kmdo

Release 1.0.6

Aug 08, 2023

Contents

Maintaining documentation can be prone to error and cumbersome. Especially for demos, tutorials, and usage guides for command-line tools. To aid that, `kmdo` runs through a directory and executes command-files, storing `stdout` and `stderr` in corresponding output-files.

This documentation is an example of how this can be used in concert with Sphinx and Read the Docs.

CHAPTER 1

Installation

Install `kmdo` system-wide via the pip using `pipx`:

```
pipx install kmdo
```

Note: Ensure that command-line tools installed using `pipx` are added to `PATH` by running `pipx ensurepath`.

Usage

```
usage: kmdo [-h] [-r] [-s SHELL] path

Run commands from .cmd files, storing output in .out files

positional arguments:
  path                  Path to DIR containing .cmd files

options:
  -h, --help            show this help message and exit
  -r, --recursive       go deepah!
  -s SHELL, --shell SHELL
                        Absolute path to the Shell to use
```

2.1 Error-handling

kmdo has exit code 0 upon success, that is when all commands succeed, ignoring command errors from command-files with .uone in the file name. On error, kmdo has a non zero exit code.

Additionally, kmdo outputs a **YAML** representation of what it has executed to stdout. For example, when using kmdo to generate command output for the documentation you are reading now.

```
kmdo src/examples
```

Outputs the following **YAML**:

```
args:
  path: '/Users/safl/git/kmdo/docs/src/examples'
  recursive: False
results:
- out_fp: '/Users/safl/git/kmdo/docs/src/examples/kmdo.out'
  cmd_fp: '/Users/safl/git/kmdo/docs/src/examples/kmdo.cmd'
```

```
  cmd: 'kmdo --help'
  rcode: 0
  uone: False
  err: False
- out_fp: '/Users/safl/git/kmdo/docs/src/examples/kmdo.uone.out'
  cmd_fp: '/Users/safl/git/kmdo/docs/src/examples/kmdo.uone.cmd'
  cmd: 'kmdo'
  rcode: 2
  uone: True
  err: False
nerrs: 0
```

2.2 Empty command-file and update-on-error

When the command-file is empty, then the **fname** part of the command-file file name is treated as the command to execute.

For example, the empty file named `kmdo.uone.cmd`, will execute the command `kmdo`, and because of `.uone` in the file name then it create the output file `kmdo.uone.out`:

```
usage: kmdo [-h] [-r] [-s SHELL] path
kmdo: error: the following arguments are required: path
```