
ctyparser

Release 2.2.0

Mar 20, 2022

Contents

1	Installation	3
2	License	5
3	API	7
4	CLI Usage	9
	Python Module Index	11
	Index	13

A CTY.DAT parser for modern amateur radio programs.

CHAPTER 1

Installation

ctyparser requires Python 3.6 at minimum. Install by running:

```
$ pip install ctyarser
```


CHAPTER 2

License

Copyright 2019-2022 classabbyamp, 0x5c

Released under the MIT License. See `LICENSE` for the full license text.

```

class ctyparser.BigCty (file_path: Union[str, os.PathLike, None] = None)
    Class representing a BigCTY dataset. Can be initialised with data by passing the path to a valid cty.json file
    to the constructor.

        Parameters file_path (str or os.PathLike, optional) – Location of the cty.json
        file to load.

        Variables version (str) – the datestamp of the data, YYYYMMDD format.

dump (cty_file: Union[str, os.PathLike]) → None
    Dumps the data of the instance to a cty.json file.

        Parameters cty_file (str or os.PathLike) – Path to the file to dump to.

        Returns None

formatted_version
    Formatted representation of the version/date of the current BigCTY data.

        Getter Returns version in YYYY-MM-DD format, or 0000-00-00 (if invalid date)

        Type str

import_dat (dat_file: Union[str, os.PathLike]) → None
    Imports CTY data from a CTY.DAT file.

        Parameters dat_file (str or os.PathLike) – Path to the file to import.

        Returns None

load (cty_file: Union[str, os.PathLike]) → None
    Loads a cty.json file into the instance.

        Parameters cty_file (str or os.PathLike) – Path to the file to load.

        Returns None

update () → bool
    Updates the instance's data from the feed.

```

Raises `Exception` – If there is no date in the feed.

Returns `True` if an update was done, otherwise `False`.

Return type `bool`

version

The version/date of the current BigCTY data.

Getter Returns version in YYYYMMDD format

Type `str`

CHAPTER 4

CLI Usage

Note: CLI is a work in progress!

Currently, only updating/creating a file named `cty.json` in the current working directory is supported:

```
$ python3 -m ctyparser
```


C

`ctyparser`, [7](#)

B

`BigCty` (*class in ctyparser*), 7

C

`ctyparser` (*module*), 7

D

`dump()` (*ctyparser.BigCty method*), 7

F

`formatted_version` (*ctyparser.BigCty attribute*), 7

I

`import_dat()` (*ctyparser.BigCty method*), 7

L

`load()` (*ctyparser.BigCty method*), 7

U

`update()` (*ctyparser.BigCty method*), 7

V

`version` (*ctyparser.BigCty attribute*), 8