Pacifica CLI Documentation

David Brown

Jun 02, 2020

Contents:

The Pacifica CLI tool provides a user command line interface to Pacifica Core services. The CLI allows users to upload and download data in Pacifica.

CHAPTER 1

Installation

The Pacifica software is available through PyPi so creating a virtual environment to install is what is shown below. Please keep in mind compatibility with the Pacifica Core services.

1.1 Installation in Virtual Environment

These installation instructions are intended to work on both Windows, Linux, and Mac platforms. Please keep that in mind when following the instructions.

Please install the appropriate tested version of Python for maximum chance of success.

1.1.1 Linux and Mac Installation

```
mkdir ~/.virtualenvs
python -m virtualenv ~/.virtualenvs/pacifica
. ~/.virtualenvs/pacifica/bin/activate
pip install pacifica-cli
```

1.1.2 Windows Installation

This is done using PowerShell. Please do not use Batch Command.

```
mkdir "$Env:LOCALAPPDATA\virtualenvs"
python.exe -m virtualenv "$Env:LOCALAPPDATA\virtualenvs\pacifica"
& "$Env:LOCALAPPDATA\virtualenvs\pacifica\Scripts\activate.ps1"
pip install pacifica-cli
```

CHAPTER 2

Configuration

The `configure` subcommand generates a local configuration file for the user. It will read the system configuration to preseed its defaults and asks the user to enter the values required. An example configuration is located here.

The system configuration is processed first and the two directories are, `/etc/pacifica-cli/config.ini` then `PYTHON_PREFIX/pacifica-cli/config.ini`. Which ever is found first the client uses that as the system default.

The user configuration is processed second, if found. The directory the client looks in by default is `~/.pacifica_cli/config.ini`. The ~ translates to the users home directory on any platform.

Optionally, users can manage their config files in their home directory and set the `--config-ini` command-line argument. This switch will try to open the given file in `~/.pacifica_cli/` and merge it with the system configurations. Also, setting the environment variable `PACIFICA_CLI_INI` can also be sufficient to change the file name.

2.1 System Metadata

The metadata is managed by a JSON configuration file referenced by an environment variable `UPLOADER_CONFIG`. By default the environment variable is set to `uploader.json`. However, it could be managed at a system level or changed on the command line by the `--config` option.

The directories the `UPLOADER_CONFIG` are looked for in order are:

- `/etc/pacifica-cli/uploader.json`
- `VIRTUAL_ENV_ROOT/pacifica-cli/uploader.json`
- `~/.pacifica_cli/uploader.json`
- `$PWD/uploader.json`

The command line is evaluated last so it will override any of the previous paths.

The contents of the metadata configuration file is complex and should be read from here. Please get your systems administrator to help create this file for you. An example to start from is here.

CHAPTER 3

CLI Options

These are the command line options reference all the options are present here and have a description.

3.1 Global Options

These options are common to all commands and affects behavior of the cli.

--verbose This options requires an option which tells python logging what messages to print.

--config This specifies the system metadata configuration for use when uploading data.

3.2 Download Sub-Command Options

The download sub-command requires interfacing with the cartd service. The cart requires file metadata about what to download and there is several methods for getting that file metadata.

--destination Download the files to this folder (default to cwd).

--cloudevent Use the cloudevents file from the notifications service to download.

--transaction-id Setup a cart from this transaction ID.

3.3 Upload Sub-Command Options

There are two sets of options to the upload sub-command. The first comes from the missing values in the metadata configuration json file. The second are common for all uploads and will be documented further.

--follow-links This will follow symlinked directories as you walk a directory tree building the list of files to upload.

--nowait	The uploader will not wait for the successful or failed ingest of the upload. Instead you will have to query for this information later.
--local-retry	Upload an already generated bundle by the *–local-save* option. This option will honor the *–local-save* and *–tar-in-tar* options as well.
--local-save	The uploader will save off a copy of the uploaded bundle of data. The path is defined as the argument to the option.
--local-compress	Use a compression algorithm (gzip, bzip2) when creating the local save.
--tar-in-tar	A second bundler will be running generating a TAR in a TAR for upload.
--dry-run	Do just the query portion of the upload and print off what the metadata will be set to. This will not upload or generate a local save or retry.
--interactive	Interact with the results of the query engine to manually select the values for each metadata entry requested.
--do-not-upload	This forces the upload process to stop before uploading. This works well with the –local-save option.

CHAPTER 4

Example Usage

There are many things to consider when packaging up yours or someone elses data. We will help by going over some senerios we have experienced in the past and their solutions.

4.1 Example Configuration Output

```
$ pacifica-cli configure
Generating New Configuration.

Endpoints are an HTTP URL that looks similar to a website but
are designed for an uploader to interact with.

What are the endpoint URLs for the following...

Upload URL (https://ingest.example.com/upload):
Upload Status URL (https://ingest.example.com/get_state):
Upload Policy URL (https://policy.example.com/uploader):
Upload Validation URL (https://policy.example.com/ingest):
Download URL (https://cartd.example.com):
Download Policy URL (https://policy.example.com/status/transactions/by_id):

CA certificate bundle is the path to your certificate authority bundle.

Use this if you have a custom site SSL Certificate for your Site.

Valid values:
- True: verify the SSL server certificiate using system bundle
- False: do not verify the SSL server certificate (not recommended)
- a/path/to/a/cacert/bundle: custom path to the server certificate

CA Certificate Bundle (True):

There are three kinds of authentication types supported.
```

(continues on next page)

(continued from previous page)

```
- clientssl - This is where you have an SSL client key and cert
- basic     - This is a username and password
- gssapi    - Use GSSAPI tickets to authenticate
- None      - Do not perform any authentication

Authentication Type (None): basic
Username (None): jdoe
Password (None): password
```

4.2 Example Usage

4.2.1 Download Examples

```
$ pacifica-cli download --destination down --transaction-id 1234
```

4.2.2 Upload Examples

```
$ pacifica-cli upload --interactive test_file_upload.txt

Instrument ID - Select an ID
====================================

54 Nittany Liquid Probes - NMR PROBES: Nittany Liquid Probes
104 Transmogriscope - Nanoscale Transmogriscope
Select ID (54): 54

Project ID - Select an ID
====================================

1234a - Pacifica Development (active no close)
1235 - Pacifica Development (no close or end)
1236e - Pacifica Development (expired closed and end)
1237 - Pacifica Development (expired closed no end)
1238 - Pacifica Development (pre-active)
Select ID (1234a): 1234a
Done 10240.
Waiting job to complete (1).
Done.
{
    "created": "2017-09-26 01:32:34",
    "exception": "",
    "job_id": 1,
    "state": "OK",
    "task": "ingest metadata",
    "task_percent": "100.00000",
    "updated": "2017-09-26 01:32:36"
}
```

CHAPTER 5

CLI Python Module

5.1 Configure Module

Methods for configuring the client.

pacifica.cli.configure.**configure_url_endpoints**(*global_ini*)
Query and set the URL endpoints.

pacifica.cli.configure.**configure_ca_bundle**(*global_ini*)
Query for the ca bundle when using https.

pacifica.cli.configure.**configure_auth**(*global_ini*)
Query and set the authentication configuration.

5.2 Query Module

These are the query methods used for interactive query.

pacifica.cli.query.**execute_pager**(*content*)
Find the appropriate pager default is embedded python pager.

pacifica.cli.query.**filter_results**(*md_update*, *query_obj*, *regex*)
Filter the results of query_obj by regex and save result back into md_update.

pacifica.cli.query.**find_leaf_node**(*md_update*)
Find a leaf node that has all deps resolved.

pacifica.cli.query.**format_query_results**(*md_update*, *query_obj*)
Format the query results and return some data structures.

pacifica.cli.query.**interactive_select_loop**(*md_update*, *query_obj*, *default_id*)
While loop to ask users what they want to select.

pacifica.cli.query.**paged_content**(*title*, *display_data*, *valid_ids*)
Display the data yielding results.

pacifica.cli.query.**parse_command**(*exe*)
Walk the system path and return executable path.

pacifica.cli.query.**query_main**(*md_update*, *args*)
Query from the metadata configuration.

pacifica.cli.query.**remove_results**(*md_update*)
Remove the query results before passing it on.

pacifica.cli.query.**set_query_obj**(*dep_meta_ids*, *md_update*, *obj*)
Return the query object or false.

pacifica.cli.query.**set_results**(*md_update*, *query_obj*, *default_id*, *interactive=False*)
Set results of the query and ask if interactive.

pacifica.cli.query.**set_selected_id**(*selected_id*, *default_id*, *valid_ids*)
Return the selected ID validating it first.

5.3 Methods Module

Methods for the sub commands to run.

pacifica.cli.methods.**configure**(*args*, *_config_data*)
Configure the client by parsing current configuration.

pacifica.cli.methods.**download**(*args*, *_interface_data*)
Download data specified in args.

pacifica.cli.methods.**generate_global_config**(*config_ini='config.ini'*)
Generate a default configuration.

pacifica.cli.methods.**generate_requests_auth**(*global_ini*)
Generate arguments to requests for authentication.

pacifica.cli.methods.**query**(*args*, *interface_data*)
Query from the metadata configuration.

pacifica.cli.methods.**save_user_config**(*global_ini*)
Save the global config to the path.

pacifica.cli.methods.**set_environment_vars**(*global_ini*)
Set some environment variables to be used later.

pacifica.cli.methods.**set_verbose**(*verbose*)
Set the log level to arg value.

pacifica.cli.methods.**upload**(*args*, *interface_data*)
Upload the data based on bits.

pacifica.cli.methods.**verify_type**(*obj*)
Convert obj to requests verify argument.

Verify the type of obj that it will be consumed by requests verify option correctly.

5.4 Upload Module

The upload module used to send the data to ingest.

pacifica.cli.upload.**build_file_list_from_args**(*file_list*, *followlinks*)
Build a file list from args passed on cmdline.

pacifica.cli.upload.**check**(*status*)
Check the status since it's complicated.

pacifica.cli.upload.**check_okay**(*status*)
Check if the status is something wrong.

pacifica.cli.upload.**determine_sizes**(*md_update*, *args*)
Return the sizes of the tar, tar_in_tar and overall content length.

pacifica.cli.upload.**fake_uploader**(*rfd*, *content_length*)
Fake the upload by reading all the content then returning.

pacifica.cli.upload.**generate_names_from_dir**(*dirpath*, *followlinks*)
Generate a file list from a dirpath.

pacifica.cli.upload.**get_size_of_tar**(*md_update*, *args*)
Get the size of the tar file, no tarintar.

pacifica.cli.upload.**get_size_of_tar_in_tar**(*md_update*, *args*, *tar_size*)
Return the content-length for the upload.

pacifica.cli.upload.**invoke_uploader**(*md_update*, *rfd*, *content_length*)
Invoke the uploader code to actually upload.

pacifica.cli.upload.**perform_upload**(*md_update*, *args*, *content_length*, *tar_size*)
Setup threads and perform the upload.

pacifica.cli.upload.**pipefds**()
Setup pipe but return file objects instead.

pacifica.cli.upload.**save_local**(*rfd*, *wfd*, *save_filename*, *compressor*)
Save the bytes from rfd to args.savelocal and wfd.

pacifica.cli.upload.**setup_bundler**(*wfd*, *md_update*, *args*, *wthreads*)
Setup the bundler or local retry if passed.

pacifica.cli.upload.**setup_chain_thread**(*pipes*, *args*, *func*, *wthreads*, *doit*)
Setup a local thread if we doit using func.

pacifica.cli.upload.**tar_in_tar**(*rfd*, *wfd*, *md_update*, *bundle_size*)
Generate another bundler and wrap rfd in that tar.

pacifica.cli.upload.**upload_files_from_args**(*file_list*, *followlinks*, *prefix*)
Generate a files structure required by bundler.

pacifica.cli.upload.**upload_main**(*md_update*, *args*)
Main upload method.

pacifica.cli.upload.**wait_for_upload**(*args*, *jobid*, *up_obj*)
Wait (or not) for the jobid to complete the ingest process.

5.5 Utils Module

Utilities module for common methods.

pacifica.cli.utils.**compressor_generator**(*compressor_type*)
Return a compressor based on type, bzip2, gzip.

pacifica.cli.utils.**system_config_path**(*config_file*)
Return the system configuration path.

pacifica.cli.utils.**user_config_path**(*config_file*)
Return the global configuration path.

The main cli module describes the high level interface (the CLI).

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

p

Index

T

U

V

W