
mmcif_pdbx

Release 1.1.1

Jul 18, 2020

Contents

1	Origin of this software	3
2	Versions	5
3	Installation	7
4	Testing	9
5	Indices and tables	11
6	Contents	13
6.1	API Reference	13
6.1.1	Input/Output functions	13
6.1.2	Input/Output data structures	14
6.1.3	Error classes	15
	Python Module Index	17
	Index	19

Release 1.1.1

Date Jul 18, 2020

This is yet another PyPI package for <http://mmcif.wwpdb.org/pdbx-mmcif-home-page.html>. It emphasizes a simple and pure Python interface to basic mmCIF functionality.

The canonical mmCIF Python package can be found at <https://github.com/rcsb/py-mmcif>. It is full-featured and includes C/C++ code to accelerate I/O functions.

This package provides the module *pdbx*. More information about the *pdbx* module can be found in the *API Reference*. section.

CHAPTER 1

Origin of this software

All of the code in this repository is based on <http://mmcif.wwpdb.org/>. Specifically, this code is directly derived from <http://mmcif.wwpdb.org/docs/sw-examples/python/src/pdbx.tar.gz> linked from <http://mmcif.wwpdb.org/docs/sw-examples/python/html/>.

See <http://mmcif.wwpdb.org/docs/sw-examples/python/html/> for more information about this package, including examples.

CHAPTER 2

Versions

Versions 0.* maintain API compatibility with the original code. Subsequent versions break that compatibility, primarily by renaming methods in compliance with PEP8.

CHAPTER 3

Installation

This python package can be installed via [setuptools](#), `pip install .`, or via [PyPI](#).

CHAPTER 4

Testing

The software can be tested with `pytest` by running:

```
python -m pytest
```

from the top-level directory.

CHAPTER 5

Indices and tables

- `genindex`
- `modindex`
- `search`

6.1 API Reference

The API of *pdbx* is documented here for developers to use its mmCIF/PDBx functionality.

Note: The API is still changing and there is currently no guarantee that it will remain stable between minor releases.

6.1.1 Input/Output functions

<i>load</i> (fp)	Parse a CIF file into a list of DataContainer objects
<i>loads</i> (s)	Parse a CIF string into a list of DataContainer objects
<i>dump</i> (datacontainers, fp)	Write a list of DataContainer objects to a CIF file
<i>dumps</i> (datacontainers)	Serialize a list of DataContainer objects to a CIF formatted string

pdbx.load

`pdbx.load(fp) → list`
 Parse a CIF file into a list of DataContainer objects

pdbx.loads

`pdbx.loads(s: str) → list`
 Parse a CIF string into a list of DataContainer objects

pdbx.dump

`pdbx.dump` (*datacontainers: list, fp*)
Write a list of DataContainer objects to a CIF file

pdbx.dumps

`pdbx.dumps` (*datacontainers: list*) → str
Serialize a list of DataContainer objects to a CIF formatted string

6.1.2 Input/Output data structures

<i>containers</i>	A collection of container classes supporting the PDBx/mmCIF storage model.
<i>reader</i>	PDBx/mmCIF dictionary and data file parser.
<i>writer</i>	Classes for writing data and dictionary containers in PDBx/mmCIF format.

pdbx.containers

A collection of container classes supporting the PDBx/mmCIF storage model.

A base container class is defined which supports common features of data and definition containers. PDBx data files are organized in sections called data blocks which are mapped to data containers. PDBx dictionaries contain definition sections and data sections which are mapped to definition and data containers respectively.

Data in both PDBx data files and dictionaries are organized in data categories. In the PDBx syntax individual items or data identified by labels of the form ‘_categoryName.attribute_name’. The terms category and attribute in PDBx jargon are analogous table and column in relational data model, or class and attribute in an object oriented data model.

The DataCategory class provides base storage container for instance data and definition meta data.

Classes

<code>CifName</code>	Class of utilities for CIF-style data names.
<code>ContainerBase(name)</code>	Container base class for data and definition objects.
<code>DataCategory(name[, attribute_name_list, ...])</code>	Methods for creating, accessing, formatting PDBx cif data categories.
<code>DataCategoryBase(name[, ...])</code>	Base object definition for a data category.
<code>DataContainer(name)</code>	Container class for DataCategory objects.
<code>DefinitionContainer(name)</code>	Container for definitions.

pdbx.reader

PDBx/mmCIF dictionary and data file parser.

Acknowledgements:

The tokenizer used in this module is modeled after the clever parser design used in the PyMMLIB package.

PyMMLib Development Group Authors: Ethan Merritt: merritt@u.washington.ed & Jay Painter: jay.painter@gmail.com See: <http://pymmllib.sourceforge.net/>

Classes

<code>PdbxReader(input_file)</code>	PDBx reader for data files and dictionaries.
-------------------------------------	--

pdbx.writer

Classes for writing data and dictionary containers in PDBx/mmCIF format.

Classes

<code>PdbxWriter([output_file])</code>	Write PDBx data files or dictionaries.
--	--

6.1.3 Error classes

<code>errors</code>	Error classes for PDBx/mmCIF.
---------------------	-------------------------------

pdbx.errors

Error classes for PDBx/mmCIF.

Exceptions

<code>PdbxError</code>	Class for catching general errors.
<code>PdbxSyntaxError(line_number, text)</code>	Class for catching syntax errors.

p

- `pdbx`, [13](#)
- `pdbx.containers`, [14](#)
- `pdbx.errors`, [15](#)
- `pdbx.reader`, [14](#)
- `pdbx.writer`, [15](#)

D

`dump()` (*in module pdbx*), 14
`dumps()` (*in module pdbx*), 14

L

`load()` (*in module pdbx*), 13
`loads()` (*in module pdbx*), 13

P

`pdbx` (*module*), 13
`pdbx.containers` (*module*), 14
`pdbx.errors` (*module*), 15
`pdbx.reader` (*module*), 14
`pdbx.writer` (*module*), 15