
pwclient

Stephen Finucane

May 04, 2023

CONTENTS

1	Installation	3
2	Getting Started	5
3	Development	7
4	Documentation	9
4.1	Usage	9
4.2	Configuration	15
4.3	Release Notes	16
	Index	21

pwclient is a VCS-agnostic tool for interacting with [Patchwork](#), the web-based patch tracking system.

INSTALLATION

The easiest way to install *pwclient* and its dependencies is using `pip`. To do so, run:

```
$ python3 -m pip install pwclient
```

You can also install *pwclient* manually. First, install the required dependencies. On Fedora, run:

```
$ sudo dnf install python-pbr
```

On Ubuntu, run:

```
$ sudo apt-get install python-pbr
```

Once dependencies are installed, clone this repo and run `setup.py`:

```
$ git clone https://github.com/getpatchwork/pwclient
$ cd pwclient
$ python3 -m pip install --user .
```


GETTING STARTED

To use *pwclient*, you will need a `.pwclientrc` file, located in your home directory (`$HOME` or `~`). You can point to another path with the environment variable `PWCLIENTRC`. Patchwork itself provides sample `.pwclientrc` files for projects at `/project/{projectName}/pwclientrc/`. For example, [here](#) is the `.pwclientrc` file for Patchwork itself.

DEVELOPMENT

If you're interested in contributing to *pwclient*, first clone the repo:

```
$ git clone https://github.com/getpatchwork/pwclient  
$ cd pwclient
```

Create a *virtualenv*, then install the package in *editable* mode:

```
$ virtualenv .venv  
$ source .venv/bin/activate  
$ python3 -m pip install --editable .
```


DOCUMENTATION

4.1 Usage

4.1.1 pwclient

```
usage: pwclient [-h]
                {apply,git-am,get,info,projects,check-get,check-list,check-info,check-
↪create,states,view,update,list,search}
                ...
```

-h, --help

show this help message and exit

Use 'pwclient <command> -help' for more info.

To avoid unicode encode/decode errors, you should export the LANG or LC_ALL environment variables according to the configured locales on your system. If these variables are already set, make sure that they point to valid and installed locales.

pwclient apply

```
usage: pwclient apply [--help] [-h] [-p PROJECT] PATCH_ID [PATCH_ID ...]
```

patch_id

patch ID

--help

show this help message and exit

-h, --use-hashes

lookup by patch hash

-p <project>, --project <project>

lookup patch in project

pwclient check-create

```
usage: pwclient check-create [--help] [-h] [-p PROJECT] [-c CONTEXT]
                             [-s {pending,success,warning,fail}]
                             [-u TARGET_URL] [-d DESCRIPTION]
                             PATCH_ID [PATCH_ID ...]
```

patch_id

patch ID

--help

show this help message and exit

-h, --use-hashes

lookup by patch hash

-p <project>, --project <project>

lookup patch in project

-c <context>, --context <context>

-s {pending,success,warning,fail}, --state {pending,success,warning,fail}

-u <target_url>, --target-url <target_url>

-d <description>, --description <description>

pwclient check-get

```
usage: pwclient check-get [--help] [-h] [-p PROJECT] [-f FORMAT]
                           PATCH_ID [PATCH_ID ...]
```

patch_id

patch ID

--help

show this help message and exit

-h, --use-hashes

lookup by patch hash

-p <project>, --project <project>

lookup patch in project

-f <format>, --format <format>

print output in the given format. You can use tags matching fields, e.g. `%{context}`, `%{state}`, or `%{msgid}`.

pwclient check-info

```
usage: pwclient check-info [-h] [PATCH_ID] CHECK_ID
```

patch_id

patch ID (required if using the REST API backend)

check_id

check ID

-h, --help

show this help message and exit

pwclient check-list

```
usage: pwclient check-list [-h] [-u USER] [PATCH_ID]
```

patch_id

patch ID (required if using the REST API backend)

-h, --help

show this help message and exit

-u <user>, --user <user>

user (name or ID) to filter checks by

pwclient get

```
usage: pwclient get [--help] [-h] [-p PROJECT] PATCH_ID [PATCH_ID ...]
```

patch_id

patch ID

--help

show this help message and exit

-h, --use-hashes

lookup by patch hash

-p <project>, --project <project>

lookup patch in project

pwclient git-am

```
usage: pwclient git-am [--help] [-h] [-p PROJECT] [-s] [-3] [-m]
      PATCH_ID [PATCH_ID ...]
```

patch_id

patch ID

pwclient

--help

show this help message and exit

-h, --use-hashes

lookup by patch hash

-p <project>, --project <project>

lookup patch in project

-s, --signoff

pass ‘-signoff’ to ‘git-am’

-3, --3way

pass ‘-3way’ to ‘git-am’

-m, --msgid

pass ‘-message-id’ to ‘git-am’

pwclient info

```
usage: pwclient info [--help] [-h] [-p PROJECT] PATCH_ID [PATCH_ID ...]
```

patch_id

patch ID

--help

show this help message and exit

-h, --use-hashes

lookup by patch hash

-p <project>, --project <project>

lookup patch in project

pwclient list

```
usage: pwclient list [-h] [-s STATE] [-a ARCHIVED] [-p PROJECT] [-w WHO]
                    [-d WHO] [-n MAX#] [-N MAX#] [-m MESSAGEID] [-H HASH]
                    [-f FORMAT]
                    [STR]
```

str

substring to search for patches by name

-h, --help

show this help message and exit

-s <state>, --state <state>

filter by patch state (e.g., ‘New’, ‘Accepted’, etc.)

-a, --archived

filter by patch archived state

-p <project>, **--project** <project>
filter by project name (see ‘projects’ for list)

-w <who>, **--submitter** <who>
filter by submitter (name, e-mail substring search)

-d <who>, **--delegate** <who>
filter by delegate (name, e-mail substring search)

-n <max#>
limit results to first n

-N
limit results to last N

-m <messageid>, **--msgid** <messageid>
filter by Message-Id

-H <hash>, **--hash** <hash>
filter by hash

-f <format>, **--format** <format>
print output in the given format. You can use tags matching fields, e.g. `%{id}`, `%{state}`, or `%{msgid}`.

pwclient projects

```
usage: pwclient projects [-h]
```

-h, **--help**
show this help message and exit

pwclient search

```
usage: pwclient search [-h] [-s STATE] [-a ARCHIVED] [-p PROJECT] [-w WHO]
                        [-d WHO] [-n MAX#] [-N MAX#] [-m MESSAGEID] [-H HASH]
                        [-f FORMAT]
                        [STR]
```

str
substring to search for patches by name

-h, **--help**
show this help message and exit

-s <state>, **--state** <state>
filter by patch state (e.g., ‘New’, ‘Accepted’, etc.)

-a, **--archived**
filter by patch archived state

-p <project>, **--project** <project>
filter by project name (see ‘projects’ for list)

pwclient

-w <who>, **--submitter** <who>
filter by submitter (name, e-mail substring search)

-d <who>, **--delegate** <who>
filter by delegate (name, e-mail substring search)

-n <max#>
limit results to first n

-N
limit results to last N

-m <messageid>, **--msgid** <messageid>
filter by Message-Id

-H <hash>, **--hash** <hash>
filter by hash

-f <format>, **--format** <format>
print output in the given format. You can use tags matching fields, e.g. `%{id}`, `%{state}`, or `%{msgid}`.

pwclient states

```
usage: pwclient states [-h]
```

-h, **--help**
show this help message and exit

pwclient update

```
usage: pwclient update [--help] [-h] [-p PROJECT] [-c COMMIT-REF] [-s STATE]
                        [-a {yes,no}]
                        PATCH_ID [PATCH_ID ...]
```

patch_id

patch ID

--help

show this help message and exit

-h, --use-hashes

lookup by patch hash

-p <project>, --project <project>

lookup patch in project

-c <commit-ref>, --commit-ref <commit-ref>

commit reference hash

-s <state>, --state <state>

set patch state (e.g., 'Accepted', 'Superseded' etc.)

-a {yes,no}, --archived {yes,no}

set patch archived state

using a COMMIT-REF allows for only one ID to be specified

pwclient view

```
usage: pwclient view [--help] [-h] [-p PROJECT] PATCH_ID [PATCH_ID ...]
```

patch_id

patch ID

--help

show this help message and exit

-h, --use-hashes

lookup by patch hash

-p <project>, --project <project>

lookup patch in project

4.2 Configuration

pwclient reads configuration from the `.pwclientrc` file, located in your home directory (`$HOME` or `~`). You can point to another path with the environment variable `PWCLIENTRC`. Patchwork itself provides sample `.pwclientrc` files for projects at:

`/project/{projectName}/pwclientrc/`

For example, [here](#) is the `.pwclientrc` file for Patchwork itself.

4.2.1 Format

The `.pwclientrc` file is an [INI-style](#) config file, **containing** an `options` section along with a section for each project.

The options section provides the following configuration options:

default

The default project to use. Must be configured if not specifying a project via the command line.

signoff

Add a Signed-Off-By: line to commit messages when applying patches using the **git-am** command. Defaults to `False`.

3way

Enable three-way merge when applying patches using the **git-am** command. Defaults to `False`.

msgid

Add a Message-Id: line to commit messages when applying patches using the **git-am** command. Defaults to `False`.

The names of the project sections must correspond to the project names in Patchwork, as reflected in the project's URL in Patchwork. Multiple projects can be defined, but no two projects can share the same name. Project sections require the following configuration options:

url

The URL of the API endpoint for the Patchwork instance that the project is available on. This depends on the API *backend* in use. For the `rest` backend, this will typically be `$PATCHWORK/api`. For example:

<https://patchwork.ozlabs.org/api>

For the `xmlrpc` backend, this is will typically be `$PATCHWORK_URL/xmlrpc`. For example:

<https://patchwork.ozlabs.org/xmlrpc>

In addition, the following options are optional:

backend

The API backend to use. One of: `rest`, `xmlrpc`

username

Your Patchwork username.

password

Your Patchwork password.

token

Your Patchwork API token. (only supported with `rest` backend)

Note: Patchwork credentials are only needed for certain operations, such as updating the state of a patch. You will also require admin privileges on the instance in question.

4.2.2 Example

```
[options]
default = patchwork

[patchwork]
backend = rest
url = http://patchwork.ozlabs.org/api/
token = 088cade25e52482e6486794ef4a4561d3e5fe727
```

4.2.3 Legacy Format

Older Patchwork instances may provide a legacy version of the `.pwclientrc` file that did not support multiple projects. `pwclient` will automatically convert this version of the file to the latest version.

4.3 Release Notes

4.3.1 2.5.0

New Features

- The `pwclient list` command now accepts a `--hash` option to list patches by hash.

4.3.2 2.4.0

New Features

- Python 3.10 support has been added.

4.3.3 2.3.0

Prelude

pwclient version 2.3.0 fixes a number of bugs introduced in 2.2.0 as part of the Patchwork REST API feature.

4.3.4 2.2.0

Prelude

pwclient version 2.2 introduces preliminary support for the Patchwork REST API, first introduced in Patchwork 2.0. We encourage users to test this functionality by setting the new `[$project]` backend setting to `rest`. Please report any bugs encountered. More details are provided below and in the documentation.

New Features

- The `check-list` and `check-create` commands now accept a `-h` (help) paramter.
- It is now possible to set the path of the `pwclientrc` file using the `PWCLIENRC` environment variable. This can be useful for automation.
- *pwclient* now provides experimental support for the REST API first introduced in Patchwork 2.0. This can be enabled by adding the `backend = rest` setting to your `pwclientrc` file. For example:

```
[options]
default = patchwork

[patchwork]
backend = rest
url = https://patchwork.ozlabs.org/api/
token = 088cade25e52482e6486794ef4a4561d3e5fe727
```

There are likely bugs and features gaps in this implementation. Report any bugs to either [the mailing list](#) or [GitHub issue tracker](#).

Upgrade Notes

- Python 3.6 support has been removed.
- Python 3.10 support has been added.
- The `check-list` command now accepts a `PATCH_ID` positional argument and `--user USER` option to allow users to filter checks by patch and creator respectively. The `PATCH_ID` argument is currently optional for the XML-RPC API backend but is required for the REST API backend.

- The `check-info` command now takes two positional arguments, `PATCH_ID` and `CHECK_ID`. Previously, it only took `CHECK_ID`. The `PATCH_ID` argument remains optional for the XML-RPC API backend but is required for the REST API backend.

4.3.5 2.0.0

Upgrade Notes

- Support for Python 2.7 and 3.5 has been dropped. The minimum Python version now supported is Python 3.6.

4.3.6 1.3.0

New Features

- Add a new `pwclient check-get` command to query the checks run against a specific patch.

4.3.7 1.2.0

New Features

- The `pwclient git-am` command can now passthrough the `-m` flag to `-m`.

4.3.8 1.1.1

Upgrade Notes

- PyPy is no longer officially supported.

4.3.9 1.1.0

New Features

- Most options now have a long opt equivalent. For example:
\$ `pwclient update --archived yes 123`

Bug Fixes

- The `pwclient view` command will now decode received mboxes on Python 2.7.

4.3.10 1.0.0

Prelude

Initial release of *pwclient* package.

Symbols

- 3
 - pwclient-git-am command line option, 12
- H
 - pwclient-list command line option, 13
 - pwclient-search command line option, 14
- N
 - pwclient-list command line option, 13
 - pwclient-search command line option, 14
- 3way
 - pwclient-git-am command line option, 12
- archived
 - pwclient-list command line option, 12
 - pwclient-search command line option, 13
 - pwclient-update command line option, 14
- commit-ref
 - pwclient-update command line option, 14
- context
 - pwclient-check-create command line option, 10
- delegate
 - pwclient-list command line option, 13
 - pwclient-search command line option, 14
- description
 - pwclient-check-create command line option, 10
- format
 - pwclient-check-get command line option, 10
 - pwclient-list command line option, 13
 - pwclient-search command line option, 14
- hash
 - pwclient-list command line option, 13
 - pwclient-search command line option, 14
- help
 - pwclient command line option, 9
 - pwclient-apply command line option, 9
 - pwclient-check-create command line option, 10
 - pwclient-check-get command line option, 10
 - pwclient-check-info command line option, 11
 - pwclient-check-list command line option, 11
 - pwclient-get command line option, 11
 - pwclient-git-am command line option, 11
 - pwclient-info command line option, 12
 - pwclient-list command line option, 12
 - pwclient-projects command line option, 13
 - pwclient-search command line option, 13
 - pwclient-states command line option, 14
 - pwclient-update command line option, 14
 - pwclient-view command line option, 15
- msgid
 - pwclient-git-am command line option, 12
 - pwclient-list command line option, 13
 - pwclient-search command line option, 14
- project
 - pwclient-apply command line option, 9
 - pwclient-check-create command line option, 10
 - pwclient-check-get command line option, 10
 - pwclient-get command line option, 11
 - pwclient-git-am command line option, 12
 - pwclient-info command line option, 12
 - pwclient-list command line option, 12
 - pwclient-search command line option, 13
 - pwclient-update command line option, 14
 - pwclient-view command line option, 15
- signoff
 - pwclient-git-am command line option, 12
- state
 - pwclient-check-create command line option, 10
 - pwclient-list command line option, 12
 - pwclient-search command line option, 13
 - pwclient-update command line option, 14
- submitter
 - pwclient-list command line option, 13
 - pwclient-search command line option, 13
- target-url
 - pwclient-check-create command line

option, 10

--use-hashes

- pwclient-apply command line option, 9
- pwclient-check-create command line option, 10
- pwclient-check-get command line option, 10
- pwclient-get command line option, 11
- pwclient-git-am command line option, 12
- pwclient-info command line option, 12
- pwclient-update command line option, 14
- pwclient-view command line option, 15

--user

- pwclient-check-list command line option, 11

-a

- pwclient-list command line option, 12
- pwclient-search command line option, 13
- pwclient-update command line option, 14

-c

- pwclient-check-create command line option, 10
- pwclient-update command line option, 14

-d

- pwclient-check-create command line option, 10
- pwclient-list command line option, 13
- pwclient-search command line option, 14

-f

- pwclient-check-get command line option, 10
- pwclient-list command line option, 13
- pwclient-search command line option, 14

-h

- pwclient command line option, 9
- pwclient-apply command line option, 9
- pwclient-check-create command line option, 10
- pwclient-check-get command line option, 10
- pwclient-check-info command line option, 11
- pwclient-check-list command line option, 11
- pwclient-get command line option, 11
- pwclient-git-am command line option, 12
- pwclient-info command line option, 12
- pwclient-list command line option, 12
- pwclient-projects command line option, 13
- pwclient-search command line option, 13
- pwclient-states command line option, 14
- pwclient-update command line option, 14
- pwclient-view command line option, 15

-m

- pwclient-git-am command line option, 12
- pwclient-list command line option, 13
- pwclient-search command line option, 14

-n

- pwclient-list command line option, 13
- pwclient-search command line option, 14

-p

- pwclient-apply command line option, 9
- pwclient-check-create command line option, 10
- pwclient-check-get command line option, 10
- pwclient-get command line option, 11
- pwclient-git-am command line option, 12
- pwclient-info command line option, 12
- pwclient-list command line option, 12
- pwclient-search command line option, 13
- pwclient-update command line option, 14
- pwclient-view command line option, 15

-s

- pwclient-check-create command line option, 10
- pwclient-git-am command line option, 12
- pwclient-list command line option, 12
- pwclient-search command line option, 13
- pwclient-update command line option, 14

-u

- pwclient-check-create command line option, 10
- pwclient-check-list command line option, 11

-w

- pwclient-list command line option, 13
- pwclient-search command line option, 13

C

check_id

- pwclient-check-info command line option, 11

P

patch_id

- pwclient-apply command line option, 9
- pwclient-check-create command line option, 10
- pwclient-check-get command line option, 10
- pwclient-check-info command line option, 11
- pwclient-check-list command line option, 11
- pwclient-get command line option, 11
- pwclient-git-am command line option, 11
- pwclient-info command line option, 12

```

    pwclient-update command line option, 14
    pwclient-view command line option, 15
pwclient command line option
    --help, 9
    -h, 9
pwclient-apply command line option
    --help, 9
    --project, 9
    --use-hashes, 9
    -h, 9
    -p, 9
    patch_id, 9
pwclient-check-create command line option
    --context, 10
    --description, 10
    --help, 10
    --project, 10
    --state, 10
    --target-url, 10
    --use-hashes, 10
    -c, 10
    -d, 10
    -h, 10
    -p, 10
    -s, 10
    -u, 10
    patch_id, 10
pwclient-check-get command line option
    --format, 10
    --help, 10
    --project, 10
    --use-hashes, 10
    -f, 10
    -h, 10
    -p, 10
    patch_id, 10
pwclient-check-info command line option
    --help, 11
    -h, 11
    check_id, 11
    patch_id, 11
pwclient-check-list command line option
    --help, 11
    --user, 11
    -h, 11
    -u, 11
    patch_id, 11
pwclient-get command line option
    --help, 11
    --project, 11
    --use-hashes, 11
    -h, 11
    -p, 11
    patch_id, 11
    pwclient-git-am command line option
        -3, 12
        --3way, 12
        --help, 11
        --msgid, 12
        --project, 12
        --signoff, 12
        --use-hashes, 12
        -h, 12
        -m, 12
        -p, 12
        -s, 12
        patch_id, 11
    pwclient-info command line option
        --help, 12
        --project, 12
        --use-hashes, 12
        -h, 12
        -p, 12
        patch_id, 12
    pwclient-list command line option
        -H, 13
        -N, 13
        --archived, 12
        --delegate, 13
        --format, 13
        --hash, 13
        --help, 12
        --msgid, 13
        --project, 12
        --state, 12
        --submitter, 13
        -a, 12
        -d, 13
        -f, 13
        -h, 12
        -m, 13
        -n, 13
        -p, 12
        -s, 12
        -w, 13
        str, 12
    pwclient-projects command line option
        --help, 13
        -h, 13
    pwclient-search command line option
        -H, 14
        -N, 14
        --archived, 13
        --delegate, 14
        --format, 14
        --hash, 14
        --help, 13
        --msgid, 14

```

- project, 13
- state, 13
- submitter, 13
- a, 13
- d, 14
- f, 14
- h, 13
- m, 14
- n, 14
- p, 13
- s, 13
- w, 13
- str, 13

pwclient-states command line option

- help, 14
- h, 14

pwclient-update command line option

- archived, 14
- commit-ref, 14
- help, 14
- project, 14
- state, 14
- use-hashes, 14
- a, 14
- c, 14
- h, 14
- p, 14
- s, 14

patch_id, 14

pwclient-view command line option

- help, 15
- project, 15
- use-hashes, 15
- h, 15
- p, 15

patch_id, 15

S

str

- pwclient-list command line option, 12
- pwclient-search command line option, 13