awftp Documentation

Release 1.3.4

Thorsten Simons

Dec 15, 2021

Contents

A FTP-style client for HCP Anywhere File Sync & Share

HCP Anywhere is a File Sync & Share solution developed by Hitachi Vantara. It offers a wide range of FS&S clients for Desktop computers, Mobile devices as well as Browsers. A lacking piece so far is a CLI client that allows to access HCP Anywhere on devices where no GUI is available (Linux servers, for example).

awftp tries to fill this gap by providing a look and feel as close as possible to well-known CLI-based ftp clients. The features available are a subset of what is offered by ftp clients, due to the functionality the HCP Anywhere FS&S service offers through its API; some other functions in ftp clients simply doesn't make sense for use with HCP Anywhere. On the other hand, there are some features (*snap*, for example) not found with ftp...

Features

- Works with any FS&S API version >= 2.1.1 while using the highest version known
- Navigate the folders stored in HCP Anywhere, including Mobilized NAS, Shared Folders, Team Folders, Backup Folders
- List folders content, including deleted files/folders
- Store, update and retrieve files
- Move files and folders
- Undelete files/folders
- Create/delete folders (even recursive)
- Link handling:
  - Create links to share content with others - internal/public, read-only, read-write, write-only
  - List links
  - Delete links
  - Work with folders shared with us:
    - List all collaboration folders
    - List invitations
    - Accept invitations
    - Reject invitations
- Work with snapshots
  - List available snapshots
  - Provide view into snapshots
  - Restore files from snapshots
- Dynamically enables/disables features depending on the HCP Anywhere version connected to

CHAPTER 1

Installation

1.1 Using a local Python 3 installation

1.1.1 Dependencies

You need to have at least Python 3.7 installed to run **awftp**.

It uses the famous requests package[1] for communication with HCP Anywhere. And it uses click[2], for the progress bar and console output.

Tip: It's suggested to use a virtual environment to fence the dependency from your primary Python environment.

1.1.2 Installing on a local Python 3

Make sure you have Python 3.7 (or better) installed

In case it's not installed, get it here: https://www.python.org/downloads/.

There are two ways to install **awftp**:

1. System-wide
   - Install **awftp** by running:

     ```
     $ pip install awftp
     ```

   -or-

   - Get the source from gitlab.com[3], either
     - by downloading the source archive, or
     - by cloning the repository:

[1] http://docs.python-requests.org/en/master/
[2] http://click.pocoo.org/6/
[3] https://gitlab.com/simont3/awftp

```
$ git clone https://gitlab.com/simont3/awftp
```

- Install locally, including the dependency:

```
$ python setup.py install
```

2. In a virtual environment

Linux / Mac OS X / etc.

- Create a fresh virtual environment:

```
$ python3 -m venv .awftp
```

- Activate the virtual environment:

```
$ source .awftp/bin/activate
```

- Update the base tools:

```
(.awftp) $ pip install -U pip setuptools
Requirement already up-to-date: pip in ./.awftp/lib/python3.6/site-packages
Collecting setuptools
  Downloading setuptools-36.0.1-py2.py3-none-any.whl (476kB)
Installing collected packages: setuptools
  Found existing installation: setuptools 28.8.0
    Uninstalling setuptools-28.8.0:
      Successfully uninstalled setuptools-28.8.0
Successfully installed setuptools-36.0.1
```

- Install **awftp**:

```
(.awftp) $ pip install awftp
Processing awftp-1.1.0-py3-none-any.whl
Collecting requests>=2.13.0 (from awftp==1.1.0)
  Downloading requests-2.18.1-py2.py3-none-any.whl (88kB)
Collecting click>=6.7 (from awftp==1.1.0)
  Using cached click-6.7-py2.py3-none-any.whl
Collecting beautifulsoup4>=4.6.0 (from awftp==1.1.0)
  Using cached beautifulsoup4-4.6.0-py3-none-any.whl
Collecting chardet<3.1.0,>=3.0.2 (from requests>=2.13.0->awftp==1.1.0)
  Using cached chardet-3.0.4-py2.py3-none-any.whl
Collecting certifi>=2017.4.17 (from requests>=2.13.0->awftp==1.1.0)
  Using cached certifi-2017.4.17-py2.py3-none-any.whl
Collecting idna<2.6,>=2.5 (from requests>=2.13.0->awftp==1.1.0)
  Using cached idna-2.5-py2.py3-none-any.whl
Collecting urllib3<1.22,>=1.21.1 (from requests>=2.13.0->awftp==1.1.0)
  Using cached urllib3-1.21.1-py2.py3-none-any.whl
Installing collected packages: chardet, certifi, idna, urllib3, requests, click,
↪ beautifulsoup4, awftp
Successfully installed awftp-1.1.0 beautifulsoup4-4.6.0 certifi-2017.4.17␣
↪chardet-3.0.4 click-6.7 idna-2.5 requests-2.18.1 urllib3-1.21.1
```

Now you can run **awftp** as long as you have the virtual environment activated:

```
(.awftp) $ awftp user:password@anywhere.your.domain
About to connect to anywhere.your.domain.
User user logged in.
Remote system type is HCP Anywhere, FS&S API is v3.0.0
Using binary mode to transfer files.
awftp>
```

CHAPTER 2

Run **awftp**

Tip: Make sure you have set your system language to something that understands **UTF-8** encoding, as you might run into display issues with filenames being coded in UTF-8 (from a Mac, for example).

```
$ export LC_ALL=en_US.UTF-8
```

Depending on how you installed **awftp**, you can start it on various ways:

- When using the provided binary (and given that you placed it somewhere in your `$PATH`):

  ```
  $ awftp <anywhere.your.domain>
  ```

- If you installed **awftp** through `pip install awftp`:

  ```
  $ awftp <anywhere.your.domain>
  ```

- If you cloned the git repository:;

  $ python3 awftp.py <anywhere.your.domain>

2.1 Arguments

```
$  awftp --help
usage: awftp.py [-h] [--version] [-d] [--idp] [--api {2.1.1,3.0.0,3.1.0}] \
                [-r CMDFILE] [--nossl] [aw_server]

positional arguments:
  aw_server             the HCP Anywhere system to connect ([user[:password]@]anywhere.your.
↪domain)

options:
  -h, --help            show this help message and exit
  --version             show program's version number and exit
  -d                    enable debugging output (ugly and a bit chatty!)
  --idp                 list available identity providers and exit
  --api {2.1.1,3.0.0,3.1.0}
```

(continues on next page)

(continued from previous page)

```
                    force using a specific FS&S API version
-r CMDFILE          run commands in CMDFILE; in case this shall be w/o interaction
                    entirely, don't forget ``bye`` as the last command!
--nossl             disable SSL (most likely, this won't work, as HCP Anywhere requires
                    SSL encryption)
```

2.2 Authentication

awftp will authenticate the user against the Active Directory HCP Anywhere is integrated with.

2.3 Proxy

In case there is a proxy needed to connect to HCP Anywhere, make sure you have set the required shell variables. You will need to have `https_proxy`, at least. See here for an explanation[4].

[4] https://www.gnu.org/software/wget/manual/html_node/Proxies.html

CHAPTER 3

Commands

Tip:

- **Most commands support output redirection using "|", ">" and ">>".**
- **File names containing spaces must either:**
  - **be placed in quotation marks**
  - **or blanks need to be masked with a backslash**
- **Running commands can be cancel by pressing CTRL-C**

3.1 bye

```
awftp> bye
```

Terminate awftp session and exit.

3.2 cd

```
awftp> cd [remote-directory]
```

Change remote working directory to remote-directory, if given, otherwise to /

3.3 cdup

```
awftp> cdup
```

Change remote working directory to parent directory.

3.4 close

```
awftp> close
```

Terminate the session with HCP Anywhere, but stay in awftp.

3.5 clear

```
awftp> clear
```

Clear the screen.

3.6 coll

```
awftp> coll [-b] [-t] [-s]
```

Collaboration: list folders shared with us

`-b` backup folders
`-t` team folders
`-s` shared folders

3.7 dir

```
awftp> dir [-1] [-d] [remote-path]
```

List contents of remote path

`-1` display one name per line
`-d` show deleted files, too

3.8 exit

```
awftp> exit
```

Terminate awftp session and exit

3.9 find

```
awftp> find [-1] [-d] [-s <snap_id>] <pattern>
```

Search for files/folders containing <pattern> in their name, starting at the current directory (in realtime or the active snapshot).

`-1` (digit one) display one name per line
`-d` search for deleted files, too
`-s` search in snapshot <snap_id> (from snap -l)

<pattern> is the string to search for, min. 3 characters

3.10 get

```
awftp> get remote-file [local-file]
```

Receive (download) a file.

3.11 help

```
awftp> help [command] | ? [command]
```

List the available commands or show help for the named command.

3.12 hist

```
awftp> hist <filename>
```

Show the history of a file.

3.13 invacc

```
awftp> invacc [-s] <Id>
```

Accept an outstanding share invitation.

´´-s´´ enable sync to desktop clients
´´<Id>´´ the invitation Id (obtain by using the ´´invls´´ command)

3.14 invls

```
awftp> invls
```

List outstanding share invitations.

3.15 invrej

```
awftp> invrej <Id>
```

Reject an outstanding share invitation.

´´<Id>´´ the invitation Id (obtain by using the ´´invls´´ command)

3.16 lcd

```
awftp> lcd [local-directory]
```

Change the local working dirkectory to local-directory (or to home directory, if local-directory isn't given).

3.17 link

```
awftp> link [-a] [-i|-p] -r|-u|-ru [expiration_days] file|folder
awftp> link -d <id>
```

Create a link to share a file or folder

-a add an access code
-i force creating an internal link
-p force creating a public link
-r the link is good for view and download of files
-u the link allows to upload into the linked folder
(at least one of -r and -u is required)

If expiration_date is not given, an unlimited link will be created

Delete an existing link

-d delete link with <id>, where <id> is the leftmost integer displayed by the *links* command

3.18 links

```
awftp> links
```

List all active links

3.19 ls

```
awftp> ls [-1] [-d] [remote-path]
```

List contents of remote path

-1 display one name per line
-d show deleted files, too
-u show names URL-encoded

Example:

```
awftp> ls
M drw     0.00   B 2017/05/12 12:21:24 home on NAS2
B dr-     0.00   B 2017/05/12 20:34:40 macth
S dr-     0.00   B 2017/05/13 21:12:07 sharedfolder_ro
S drw     0.00   B 2017/05/13 21:12:06 sharedfolder_rw
T drw     0.00   B 2017/05/13 21:04:26 teamfolder
- drw     0.00   B 2017/05/12 12:23:43 test
- -rw     4.64 KiB 2017/05/13 21:16:21 testfile.txt
```

The first character per line declares the type of the entry as being (or belonging to):

B: a backup folder
M: a mobilized NAS share
S: a shared folder
T: a team folder

3.20 lls

```
awftp> lls [local-path]
```

List contents of local path

3.21 lpwd

```
awftp> lpwd
```

Print the local working directory.

3.22 ls

```
awftp> ls [-d] [remote-path]
```

List contents of remote path.

`-d` show deleted files

3.23 mkdir

```
awftp> mkdir [-R] directory-name
```

Make directory on the remote machine.

`-R` recursively make parent directories as well.

3.24 mv

```
awftp> mv [-R] old_name new_name
```

Move a file or directory to a new name or position.

`-R` recursively make parent directories if required.

3.25 open

```
awftp> open [[user[:password]@]hcpanywhere-name]
```

Connect to an HCP Anywhere server

Be aware that there is a history file - think if you want to store your password in it...

3.26 progress

```
awftp> progress
```

Toggle progress meter off/on.

3.27 put

```
awftp> put [-u] local-file [remote-file]
```

Send one file.

-u update an existing file

3.28 pwd

```
awftp> pwd
```

Print the remote working directory.

3.29 quit

```
awftp> quit
```

Terminate awftp session and exit.

3.30 restore

```
awftp> restore [-d] remote-name
```

Make the version of a file or folder within the active snapshot the current version (restore it to "now").

-d restore a deleted file

3.31 rm

```
awftp> rm [-d [-R]] remote-name
```

Remove a remote file or folder

-d remove a folder
-R recursively delete a folder and *all* its content.

3.32 rmdir

```
awftp> rmdir [-R] directory-name
```

Remove a directory on the remote machine.

-R recursively delete a the directory and *all* its content.

New in version 1.3.4.

3.33 run

```
awftp> run <commandfile>
```

Run a batch of commands stored in <commandfile>.

3.34 search

```
awftp> search [-1] [-d] [-s <snap_id>] <pattern>
```

Search for files/folders containing <pattern> in their name, starting at the current directory (in realtime or the active snapshot)

- `-1` (digit one) display one name per line
- `-d` search for deleted files, too
- `-s` search in snapshot <snap_id> (from snap -l)

<pattern> is the string to search for, min. 3 characters

3.35 snap

```
awftp> snap -l | -s <index> | -u
```

Work with restore points (snapshots).

- `-l` list available snapshots
- `-s <index>` work on this snapshot
- `-u` unset snapshot (return to "now")

Once a snapshot has been set, all operations will be based on it.'

3.36 status

```
awftp> status
```

Show the session status.

3.37 time

```
awftp> time command [args]*
```

Run *command* as usual, but print the processing time afterwards.

3.38 user

```
awftp> user
```

Get information about the user's settings in HCP Anywhere.

CHAPTER 4

ToDos

4.1 Not yet implemented commands

Copy files remotely:

```
awftp> cp remote-source-file remote-target-file
```

Work with multiple files at once:

```
awftp> prompt
awftp> mput local-files
awftp> mget remote-files
awftp> mrm remote-files
```

4.2 Not yet implemented features

- regular expressions for file handling commands (`ls`, `get`, `put`, `rm`).

  That means, behavior should be similar what you expect if you run `ls somefolder/a*.txt` and alike.

CHAPTER 5

Release History

1.3.4 2021-12-15

- added the `run` command
- added the ability to start **awftp** with a command file, to allow commands to be executed manually.

1.3.3 2021-12-07

- output of `lls` now sorted by filename

1.3.2 2021-12-06

- minor change in MonitoredReader

1.3.1 2021-12-03

- documentation corrections

1.3.0 2021-11-30

- changed API version integration so that all version-specific features are enabled up to the version provided by the contacted AW
- max. API known is 3.1.0 now
- added `coll`, `invacc`, `invls`, `invrej` commands
- disabled SAML authentication, as it us not officially supported for FS&S API

1.2.5 2021-11-26

- fixed a display flaw in a situation where the login failed
- fixed a bug that caused not asking for a password if `open` was used without parameters

1.2.4 2021-11-25

- enabled the `progress` command
- fixed `get` so that it does not load the entire file to be downloaded in memory before saving to disk (credits to John Stegeman)
- fixed `put` so that it works even when `progress` is turned off
- the list of snapshots created by `snap -l` is now properly sorted
- fixed loading/saving the command history

1.2.2 2019-05-20

- tuned *get* to work properly with huge files

1.2.0 2018-97-20

- added wildcards to *ls*

1.1.6 2019-05-20

- tuned *get* to work properly with huge files

1.1.5 2017-11-16

- fixed a bug in *ls* that caused awftp to crash for some users

1.1.4 2017-11-01

- fixed a bug that made *readline* unusable on MacOS

1.1.3 2017-10-30

- fixed a bug that caused awftp to crash on the *lpwd* and *status* commands when the local directory doesn't exist (deleted underneath)

1.1.2 2017-09-22

- added a macOS binary
- now more tolerant when opening the history file fails

1.1.1 2017-08-29

- fixed a typo in the installation description (thanks to Guido H.)

1.1.0 2017-06-23

- added an option to list the Identity Providers supported by HCP Anywhere
- added an option to authenticate against a SAML IDP –> tested against ADFS SAML IDP only at this time
- removed the *idps* command, as it is useless now

1.0.3 2017-06-14

- fixed pip install

1.0.2 2017-06-14

- CTRL-D now equals the *bye* command
- new command:
  - *idps* - list identity providers supported by the AW server

1.0.1 2017-05-21

- fixed a bug that caused *ls* to fail while a snapshot was active

1.0.0 2017-05-19

- now supporting HCP Anywhere FS&S API 3.0.0
- added support for multiple FS&S API versions (always trying to use the highest one). New/updated commands:
  - *get* - added progress meter
  - *hist* - shows the history of a file
  - *link* - added `-d` to delete a link
  - *links* - list active links
  - *put* - added `-u` to update an existing file, added progress meter

  - *search* - allows to search for folder/files with a specific pattern in their name
  - *snap* - allows to work with the data at a given snapshot
  - *restore* - restore deleted files and previous versions of files
  - *user* - info about the user's settings in HCP Anywhere
- added ability to force using a specific FS&S API version (`--api`)
- added the possibility to cancel a command by CTRL-C w/o leaving **awftp**
- added handling loss of connection to Anywhere server
- added the ability to start awftp w/o telling a HCP Anywhere server on the command line (similar to ftp, use *open* inside **awftp**)
- Now catching UnicodeEncodeErrors to remind the user to use an UTF-8 system language

0.2.0 2017-04-02

- fixed a bug in setup.py

0.1.9 2017-02-15

- re-factored the code for single-file binary creation using pyinstaller
- Fixed a bug that caused awftp to crash when using **lls**
- Fixed a missing dependency for *click*

0.1.7 2016-11-09

- Fixed failing imports

0.1.5 2016-07-07

- Changed getting the user's id from pwd.getpwuid() to os.getuid() for Windows compatibility

0.1.4 2016-06-16

- some work to get it into pypi

0.1.0 2016-06-13

- initial release

CHAPTER 6

License / Trademarks

6.1 The MIT License (MIT)

6.2 Trademarks and Copyrights of used material