

Taskwarrior: Git Integration

Eric Bailey

July 27, 2019 ¹

¹ Last updated July 27, 2019

To-Do

Describe Taskwarrior hooks, specifically this one.	1
Provide rationale and description of each option.	1
Teach finduses how to parse parameter expansions.	2
Add references, e.g. Taskwarrior documentation.	3

Describe Taskwarrior hooks, specifically this one.

Front matter

Use a fairly portable Bash shebang.

1a `< * 1a>≡`
`#!/usr/bin/env bash`

This definition is continued in chunks 1 and 2.
Root chunk (not used in this document).

Be safe.

1b `< * 1a>+≡`
`set -eufo pipefail`

Provide rationale and description
of each option.

Argument parsing

I only care about the **args**, **command**, and **data** directory.

1c `< * 1a>+≡`
`args="${2#args:}"`
`command="${3#command:}"`
`data="${5#data:}"`

Defines:

args, never used.
command, used in chunk 2h.
data, used in chunk 2d.

I don't care about the **api** version, **rc**
file, or Taskwarrior **version**.

```
api="${1#api:}"  
rc="${4#rc:}"  
version="${6#version:}"
```

If *<there are no changes 2a>* don't commit anything, and **exit** successfully, printing an informative message if *<DEBUG is set 2b>*.

```
2c <* 1a>+≡
    if <there are no changes 2a>; then
        if <DEBUG is set 2b>; then
            echo 'No changes to commit'
        fi
        exit 0
    fi
```

Git integration

If present, *<stage the changes 2e>* or die trying.

```
2f <* 1a>+≡
    if ! <stage the changes 2e>; then
        echo 'Failed to add files to the index'
        exit 100
    fi
```

Try to *<commit the changes 2g>*.

```
2i <* 1a>+≡
    if ! <commit the changes 2g>; then
        echo 'Failed to record changes to the repository'
        exit 101
    fi
```

If *<DEBUG is set 2b>*, print a brief summary of the commit.

```
2j <* 1a>+≡
    if <DEBUG is set 2b>; then
        <git 2d> log -oneline -1
    fi
```

```
2a <there are no changes 2a>≡
    <git 2d> diff -quiet
```

This code is used in chunk 2c.

```
2b <DEBUG is set 2b>≡
    [ -n "${DEBUG:-}" ]
```

This code is used in chunk 2.

Run **git** as if it were started in the Taskwarrior data directory.

```
2d <git 2d>≡
    git -C "$data" -C "$data"
```

This code is used in chunk 2.
Uses data 1c.

```
2e <stage the changes 2e>≡
    <git 2d> add -A
```

This code is used in chunk 2f.

Quietly store the current contents of the index in a new commit along with a log *<message 2h>* describing the changes.

```
2g <commit the changes 2g>≡
    <git 2d> commit -qm "<message 2h>"
```

This code is used in chunk 2i.

```
2h <message 2h>≡
    $command: ${args#task $command }
```

This code is used in chunk 2g.
Uses command 1c.

Teach **finduses** how to parse parameter expansions.

Chunks

⟨* [1a](#)⟩ [1a](#), [1b](#), [1c](#), [2c](#), [2f](#), [2i](#), [2j](#)
 ⟨commit the changes [2g](#)⟩ [2g](#), [2i](#)
 ⟨DEBUG is set [2b](#)⟩ [2b](#), [2c](#), [2j](#)
 ⟨git [2d](#)⟩ [2a](#), [2d](#), [2e](#), [2g](#), [2j](#)
 ⟨message [2h](#)⟩ [2g](#), [2h](#)
 ⟨stage the changes [2e](#)⟩ [2e](#), [2f](#)
 ⟨there are no changes [2a](#)⟩ [2a](#), [2c](#)

Index

args: [1c](#)
 command: [1c](#), [2h](#)
 data: [1c](#), [2d](#)

References

Add references, e.g. Taskwarrior documentation.