tool to invoke.

`buildme` also handles most video game console build systems automatically. If any of these are missing, the default behavior will be used.

4.1.1 build_rules.py

A `build_rules.py` file can control the behavior of buildme by listing dependencies and custom build rules.

Full documentation on the operation of *build_rules.py is here*.

BUILDME_DEPENDENCIES

Set BUILDME_DEPENDENCIES with either a single string of a folder or file that must be built first. Also, a list of files or folders can be set and they will be built in the order of their position in the list. The default is no dependencies.

BUILDME_NO_RECURSE

Set BUILDME_NO_RECURSE to True if all subdirectories below this folder are not to be processed due to them not having any build project files. This defaults to `False`, but set this to `True` to prevent parsing folders that don't need processing.

prebuild(working_directory, configuration)

If this function exists, it will be called **FIRST** with the directory that the build_rules.py file exists in and the configuration requested to build. Normally the configuration is set to “all”, but can be ignored if it isn’t relevant to the custom build rules.

build(working_directory, configuration)

If this function exists, it will be called with the directory that the build_rules.py file exists in and the configuration requested to build. Normally the configuration is set to “all”, but can be ignored if it isn’t relevant to the custom build rules.

postbuild(working_directory, configuration)

If this function exists, it will be called **LAST** with the directory that the build_rules.py file exists in and the configuration requested to build. Normally the configuration is set to “all”, but can be ignored if it isn’t relevant to the custom build rules.

4.1.2 Usage

Navigate to a directory of interest or use a full directory path and run `buildme`. By default, it check the folder for a `build_rules.py` file for override rules, and it will either use the rules found to build the projects in the directory, and then it will scan the directory for known project files and invoke the appropriate build system to build the project. If a configuration is passed a parameter, only that configuration will be built in all projects being compiled.

“`buildme` Release”, build only the “Release” configuration of all project files.

“`buildme`”, build all configurations of the project files.

“`buildme` helloworld.sln”, build all configurations in helloworld.sln.

4.1.3 Visual Studio

If the project file ends with .sln, it’s assumed to be a Visual Studio project file.

If the host is Windows, MSYS2, Cygwin, or Windows Subsystem for Linux, it will build Visual Studio projects if the appropriate version of Visual Studio was installed.

These are the supported versions of Visual Studio:

- Visual Studio 2003 .NET
- Visual Studio 2005
- Visual Studio 2008
- Visual Studio 2010
- Visual Studio 2012
- Visual Studio 2013
- Visual Studio 2015
- Visual Studio 2017
- Visual Studio 2019

- Visual Studio 2022

These are the supported Visual Studio plug ins.

- Win32 (Windows 32 bit x86)
- x64 (Windows 64 bit x64)
- ARM (Windows 32 bit ARM)
- ARM64 (Windows 64 bit ARM)
- Xbox (Xbox classic)
- Xbox 360
- Durango (Xbox ONE)
- PS3
- PS4
- PSVita
- NX32 (Switch 32 bit)
- NX64 (Switch 64 bit)
- Tegra-Android (nVidia Shield)

4.1.4 XCode

If the folder ends with .xcodeproj and contains the file project.pbxproj, it's assumed to be Apple XCode. This type of project can only be built on macOS hosts.

4.1.5 Watcom

If the file ends with .wmk, it's assumed to be a Watcom WMAKE file. It can be built on Linux and Windows hosts.

4.1.6 Codeblocks

If the file ends with .cdp, it's assumed to be a CodeBlocks project file. It's invoked on Windows and Linux hosts.

4.1.7 CodeWarrior

If the file ends with .mcp, it's assumed to be a Metrowerks / NXP CodeWarrior file. It's invoked on a Windows or macOS hosts.

4.1.8 Linux Make

If the file is makefile, it's assumed to be a Linux make file and is invoked only on Linux hosts.

4.1.9 Ninja

If the file ends with .ninja, it's assumed to be a ninja file and is invoked on all hosts.

4.1.10 Credits

`buildme` is the insane creation of Rebecca Ann Heineman.

If bugs are found, please send all information on how to recreate the bug to becky@burgerbecky.com

4.2 Cleanme

`cleanme` is a tool to quickly and easily remove all temporary and generated files to force a full rebuild of a project on the next time a build is invoked. The tool has the ability to clean all projects in a single directory, or recursively through subdirectories with the `-r` parameter.

The tool has the ability to parse all project files to determine which version of Visual Studio, XCode, etc. to invoke so there is no needed to determine what build system to use, since it has the ability to determine the proper IDE/build tool to invoke. Custom clean rules are defined in the file `build_rules.py` <*see below*>

4.2.1 Usage

Navigate to a directory of interest or use a full directory path and run `cleanme`. By default, it checks the folder for a `build_rules.py` file for override rules, and it will either use the rules found to clean the projects in the directory, and then it will scan the directory for known project files and invoke the appropriate clean system to remove the files. If a configuration is passed a parameter, only that configuration will be cleaned.

"`cleanme` Release", clean only the "Release" configuration of all project files.

"`cleanme`", clean all configurations of the project files.

"`cleanme` helloworld.sln", clean all configurations in helloworld.sln.

4.2.2 Directory traversal

When the command line option `-r` is used, `cleanme` will traverse all folders recursively and process all folders found. Due to the nature of cleaning, for performance reasons, the directories will be processed under the current directory first, and then it will process all child directories secondly. This is the reverse order of `buildme` because in most cases, directories will be deleted when being cleaned, which will not exist when the directory is scanned for subdirectories to prevent processing directories that were removed.

Example:

If `build_rules.py` deletes `temp`, then if `cleanme` is executed at the root with `-r`, `build_rules.py` will be executed at the root which will remove the folder `temp`, and then it will process the remaining folders `source` and `data`, skipping over temp because it doesn't exist.

In this example, the folders `.`, `data`, and `source` will be processed but `temp` will not because it and its contents are removed.

cleanme_dir.png

```
.
+-- .gitignore
+-- build_rules.py
+-- data
|   +--- build_rules.py
|   +--- foo.png
+-- temp
|   +--- foo.obj
+-- source
|   +--- foo.cpp
```

4.2.3 build_rules.py

A build_rules file contains both static variables and a function to process a folder for cleaning. The static variables are checked first to guide the behavior of the `cleanme` tool, and if present, the function `clean(working_directory)` is called for custom clean rules. The function can return an error code which is returned to the command shell that invoked `cleanme`. Returning `None` acts if no error occured.

Full documentation on the operation of *build_rules.py is here*.

4.2.4 Visual Studio

If the project file ends with .sln, it's assumed to be a Visual Studio project file.

If the host is Windows, MSYS2, Cygwin, or Windows Subsystem for Linux, it will clean Visual Studio projects if the appropriate version of Visual Studio was installed.

These are the supported versions of Visual Studio:

- Visual Studio 2003 .NET
- Visual Studio 2005
- Visual Studio 2008
- Visual Studio 2010
- Visual Studio 2012
- Visual Studio 2013
- Visual Studio 2015
- Visual Studio 2017
- Visual Studio 2019
- Visual Studio 2022

4.2.5 XCode

If the folder ends with .xcodeproj and contains the file project.pbxproj, it's assumed to be Apple XCode. This type of project can only be built on macOS hosts.

4.2.6 Watcom

If the file ends with .wmk, it's assumed to be a Watcom WMAKE file. It can be built on Linux and Windows hosts. It will invoke the "clean" target.

4.2.7 Codeblocks

If the file ends with .cdp, it's assumed to be a CodeBlocks project file. It's invoked on Windows and Linux hosts.

4.2.8 CodeWarrior

If the file ends with .mcp, it's assumed to be a Metrowerks / NXP CodeWarrior file. It's invoked on a Windows or macOS hosts.

4.2.9 Linux Make

If the file is makefile, it's assumed to be a Linux make file and is invoked only on Linux hosts. It will invoke the "clean" target.

4.2.10 Ninja

If the file ends with .ninja, it's assumed to be a ninja file and is invoked on all hosts. It will invoke the "clean" target.

4.2.11 Credits

`cleanme` is the insane creation of Rebecca Ann Heineman.

If bugs are found, please send all information on how to recreate the bug to becky@burgerbecky.com

4.3 Rebuildme

`rebuildme` is a build system launcher that by using a single command, multiple build systems could be launched using a singluar command line. The tool has the ability to build all projects in a single directory, or even recursively through subdirectories with the `-r` parameter.

This is the launcher for the `makeprojects` project creation system and the `build_rules.py` file.

Full documentation on the operation of *build_rules.py is here*.

The tool has the ability to parse all project files to determine which version of Visual Studio, XCode, etc. to invoke so there is no needed to determine what build system to use, since it has the ability to determine the proper IDE/build tool to invoke.

`rebuildme` also handles most video game console build systems automatically.

4.3.1 Usage

Navigate to a directory of interest or use a full directory path and run `rebuildme`. By default, it check the folder for a `build_rules.py` file for override rules, and it will either use the rules found to build the projects in the directory, it will scan the directory for known project files and invoke the appropriate build system to build the project. If a configuration is passed a parameter, only that configuration will be built in all projects being compiled.

“`rebuildme` Release”, build only the “Release” configuration of all project files.

“`rebuildme`”, build all configurations of the project files.

“`rebuildme` helloworld.sln”, build all configurations in helloworld.sln.

4.3.2 Visual Studio

If the project file ends with .sln, it’s assumed to be a Visual Studio project file.

If the host is Windows, MSYS2, Cygwin, or Windows Subsystem for Linux, it will build Visual Studio projects if the appropriate version of Visual Studio was installed.

These are the supported versions of Visual Studio:

- Visual Studio 2003 .NET
- Visual Studio 2005
- Visual Studio 2008
- Visual Studio 2010
- Visual Studio 2012
- Visual Studio 2013
- Visual Studio 2015
- Visual Studio 2017
- Visual Studio 2019
- Visual Studio 2022

4.3.3 XCode

If the folder ends with .xcodeproj and contains the file project.pbxproj, it’s assumed to be Apple XCode. This type of project can only be built on macOS hosts.

4.3.4 Watcom

If the file ends with .wmk, it’s assumed to be a Watcom WMAKE file. It can be built on Linux and Windows hosts.

4.3.5 Codeblocks

If the file ends with .cdp, it's assumed to be a CodeBlocks project file. It's invoked on Windows and Linux hosts.

4.3.6 CodeWarrior

If the file ends with .mcp, it's assumed to be a Metrowerks / NXP CodeWarrior file. It's invoked on a Windows or macOS hosts.

4.3.7 Linux Make

If the file is makefile, it's assumed to be a Linux make file and is invoked only on Linux hosts.

4.3.8 Ninja

If the file ends with .ninja, it's assumed to be a ninja file and is invoked on all hosts.

4.3.9 Credits

`rebuildme` is the insane creation of Rebecca Ann Heineman.

If bugs are found, please send all information on how to recreate the bug to becky@burgerbecky.com

4.4 Build Rules

Makeprojects is controlled by a configuration file called `build_rules.py`. It's a python script with global variables and functions that will control how `buildme`, `cleanme`, `rebuildme`, and `makeprojects` behave. Since it's a python script, there is no limit to what the script can do to perform any build or clean operation for any project.

When a directory is checked for processing, a `build_rules.py` file is checked. If it doesn't exist, the parent directory is checked until the root directory is found which stops the scanning. If the file is not found, processing will stop. If found, it will be checked if it is in the folder being processed and is used if so. If the file is in a parent folder, a `GENERIC_*` variable is checked to see if the `build_rules.py` qualifies as a "catch all" file that handles rules for all child folders.

Below, are the functions and variables that the `build_rule.py` may or may not contain to control the tool's behavior. If the value is it not found, the defaults are shown below.

Note: `rebuildme` performs a `cleanme` and then a `buildme` operation, so it processes both `CLEANME_*` and `BUILDME_*` parameters.

4.4.1 Cleanme rules

cleanme and rebuildme check for several global variables and the existence of a single function that will perform cleaning operations.

Full documentation on the operation of *cleanme is here*.

CLEANME_GENERIC

```
# ``cleanme`` will process any child directory with the clean() function if
# True.
CLEANME_GENERIC = False
```

If set to True, cleanme and rebuildme will assume this build_rules.py file is designed to be generic and if invoked from a child directory, it will be given the child's directory for processing. If False it will only be invoked with the directory that the build_rules files resides in. This is needed to prevent a build_rules.py file from processing directories that it was not meant to handle when parent directory traversal is active. If this does not exist, the default of False is used.

CLEANME_CONTINUE

```
# ``cleanme`` will process build_rules.py in the parent folder if True.
CLEANME_CONTINUE = False
```

If set to True, cleanme and rebuildme will process this file and then traverse the parent directory looking for another build_rules.py file to continue the cleanme operation. This is useful when there's a generic clean operation in a root folder and this function performs custom operations unknown to the parent rules file. If this doesn't exist, the default of False is assumed.

CLEANME_DEPENDENCIES

```
# ``cleanme`` will clean the listed folders using their rules before cleaning.
# this folder.
CLEANME_DEPENDENCIES = []
```

cleanme and rebuildme will clean the listed folders using their rules before cleaning this folder. Only folders are allowed, files generate an error. If this doesn't exist, the default of an empty list is assumed.

CLEANME_NO_RECURSE

```
# If set to True, ``cleanme -r``` will not parse directories in this folder.
CLEANME_NO_RECURSE = False
```

If set to True, cleanme -r and rebuildme -r will not parse directories in this folder. If this does not exist, the default of False is used. The main purpose of this is to prevent scanning child folders when it is already known that there are no child folders that need processing or that CLEANME_DEPENDENCIES lists every child folder of interest, so recursion is not necessary.

CLEANME_PROCESS_PROJECT_FILES

```
# ``cleanme`` will assume only the function ``clean()`` is used if True.
CLEANME_PROCESS_PROJECT_FILES = False
```

If set to `False`, `cleanme` will disable scanning for project files and assume that the function `clean()` in build_rules.py performs all actions to clean the directory. If this doesn't exist, the default of `False` is assumed. Set this to `True` if the `clean()` function performs all of the operations needed to remove temporary files without the need to invoke any IDE.

clean(working_directory)

```
def clean(working_directory):
    return None
```

This function should delete all temporary files that were created after a project is built. In most cases, IDEs will be able to handle this, but for some projects, there are other files such as headers or compiled shaders that the IDE is not aware of. This function will perform the deletion functions and return either 0 for no error, or non-zero for an error that will be reported once `cleanme` is finished processing.

If the variable `CLEANME_GENERIC` is `False`, the working_directory is guaranteed to only be the directory that the `build_rules.py` resides in. If `CLEANME_GENERIC` is `False` then the directory could be any of the child folders the `build_rules.py` file resides in.

Returning `None` alerts `cleanme` and `rebuildme` that this function is not implemented and no action was performed.

4.4.2 Buildme rules

`buildme` and `rebuildme` checks for several global variables and the existence of three functions that will perform building operations.

Full documentation on the operation of *buildme is here*.

BUILDME_GENERIC

```
# Process any child directory with the prebuild(), build(), and postbuild()
# functions if True.
BUILDME_GENERIC = False
```

If set to `True`, `buildme` will assume this `build_rules.py` file is designed to be generic and if invoked from a child directory, it will be given the child's directory for processing. If `False` it will only be invoked with the directory that the `build_rules.py` files resides in. This is needed to prevent a `build_rules.py` file from processing directories that it was not meant to handle when parent directory traversal is active. If this does not exist, the default of `False` is used.

BUILDME_CONTINUE

```
# ``buildme`` will process build_rules.py in the parent folder if True.
BUILDME_CONTINUE = False
```

If set to True, buildme and rebuildme will process this file and then traverse the parent directory looking for another build_rules.py file to continue the buildme operation. This is useful when there's a generic build operation in a root folder and this function performs custom operations unknown to the parent rules file. If this doesn't exist, the default of False is assumed.

BUILDME_DEPENDENCIES

```
# Build the folders listed before processing this folder.
BUILDME_DEPENDENCIES = []
```

Set BUILDME_DEPENDENCIES with a list of folders or project files that must be built first. They will be built in the order they appear in this list. The default is no dependencies.

BUILDME_NO_RECURSE

```
# Disable recursion in all directories found in this directory.
BUILDME_NO_RECURSE = True
```

Set BUILDME_NO_RECURSE to True if all subdirectories below this folder are not to be processed due to them not having any build project files. This defaults to False, but set this to True to prevent parsing folders that don't need processing.

BUILDME_PROCESS_PROJECT_FILES

```
# ``buildme`` will assume only the three functions are used if True.
BUILDME_PROCESS_PROJECT_FILES = True
```

If set to False, buildme will disable scanning for project files and assume that the functions prebuild(), build(), and postbuild() in build_rules.py perform all the actions to build the files in this directory. If this doesn't exist, the default of True is assumed.

prebuild(working_directory, configuration)

```
def prebuild(working_directory, configuration)
    return None
```

If this optional function exists, it will be called **FIRST** with the directory requested to build and the configuration requested to build. Normally the configuration is set to "all", but can be ignored if it isn't relevant to the custom build rules. If BUILDME_GENERIC is False, only the directory that the build_rules.py file resides in will be passed as the working_directory.

This function will perform build functions and return either 0 for no error, or non-zero for an error that will be reported once buildme is finished processing. Return None if this function does no operation.

build(working_directory, configuration)

```
def build(working_directory, configuration)
    return None
```

If this optional function exists, it will be called after `prebuild()` is called but before any other the IDE project files are processed. It will passed the directory requested to build and the configuration requested to build. Normally the configuration is set to "all", but can be ignored if it isn't relevant to the custom build rules. If `BUILDME_GENERIC` is `False`, only the directory that the `build_rules.py` file resides in will be passed as the `working_directory`.

This function will perform build functions and return either 0 for no error, or non-zero for an error that will be reported once `buildme` is finished processing. Return `None` if this function does no operation.

postbuild(working_directory, configuration)

```
def postbuild(working_directory, configuration)
    return None
```

If this optional function exists, it will be called **LAST** with the directory requested to build and the configuration requested to build. Normally the configuration is set to "all", but can be ignored if it isn't relevant to the custom build rules. If `BUILDME_GENERIC` is `False`, only the directory that the `build_rules.py` file resides in will be passed as the `working_directory`.

This function will perform build functions and return either 0 for no error, or non-zero for an error that will be reported once `buildme` is finished processing. Return `None` if this function does no operation.

4.4.3 Makeprojects rules

DEFAULT_PROJECT_NAME

```
# Default project name to use instead of the name of the working directory
# DEFAULT_PROJECT_NAME = os.path.basename(working_directory)
```

When `makeprojects` is invoked, if a project name is not specified, this variable will declare the default project name. The default is the name of the folder that is being processed.

4.4.4 Global rules

These rules act like the ones that are specific to each tool, except they affect all of the tools.

GENERIC

```
# Process any child directory if True.
GENERIC = False
```

If set to `True`, all tools will assume this `build_rules.py` file is designed to be generic and if invoked from a child directory, it will be given the child's directory for processing. If `False` it will only be invoked with the directory that the build_rules files resides in. This is needed to prevent a `build_rules.py` file from processing directories that it was not meant to handle when parent directory traversal is active. If this does not exist, the default of `False` is used.

CONTINUE

```
# Process ``build_rules.py`` in the parent folder if True.
CONTINUE = False
```

If set to `True`, all tools will process this file and then traverse the parent directory looking for another `build_rules.py` file to continue the their operations. This is useful when there's a generic `build_rules.py` file in a root folder and this function performs custom operations unknown to the parent rules file. If this doesn't exist, the default of `False` is assumed.

DEPENDENCIES

```
# Process listed folders using their rules before processing this folder.
DEPENDENCIES = []
```

Process the listed folders using their rules before this folder. Only folders are allowed, files generate an error. If this doesn't exist, the default of an empty list is assumed.

NO_RECURSE

```
# If set to True, ``-r``` will not parse sub directories in this folder.
NO_RECURSE = False
```

If set to True, `-r` will not parse sub directories in this folder. If this does not exist, the default of `False` is used. The main purpose of this is to prevent scanning child folders when it is already known that there are no child folders that need processing or that `DEPENDENCIES` lists every child folder of interest, so recursion is not necessary.

4.4.5 Main

```
# If called as a command line and not a class, perform the build
if __name__ == "__main__":
    sys.exit(build(os.path.dirname(os.path.abspath(__file__)), 'all'))
```

The `build_rules.py` file can be run as a standalone script. If the line above exists in the script, it will call whatever function is declared and exit to the operating system. The example above will call the `build()` function with the current directory, however, it's up to the programmer to decide what is the default action and if parameters should be passed and what to do with them.

4.5 Constants

4.5.1 Setup strings

These strings are used for version control and setup.py for distribution.

__numversion__

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

__version__

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

__author__

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

__title__

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

__summary__

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

__uri__

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

__email__

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

__license__

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

__copyright__

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

4.5.2 Internal constants

Constants used internally by this package.

_XCODEPROJ_MATCH

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

4.5.3 Internal tables

enums._FILETYPES_LOOKUP

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

enums._FILETYPES_READABLE

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

enums._IDETYPES_CODES

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

enums._IDETYPES_READABLE

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

enums._PLATFORMTYPES_CODES

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

enums._PLATFORMTYPES_EXPANDED

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

enums._PLATFORMTYPES_READABLE

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

enums._PLATFORMTYPES_VS

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

enums._PROJECTTYPES_READABLE

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

4.5.4 Folder locations

config.BUILD_RULES_PY

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

config._BUILD_RULES_VAR

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

config.USER_HOME

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

config.PROJECTS_HOME

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

config.DEFAULT_BUILD_RULES

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

4.5.5 Build Constants

buildme.BUILD_LIST

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

buildme.CODEWARRIOR_ERRORS

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

buildme._CW_SUPPORTED_LINKERS

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

buildme._VS_VERSION_YEARS

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

buildme._VS_OLD_VERSION_YEARS

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

buildme._VS_SDK_ENV_VARIABLE

Warning: doxygenvariable: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/t

4.6 Classes

4.6.1 Enumerations

enums.FileTypes

Warning: doxygenclass: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/tem

enums.IDETypes

Warning: doxygenclass: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/tem

enums.PlatformTypes

Warning: doxygenclass: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/tem

enums.ProjectTypes

Warning: doxygenclass: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/tem

4.6.2 Project Classes

core.Attributes

Warning: doxygenclass: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/tem

core.SourceFile

Warning: doxygenclass: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/tem

core.Configuration

Warning: doxygenclass: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/tem

core.Project

Warning: doxygenclass: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/tem

core.Solution

Warning: doxygenclass: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/tem

4.6.3 Build Classes

core.BuildError

Warning: doxygenclass: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/tem

core.BuildObject

Warning: doxygenclass: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/tem

4.7 Functions

4.7.1 Dispatchers

makeprojects.build

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

makeprojects.clean

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

makeprojects.rebuild

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

4.7.2 Generators

makeprojects.new_solution

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

makeprojects.new_project

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

makeprojects.new_configuration

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

4.7.3 Configuration

config.save_default

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

config.find_default_build_rules

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

4.7.4 Clean

cleanme.dispatch

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

cleanme.process

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

cleanme.main

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

4.7.5 Build

buildme.build_rez_script

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

buildme.build_slicer_script

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

buildme.build_doxygen

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

buildme.build_watcom_makefile

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

buildme.build_makefile

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

buildme.parse_sln_file

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

buildme.build_visual_studio

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

buildme.parse_mcp_file

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

buildme.build_codewarrior

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

buildme.parse_xcodeproj_file

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

buildme.build_xcode

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

buildme.parse_codeblocks_file

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

buildme.build_codeblocks

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

buildme.add_build_rules

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

buildme.add_project

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

buildme.get_projects

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

buildme.process

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

buildme.main

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

4.7.6 Rebuild

rebuild.main

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

4.7.7 Enums

enums.get_installed_visual_studio

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

enums.get_installed_xcode

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

enums.platformtype_short_code

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

4.7.8 Util

util.validate_enum_type

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

util.regex_dict

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

util.validate_boolean

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

util.validate_string

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

util.source_file_filter

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

util.add_build_rules

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

util.get_build_rules

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

util.getattr_build_rules

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

util.remove_ending_os_sep

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

util.was_processed

Warning: doxygenfunction: Cannot find file: /home/docs/checkouts/readthedocs.org/user_builds/makeprojects/checkouts/latest/docs/

4.8 License

4.8.1 MIT License

The gist of the license… Have fun using this code, I won't sue you and you can't sue me. However, please be nice about it and give me a credit in your software that you used my code in.

Please?
