
CycloneDX Python

Release 3.11.7

Paul Horton, Jan Kowalleck, Steve Springett, Patrick Dwyer

Nov 03, 2023

CONTENTS:

1	Installation	3
2	Usage	5
2.1	Command Line Usage	5
2.2	Programmatic Usage	7
2.3	Parser Schema Support	9
3	Support	11
4	Python Version Support	13
5	Changelog	15
5.1	v3.11.7 (2023-11-03)	15
5.2	v3.11.6 (2023-11-03)	15
5.3	v3.11.5 (2023-10-20)	15
5.4	v3.11.4 (2023-10-19)	15
5.5	v3.11.3 (2023-10-19)	16
5.6	v3.11.2 (2023-07-12)	16
5.7	v3.11.1 (2023-07-12)	16
5.8	v3.11.0 (2023-02-11)	16
5.9	v3.10.1 (2022-12-15)	17
5.10	v3.10.0 (2022-12-13)	17
5.11	v3.9.0 (2022-12-13)	17
5.12	v3.8.0 (2022-12-12)	17
5.13	v3.7.4 (2022-12-12)	18
5.14	v3.7.3 (2022-12-11)	18
5.15	v3.7.2 (2022-11-15)	18
5.16	v3.7.1 (2022-11-10)	18
5.17	v3.7.0 (2022-11-10)	18
5.18	v3.6.4 (2022-11-10)	19
5.19	v3.6.3 (2022-09-19)	19
5.20	v3.6.2 (2022-09-19)	19
5.21	v3.6.1 (2022-09-19)	19
5.22	v3.6.0 (2022-09-16)	19
5.23	v3.5.0 (2022-06-27)	20
5.24	v3.4.0 (2022-06-16)	20
5.25	v3.3.0 (2022-06-16)	20
5.26	v3.2.2 (2022-06-02)	20
5.27	v3.2.1 (2022-04-05)	21
5.28	v3.2.0 (2022-04-05)	21

5.29	v3.1.1 (2022-03-21)	21
5.30	v3.1.0 (2022-03-10)	22
5.31	v3.0.0 (2022-02-21)	22
5.32	v2.0.3 (2022-02-03)	23
5.33	v2.0.2 (2022-02-03)	23
5.34	v2.0.1 (2022-01-24)	23
5.35	v2.0.0 (2022-01-13)	23
5.36	v1.5.3 (2021-11-23)	23
5.37	v1.5.2 (2021-11-23)	24
5.38	v1.5.1 (2021-11-23)	24
5.39	v1.5.0 (2021-11-17)	24
5.40	v1.4.3 (2021-11-16)	24
5.41	v1.4.2 (2021-11-12)	24
5.42	v1.4.1 (2021-10-26)	25
5.43	v1.4.0 (2021-10-21)	25
5.44	v1.3.1 (2021-10-19)	25
5.45	v1.3.0 (2021-10-19)	25
5.46	v1.2.0 (2021-10-12)	26
5.47	v1.1.0 (2021-10-04)	26
5.48	v1.0.5 (2021-09-27)	26
5.49	v1.0.4 (2021-09-27)	26
5.50	v1.0.3 (2021-09-27)	26
5.51	v1.0.2 (2021-09-13)	27
5.52	v1.0.1 (2021-09-13)	27
6	API Reference	29
6.1	cyclonedx_py	29
	Python Module Index	39
	Index	41

This project provides a runnable Python-based application for generating [CycloneDX](#) bill-of-material documents from either:

- Your current Python Environment
- Your project's manifest (e.g. `Pipfile.lock`, `poetry.lock` or `requirements.txt`)
- Conda as a Package Manager

The SBOM will contain an aggregate of all your current project's dependencies, or those defined by the manifest you supply.

[CycloneDX](#) is a lightweight BOM specification that is easily created, human-readable, and simple to parse.

INSTALLATION

Install from pypi.org as you would any other Python module using your preferred package manager:

```
pip install cyclonedx-bom
```


2.1 Command Line Usage

Once installed, you can call the tool via the following methods:

```
$ python3 -m cyclonedx_py
$ cyclonedx-py
$ cyclonedx-bom # deprecated
```

The full documentation can be issued by running with `--help`:

```
$ cyclonedx-py --help
usage: cyclonedx-py [-h] (-c | -cj | -e | -p | -pip | -r) [-i FILE_PATH]
                  [--format {json,xml}] [--schema-version {1.4,1.3,1.2,1.1,1.0}]
                  [-o FILE_PATH] [-F] [-X]
```

CycloneDX SBOM Generator

optional arguments:

<code>-h, --help</code>	show this help message and exit
<code>-c, --conda</code>	Build a SBOM based on the output from <code>`conda list --explicit`</code> or <code>`conda list --explicit --md5`</code>
<code>-cj, --conda-json</code>	Build a SBOM based on the output from <code>`conda list --json`</code>
<code>-e, --e, --environment</code>	Build a SBOM based on the packages installed in your current Python environment (default)
<code>-p, --p, --poetry</code>	Build a SBOM based on a Poetry <code>poetry.lock`</code> 's contents. Use with <code>-i</code> to specify absolute path to a <code>`poetry.lock`</code> you wish to use, else we'll look for one in the current working directory.
<code>-pip, --pip</code>	Build a SBOM based on a PipEnv <code>Pipfile.lock`</code> 's contents. Use with <code>-i</code> to specify absolute path to a <code>`Pipfile.lock`</code> you wish to use, else we'll look for one in the current working directory.
<code>-r, --r, --requirements</code>	Build a SBOM based on a <code>requirements.txt`</code> 's contents. Use with <code>-i</code> to specify absolute path to a <code>`requirements.txt`</code> you wish to use, else we'll look for one in the current working directory.
<code>-X</code>	Enable debug output

(continues on next page)

Input Method:

Flags to determine how this tool obtains its input

```
-i FILE_PATH, --in-file FILE_PATH
                        File to read input from, or STDIN if not specified
```

SBOM Output Configuration:

Choose the output format and schema version

```
--format {json,xml}  The output format for your SBOM (default: xml)
--schema-version {1.4,1.3,1.2,1.1,1.0}
                    The CycloneDX schema version for your SBOM (default:
                    1.3)
-o FILE_PATH, --o FILE_PATH, --output FILE_PATH
                    Output file path for your SBOM (set to '-' to output
                    to STDOUT)
-F, --force          If outputting to a file and the stated file already
                    exists, it will be overwritten.
```

2.1.1 From your current Python Environment

This will produce the most accurate and complete CycloneDX BOM as it will include all transitive dependencies required by the packages defined in your project's manifest (think `requirements.txt`).

When using *Environment* as the source, any license information available from the installed packages will also be included in the generated CycloneDX BOM.

Simply run:

```
cyclonedx-py -e -o -
```

This will generate a CycloneDX including all packages installed in your current Python environment and output to STDOUT in XML using the default schema version 1.3 by default.

From your Python application manifest

Note: Manifest scanning limits the amount of information available. Each manifest type contains different information but all are significantly less complete than scanning your actual Python Environment.

Conda

We support parsing output from Conda in various formats:

- Explicit output (run `conda list --explicit` or `conda list --explicit --md5`)
- JSON output (run `conda list --json`)

As example:

```
conda list --explicit --md5 | cyclonedx-py -c -o cyclonedx.xml
```

Poetry

We support parsing your `poetry.lock` file which should be committed along with your `pyproject.toml` and details exact pinned versions.

You can then run `cyclonedx-py` as follows:

```
cyclonedx-py -p -i PATH/TO/poetry.lock -o sbom.xml
```

If your `poetry.lock` is in the current working directory, you can also shorten this to:

```
cyclonedx-py -p -o sbom.xml
```

Pip

We currently support `Pipfile.lock` manifest files.

You can then run `cyclonedx-bom` as follows:

```
cyclonedx-py -pip -i PATH/TO/Pipfile.lock -o sbom.xml
```

If your `Pipfile.lock` is in the current working directory, you can also shorten this to:

```
cyclonedx-py -pip -o sbom.xml
```

Requirements

We support `requirements.txt` manifest files. Note that a SBOM such as CycloneDX expects exact version numbers, therefore if you wish to generate a BOM from a `requirements.txt`, these must be frozen. This can be accomplished via:

```
pip freeze > requirements.txt
```

You can then run `cyclonedx-bom` as follows:

```
cyclonedx-py -r -i PATH/TO/requirements.txt -o sbom.xml
```

If your `requirements.txt` is in the current working directory, you can also shorten this to:

```
cyclonedx-py -r -o sbom.xml
```

This will generate a CycloneDX and output to STDOUT in XML using the default schema version *1.3*.

Note: If you failed to freeze your dependencies before passing the `requirements.txt` data to `cyclonedx-bom`, you'll be warned about this and the dependencies that do not have pinned versions WILL NOT be included in the resulting CycloneDX output.

2.2 Programmatic Usage

This library provides a number of concrete implementations of `cyclonedx.parser.BaserParser`. Parsers are provided as a quick way to generate a BOM for Python applications or from Python environments.

WARNING: Limited information will be available when generating a BOM using some Parsers due to limited information kept/supplied by those package managers. See below for details of what fields can be completed by different Parsers.

Easily create your parser instance as follows:

```
from cyclonedx_py.parser.environment import EnvironmentParser

parser = EnvironmentParser()
```

2.2.1 Conda

- `cyclonedx_py.parser.conda.CondaListJsonParser`: Parses input provided as a `str` that is output from `conda list --json`
- `cyclonedx_py.parser.conda.CondaListExplicitParser`: Parses input provided as a `str` that is output from: `conda list --explicit` or `conda list --explicit --md5`

2.2.2 Environment

- `cyclonedx_py.parser.environment.EnvironmentParser`: Looks at the packages installed in your current Python environment

2.2.3 Pip

- `cyclonedx_py.parser.pipenv.PipEnvParser`: Parses `Pipfile.lock` content passed in as a string
- `cyclonedx_py.parser.pipenv.PipEnvFileParser`: Parses the `Pipfile.lock` file at the supplied path

2.2.4 Poetry

- `cyclonedx.parser.poetry.PoetryParser`: Parses `poetry.lock` content passed in as a string
- `cyclonedx.parser.poetry.PoetryFileParser`: Parses the `poetry.lock` file at the supplied path

2.2.5 Requirements

- `cyclonedx_py.parser.requirements.RequirementsParser`: Parses a multiline string that you provide that conforms to the `requirements.txt` **PEP 508** standard.
- `cyclonedx_py.parser.requirements.RequirementsFileParser`: Parses a file that you provide the path to that conforms to the `requirements.txt` **PEP 508** standard. It supports nested files, so if there is a line in your `requirements.txt` file with the `-r requirements-nested.txt` syntax, it'll parse the nested file as part of the same file.

CycloneDX software bill-of-materials require pinned versions of requirements. If your `requirements.txt` does not have pinned versions, warnings will be recorded and the dependencies without pinned versions will be excluded from the generated CycloneDX. CycloneDX schemas (from version 1.0+) require a component to have a version when included in a CycloneDX bill of materials (according to schema).

If you need to use a `requirements.txt` in your project that does not have pinned versions an acceptable workaround might be to:

```
pip install -r requirements.txt
pip freeze > requirements-frozen.txt
```

You can then feed in the frozen requirements from `requirements-frozen.txt` or use the `Environment` parser once you have installed your dependencies.

2.3 Parser Schema Support

Different parsers support population of different information about your dependencies due to how information is obtained and limitations of what information is available to each Parser. The sections below explain coverage as to what information is obtained by each set of Parsers. It does NOT guarantee the information is output in the resulting CycloneDX BOM document.

The below tables do not state whether specific schema versions support the attributes/items, just whether this library does.

xPath is used to refer to data attributes according to the [Cyclone DX Specification](#).

2.3.1 bom.components.component

Data Path	Parser Support				
	Conda	Environment	Pip	Poetry	Requirements
.supplier	N	N - Note 1	N/A	N/A	N/A
.author	N	Y - Note 1	N/A	N/A	N/A
.publisher	N	N - Note 1	N/A	N/A	N/A
.group	N	N	N	N	N
.name	Y	Y	Y	Y	Y
.version	Y	Y	Y	Y	Y
.description	N	N	N/A	N	N/A
.scope	N	N	N/A	N	N/A
.hashes	Y - Note 2	N/A	Y - Note 3	Y - Note 3	N/A
.licenses	N	Y - Note 1	N/A	N/A	N/A
.copyright	N	N - Note 1	N/A	N/A	N/A
.cpe	N/A	N/A	N/A	N/A	N/A
.purl	Y	Y	Y	Y	Y
.swid	N/A	N/A	N/A	N/A	N/A
.modified	<i>Deprecated - not used</i>				
.pedigree	N/A	N/A	N/A	N/A	N/A
.externalReferences	Y - Note 3	N/A	Y - Note 1	Y - Note 1	N/A
.properties	N/A	N/A	N/A	N/A	N/A
.components	N/A	N/A	N/A	N/A	N/A
.evidence	N/A	N/A	N/A	N/A	N/A
.releaseNotes	N/A	N/A	N/A	N/A	N/A

Legend:

- **Y**: YES with any caveat states.
- **N**: Not currently supported, but support believed to be possible.
- **N/A**: Not supported and not deemed possible (i.e. the Parser would never be able to reliably determine this info).

Notes

1. If contained in the packages METADATA
2. MD5 hashes are available when using the `CondaListExplicitParser` with output from the conda command `conda list --explicit --md5` only
3. Python packages are regularly available as both `.whl` and `.tar.gz` packages. This means for that for a given package and version multiple artefacts are possible - which would mean multiple hashes are possible. CycloneDX

supports only a single set of hashes identifying a single artefact at `component.hashes`. To cater for this situation in Python, we add the hashes to `component.externalReferences`, as we cannot determine which package was actually obtained and installed to meet a given dependency.

SUPPORT

If you run into issues utilising this library, please raise a [GitHub Issue](#). When raising an issue please include as much detail as possible including:

- Version cyclonedx-bom you have installed
- Input(s)
- Expected Output(s)
- Actual Output(s)

PYTHON VERSION SUPPORT

We endeavour to support all functionality for all [current actively supported Python versions](#). However, some features may not be possible/present in older Python versions due to their lack of support - which are noted below.

CHANGELOG

5.1 v3.11.7 (2023-11-03)

5.1.1 Fix

- Toml-compatible fingers-crossed handling for failed input data decoding (#613) (``fb3d7bf`` <<https://github.com/CycloneDX/cyclonedx-python/commit/fb3d7bfd1216ad8b5328a1d348fea04fee31d3a4>>`_`)

5.2 v3.11.6 (2023-11-03)

5.2.1 Fix

- Added a fingers-crossed handling for failed input data decoding (#612) (``be55902`` <<https://github.com/CycloneDX/cyclonedx-python/commit/be559020e482795c6603f36e98713c6f7bde1e34>>`_`)

5.3 v3.11.5 (2023-10-20)

5.3.1 Fix

- Custom input encoding (#601) (``363934c`` <<https://github.com/CycloneDX/cyclonedx-python/commit/363934c0bc69ebbb23472f1173bf3c6b1e3c023a>>`_`)

5.4 v3.11.4 (2023-10-19)

5.4.1 Fix

- Input file encoding fallback (``0bc7296`` <<https://github.com/CycloneDX/cyclonedx-python/commit/0bc72964d0578f713f405bc101742ef096bf8fd7>>`_`)

5.5 v3.11.3 (2023-10-19)

5.5.1 Fix

- Input file encoding (#596) (``a9dda4b` <https://github.com/CycloneDX/cyclonedx-python/commit/a9dda4bfd0e68529628eab99b6db00fb5214bfc3>`_)`

5.5.2 Documentation

- Publish coverage (#600) (``bd4f48e` <https://github.com/CycloneDX/cyclonedx-python/commit/bd4f48ef7f3c4c890a138c45dbc87f6ca3e2cf7b>`_)`
- Adjust syntax highlight for code blocks (#592) (``ccac31e` <https://github.com/CycloneDX/cyclonedx-python/commit/ccac31eb4d0996236da24ca9efb57af66bd1a020>`_)`
- Mark ShellSession in README (``411cf3d` <https://github.com/CycloneDX/cyclonedx-python/commit/411cf3d0a4b5005c1591211ecdc464d4747d69f1>`_)`

5.6 v3.11.2 (2023-07-12)

5.6.1 Fix

- Referenced branch `main`, instead of `master` (#562) (``830d15c` <https://github.com/CycloneDX/cyclonedx-python/commit/830d15c27fad475fa9a15918b1d5930cd71834d>`_)`

5.7 v3.11.1 (2023-07-12)

5.7.1 Fix

- Fix typo in help page (#552) (``19bf41a` <https://github.com/CycloneDX/cyclonedx-python/commit/19bf41a52a698ee3ddee5fafc5d293ea3d9427be>`_)`

5.8 v3.11.0 (2023-02-11)

5.8.1 Feature

- Deprecated CLI command `cyclonedx-bom` prints deprecation warning on `STDERR` before execution (#489) (``2009236` <https://github.com/CycloneDX/cyclonedx-python/commit/2009236c537af212aab1d5907e02f2b003f3062c>`_)`

5.8.2 Documentation

- Fix typo in CLI help page (#490) (``a8a8445`` <<https://github.com/CycloneDX/cyclonedx-python/commit/a8a844504494d10c217ba4739e6ff09b4ca34f67>>`_`)
- Fix typos (#482) (``edbe3d4`` <<https://github.com/CycloneDX/cyclonedx-python/commit/edbe3d4e0ee62396ac10b42dd9ee5d6094817675>>`_`)
- Fix shields (#473) (``e32b288`` <<https://github.com/CycloneDX/cyclonedx-python/commit/e32b28894a8859925f22a1f45aec8608e7cd8bc3>>`_`)

5.9 v3.10.1 (2022-12-15)

5.9.1 Fix

- PURL for PyPI packages from 'conda list' have the correct format now (#471) (``1573064`` <<https://github.com/CycloneDX/cyclonedx-python/commit/157306483a21583d752714a77ad7d0c7395291e5>>`_`)

5.9.2 Documentation

- Improve CONTRIBUTION instructions - sign-off step (#470) (``578c0a8`` <<https://github.com/CycloneDX/cyclonedx-python/commit/578c0a88e63c804b1462e3d3b617f56b53b6012e>>`_`)

5.10 v3.10.0 (2022-12-13)

5.10.1 Feature

- Add support for poetry lock format v2.0 (#469) (``0b1e07f`` <<https://github.com/CycloneDX/cyclonedx-python/commit/0b1e07f91aada201088605a84ea394182ce0f10e>>`_`)

5.11 v3.9.0 (2022-12-13)

5.11.1 Feature

- Parsers can output more debug messages (#466) (``9eedb4f`` <<https://github.com/CycloneDX/cyclonedx-python/commit/9eedb4f27bb81f4ad323e9fa0f79230b0710032>>`_`)

5.12 v3.8.0 (2022-12-12)

5.12.1 Feature

- Error- and debug-output is send to STDERR, instead of STDOUT (#465) (``f543b69`` <<https://github.com/CycloneDX/cyclonedx-python/commit/f543b69ee4463df3fb4d4b7c86475562f62e4744>>`_`)

5.13 v3.7.4 (2022-12-12)

5.13.1 Fix

- Ignore broken licenses in env parser (#463) (``3118acd`` <<https://github.com/CycloneDX/cyclonedx-python/commit/3118acdf180b6d8d35a637b3e94dc6ec7c5c5b3d>>`_)

5.14 v3.7.3 (2022-12-11)

5.14.1 Fix

- Adjust dependency `pip-requirements-parser` to a working version (#450) (``6101986`` <<https://github.com/CycloneDX/cyclonedx-python/commit/610198659be408b5ef17d649aa381944d992a7dd>>`_)

5.15 v3.7.2 (2022-11-15)

5.15.1 Fix

- Add a missing space in the help pages `path to -> path to` (#443) (``bc5fe57`` <<https://github.com/CycloneDX/cyclonedx-python/commit/bc5fe5760565e608387ad7638126869550d65213>>`_)

5.16 v3.7.1 (2022-11-10)

5.16.1 Fix

- **EnvironmentParser:** Reduced crashes if no Classifiers are found (#441) (``67f56e7`` <<https://github.com/CycloneDX/cyclonedx-python/commit/67f56e7bfa4fb9d50654ebd07ece1ad14377a355>>`_)

5.17 v3.7.0 (2022-11-10)

5.17.1 Feature

- Pass `purl-bom-ref` to EnvironmentParser (#432) (``7cfefeb`` <<https://github.com/CycloneDX/cyclonedx-python/commit/7cfefeb389b3c63b69ad93aeca1a709231da2901>>`_)

5.18 v3.6.4 (2022-11-10)

5.18.1 Fix

- **EnvironmentParser:** Remove code break when classifier parsing in py>=3.8 (#431) (``4ab075e` <https://github.com/CycloneDX/cyclonedx-python/commit/4ab075ee814571a8dc8c1e7b962686b232619330>`_)`)

5.19 v3.6.3 (2022-09-19)

5.19.1 Fix

- CI release pipeline (``99ccdc6` <https://github.com/CycloneDX/cyclonedx-python/commit/99ccdc671f5a7a941f31199813bce71405bbfdd8>`_)`)

5.20 v3.6.2 (2022-09-19)

5.20.1 Fix

- CI release pipeline (``6515071` <https://github.com/CycloneDX/cyclonedx-python/commit/6515071fc95d2b460577d0fbceb7d6c34a18c508>`_)`)

5.21 v3.6.1 (2022-09-19)

5.21.1 Fix

- Properly declare licenses from environment (#417) (``25f9e29` <https://github.com/CycloneDX/cyclonedx-python/commit/25f9e29a162f20918b6f1bbe887cc7b18c623c16>`_)`)

5.22 v3.6.0 (2022-09-16)

5.22.1 Feature

- Enable dependency `cyclonedx-python-lib@^3` (#418) (``05cd51e` <https://github.com/CycloneDX/cyclonedx-python/commit/05cd51e1da261d29fb5c3e1722544a8f00a0cfd>`_)`)

5.22.2 Documentation

- Describe `cyclonedx-py` rather than `cyclonedx-bom` (``c04196e` <https://github.com/CycloneDX/cyclonedx-python/commit/c04196e4404efc0513676e5bafeaf03e6b3b8e3>`_`)`
- Minor updates to poetry usage details & `contributing.md` (#407) (``0abe230` <https://github.com/CycloneDX/cyclonedx-python/commit/0abe23049b5423f55b3e0951a00047f4e3f93056>`_`)`

5.23 v3.5.0 (2022-06-27)

5.23.1 Feature

- Optionally force `bom_ref` to be `purl` rather than the default random UUID format - thanks @RodneyRichardson (``9659d08` <https://github.com/CycloneDX/cyclonedx-python/commit/9659d08f524fd1ea2eb34234f2449105feb93f62>`_`)`

5.24 v3.4.0 (2022-06-16)

5.24.1 Feature

- Update `purl` to match specification when ingesting packages from Conda - thanks to @RodneyRichardson (``072c8f1` <https://github.com/CycloneDX/cyclonedx-python/commit/072c8f11bdef44abb0c6f7f7e99e2b833ab1c875>`_`)`

5.25 v3.3.0 (2022-06-16)

5.25.1 Feature

- Add Conda MD5 hash to `Component.hashes`, if available - thanks @RodneyRichardson (``772c517` <https://github.com/CycloneDX/cyclonedx-python/commit/772c517521da0fd8ddbd1ed8abdf22243f418217>`_`)`

5.26 v3.2.2 (2022-06-02)

5.26.1 Fix

- Add actively used (transitive) dependencies (#363) (``1f45ad9` <https://github.com/CycloneDX/cyclonedx-python/commit/1f45ad9162be511f07e9310414793218c554a097>`_`)`

5.27 v3.2.1 (2022-04-05)

5.27.1 Fix

- Cli default file for json format (``8747620` <https://github.com/CycloneDX/cyclonedx-python/commit/8747620dac7ed3eeff69369c05dfb6386a56e549>`_)`

5.28 v3.2.0 (2022-04-05)

5.28.1 Feature

- Make module callable (``5b3d8d7` <https://github.com/CycloneDX/cyclonedx-python/commit/5b3d8d7641b0f2825e5419b5ad8c8a75bf66403b>`_)`

5.28.2 Documentation

- Describe methods to call the tool (``2bac83a` <https://github.com/CycloneDX/cyclonedx-python/commit/2bac83a6c6f7354d8b7218c32b4b2e5d96b2fd0c>`_)`

5.29 v3.1.1 (2022-03-21)

5.29.1 Fix

- **conda-parser:** Version recognition for strings (#332) (``65246dd` <https://github.com/CycloneDX/cyclonedx-python/commit/65246ddfa9a55ce53fbf87f33b1f269c519f9b3a>`_)`

5.29.2 Documentation

- Add link to <https://cyclonedx.org/> to README (``fc4b8e4` <https://github.com/CycloneDX/cyclonedx-python/commit/fc4b8e44bec39b175bb8994e0a59bc5076d1b2a6>`_)`
- Add hint for RTFD to README (``cf4f534` <https://github.com/CycloneDX/cyclonedx-python/commit/cf4f534401dc90dbe093ce1a094efb02e5fb7c90>`_)`
- Add RTFD shield to README (``7fef6ee` <https://github.com/CycloneDX/cyclonedx-python/commit/7fef6eec5d553c7687e7b2d2af1ba4e330f16490>`_)`
- Fixed link to RTFD (``3a8669a` <https://github.com/CycloneDX/cyclonedx-python/commit/3a8669ad7ba4230d06d1e0965342a5a836a52d1f>`_)`

5.30 v3.1.0 (2022-03-10)

5.30.1 Feature

- Add pip-requirements-parser and update virtualenv to latest version (``73b2182` <https://github.com/CycloneDX/cyclonedx-python/commit/73b2182550d9635a0a5ab8e4f2226f37cf6b1b35>``)

5.30.2 Fix

- Sort imports (``fdec44b` <https://github.com/CycloneDX/cyclonedx-python/commit/fdec44bc111d7eb1add080a219dbc77744678f8a>``)
- Try to fix the temp file issue on Windows machines (``684d4f0` <https://github.com/CycloneDX/cyclonedx-python/commit/684d4f03ad6f8c0764dfaf8f3a38a09b91b69e5d>``)

5.30.3 Documentation

- Update RequirementsFileParser docs to include nested file support (``9e9021d` <https://github.com/CycloneDX/cyclonedx-python/commit/9e9021decb19d8262e87fe6955577c1bd1309d95>``)

5.31 v3.0.0 (2022-02-21)

5.31.1 Feature

- Bump to latest `cyclonedx-python-lib` (``5902fbf` <https://github.com/CycloneDX/cyclonedx-python/commit/5902fbf9dc5becdf7d92180242488e56b998d9de>``)
- Added marker and classifiers to denote this as typed (#313) (``f317353` <https://github.com/CycloneDX/cyclonedx-python/commit/f317353bd7a24dbf4fb31642d766d94da609eb42>``)
- Update to latest RC of `cyclonedx-python-lib` (``6c8b517` <https://github.com/CycloneDX/cyclonedx-python/commit/6c8b5173f07329b2086312d27af5d111f9b2c7ed>``)
- Update to latest RC of `cyclonedx-python-lib` (``bc8ee6b` <https://github.com/CycloneDX/cyclonedx-python/commit/bc8ee6bb115dd5214358430f64bd0581de5cb2e4>``)

5.31.2 Breaking

- Default Schema Version has been replaced by notion of LATEST supported Schema Version (``5902fbf` <https://github.com/CycloneDX/cyclonedx-python/commit/5902fbf9dc5becdf7d92180242488e56b998d9de>``)

5.32 v2.0.3 (2022-02-03)

5.32.1 Fix

- Docker image releae checkout ref w/o tags (#309) (``5d8b1e1` <https://github.com/CycloneDX/cyclonedx-python/commit/5d8b1e159c2ced59e810b9e9564e19a29fe263d0>`_`)`

5.33 v2.0.2 (2022-02-03)

5.33.1 Fix

- Properly support reading from stdin (#307) (``23f31a0` <https://github.com/CycloneDX/cyclonedx-python/commit/23f31a03a4fbf888f396b88a9413c054358b2a3a>`_`)`

5.34 v2.0.1 (2022-01-24)

5.34.1 Fix

- Bump dependencies to get latest `cyclonedx-python-lib` (``87c3fe7` <https://github.com/CycloneDX/cyclonedx-python/commit/87c3fe7747cd8abd55ad5699bfc87ad9877c8132>`_`)`

5.35 v2.0.0 (2022-01-13)

5.35.1 Feature

- Add support for CycloneDX 1.4 specification (#294) (``7bb6d32` <https://github.com/CycloneDX/cyclonedx-python/commit/7bb6d328adec59cdd4c3ab80eb5f39568ca3bc9c>`_`)`

5.35.2 Documentation

- Readme maintenance - shields & links (#266) (``a34046f` <https://github.com/CycloneDX/cyclonedx-python/commit/a34046f9b4c96d013fdf2dbdac5e930aa9204e15>`_`)`

5.36 v1.5.3 (2021-11-23)

5.36.1 Fix

- Revert to previous process for building Docker image as PyPi index update is too slow to pull straight away after publish (``67bb738` <https://github.com/CycloneDX/cyclonedx-python/commit/67bb738246bfe0ca3acd409d8c5a27fd7a305347>`_`)`

5.37 v1.5.2 (2021-11-23)

5.37.1 Fix

- Corrected docker image build process to not rely on dist folder which is cleaned up by python-semantic-release (``6c65c11` <https://github.com/CycloneDX/cyclonedx-python/commit/6c65c11d439169417e2ef7e94cacb1ec216eb11c>`_)`

5.38 v1.5.1 (2021-11-23)

5.38.1 Fix

- Re-enable build and publish of Docker Image (#263) (``478360d` <https://github.com/CycloneDX/cyclonedx-python/commit/478360db0de269159ab6e3777cd291b87e2e1174>`_)`

5.39 v1.5.0 (2021-11-17)

5.39.1 Feature

- Support for Python 3.10 (#261) (``f4f9ffe` <https://github.com/CycloneDX/cyclonedx-python/commit/f4f9ffe4b1e2d4ffe4ad0b274a067a20c9c372f>`_)`

5.40 v1.4.3 (2021-11-16)

5.40.1 Fix

- Add static code analysis, better typing and bump cyclonedx-python-lib to 0.11 (``d5d9f56` <https://github.com/CycloneDX/cyclonedx-python/commit/d5d9f563f2ceb1bdfb2f9cb39ff07af9f0deca26>`_)`

5.41 v1.4.2 (2021-11-12)

5.41.1 Fix

- If no input file is supplied and no input is provided on STDIN, we will now try to automatically locate (in the current working directory) a manifest with default name for the input type specified. This works for PIP (Pipfile.lock), Poetry (poetry.lock) and Requirements (requirements.txt) (``93f9e59` <https://github.com/CycloneDX/cyclonedx-python/commit/93f9e5985f0d0cecd865b66119276d33b2175fe9>`_)`

5.42 v1.4.1 (2021-10-26)

5.42.1 Fix

- Corrected documentation after deprecation of `-rf`, `-pf`, `--poetry-file`, `--requirements-file` and `--pip-file` (``4c4c8d8` <https://github.com/CycloneDX/cyclonedx-python/commit/4c4c8d8d4756ebc953c26504052d5469f3c47cfa>`_)`

5.43 v1.4.0 (2021-10-21)

5.43.1 Breaking Changes

- The following parameter flags have all been removed in favour of a single common parameter flag `-i`:
 - For Poetry: `-pf`, `--poetry-file`
 - For Requirements: `-rf`, `--requirements-file`

5.43.2 Feature

- Add conda support (bump cyclonedx-python-lib to ^0.10.0) (``cb24275` <https://github.com/CycloneDX/cyclonedx-python/commit/cb24275f3e8716244de2b4ef0a046b879fa88ba5>`_)`

5.43.3 Fix

- Encoding issues on Windows (bump cyclonedx-python-lib to ^0.10.2) (``da6772b` <https://github.com/CycloneDX/cyclonedx-python/commit/da6772be89ad923b1d8df6dd3b2a89c6e5805571>`_)`
- Encoding issues on Windows (bump cyclonedx-python-lib to ^0.10.1) (``fe5df36` <https://github.com/CycloneDX/cyclonedx-python/commit/fe5df3607157b2f24854ef1f69457f163d79a093>`_)`

5.44 v1.3.1 (2021-10-19)

5.44.1 Fix

- Bump to cyclonedx-python-lib to resolve issue #244 (``ebea3ef` <https://github.com/CycloneDX/cyclonedx-python/commit/ebea3ef47e917479a7474489bb274b5fa9704375>`_)`

5.45 v1.3.0 (2021-10-19)

5.45.1 Feature

- Add license information in CycloneDX BOM when using Environment as the source (``5d1f9a7` <https://github.com/CycloneDX/cyclonedx-python/commit/5d1f9a76cfa2bc1461a3dcf4c140d81876a37c40>`_)`

5.46 v1.2.0 (2021-10-12)

5.46.1 Feature

- Update to latest stable cyclonedx-python-lib (``6145bd5` <https://github.com/CycloneDX/cyclonedx-python/commit/6145bd52c450e66f42367e61e086d2a9d9818b47>`_`)

5.47 v1.1.0 (2021-10-04)

5.47.1 Feature

- Add support for generating SBOM from poetry.lock files (``bb4ac0f` <https://github.com/CycloneDX/cyclonedx-python/commit/bb4ac0f29b46db59b192191f65dfa40757268188>`_`)

5.48 v1.0.5 (2021-09-27)

5.48.1 Fix

- Handle `requirements.txt` which contain dependencies without a version statement and warn that they cannot be included in the resulting CycloneDX BOM (``e637e56` <https://github.com/CycloneDX/cyclonedx-python/commit/e637e56cada6d841dae193c106647b0b03a4e776>`_`)

5.49 v1.0.4 (2021-09-27)

5.49.1 Fix

- Error message when `requirements.txt` file is non-existent updated (``3bbc071` <https://github.com/CycloneDX/cyclonedx-python/commit/3bbc071a1ff26599bd9eb3220de38bd9c58fa294>`_`)

5.50 v1.0.3 (2021-09-27)

5.50.1 Fix

- Default to “`requirements.txt`” in current directory when “`-r`” flag is supplied but not “`-rf`” flag is supplied (``bb7e30a` <https://github.com/CycloneDX/cyclonedx-python/commit/bb7e30a869300b1e63a00d7db4bcc7f35d68552d>`_`)

5.51 v1.0.2 (2021-09-13)

5.51.1 Fix

- Release GH action (``148421b`` <<https://github.com/CycloneDX/cyclonedx-python/commit/148421bcd8cea2b5f8f3bd5958f6f7171afe859e>>`_`)

5.52 v1.0.1 (2021-09-13)

5.52.1 Fix

- **ci:** Corrected main to master branch. (``7162cd9`` <<https://github.com/CycloneDX/cyclonedx-python/commit/7162cd9385729dafbdc15dbb55e9ac5adf3906cf>>`_`)

API REFERENCE

This page contains auto-generated API reference documentation¹.

6.1 cyclonedx_py

6.1.1 Subpackages

`cyclonedx_py.exception`

Exceptions that are specific to the CycloneDX Python implementation.

Submodules

`cyclonedx_py.exception.parser`

Exceptions that are specific error scenarios during occurring within Parsers in the CycloneDX library implementation.

`cyclonedx_py.parser`

Set of concrete classes and methods which allow for quick creation of a Bom instance from your environment or Python project.

Use a Parser instead of programmatically creating a Bom as a developer.

Submodules

`cyclonedx_py.parser._debug`

The following structures are internal helpers.

¹ Created with sphinx-autoapi

Module Contents

Functions

<code>quiet</code> (→ None)	Do not print anything.
-----------------------------	------------------------

Attributes

<code>DebugMessageCallback</code>	Callback for debug messaging.
-----------------------------------	-------------------------------

`cyclonedx_py.parser._debug.DebugMessageCallback`

Callback for debug messaging.

Parameters

message – the format string.

Other Parameters

the `*args`: to `str.format()`.

Keyword Arguments

the `**kwargs` to `str.format()`.

`cyclonedx_py.parser._debug.quiet`(*message: str, *_: Any, **__: Any*) → None

Do not print anything.

Must be compatible to `DebugMessageCallback`.

`cyclonedx_py.parser.conda`

Module Contents

Classes

<code>_BaseCondaParser</code>	Internal abstract parser - not for programmatic use.
<code>CondaListJsonParser</code>	This parser is intended to receive the output from the command <code>conda list --json</code> .
<code>CondaListExplicitParser</code>	This parser is intended to receive the output from the command <code>conda list --explicit</code> or

```
class cyclonedx_py.parser.conda._BaseCondaParser(conda_data: str, use_purl_bom_ref: bool = False, *,
        debug_message: cy-
        clonedx_py.parser._debug.DebugMessageCallback
        = quiet)
```

Bases: `cyclonedx.parser.BaseParser`

Internal abstract parser - not for programmatic use.

abstract `_parse_to_conda_packages(data_str: str) → None`

Abstract method for implementation by concrete Conda Parsers.

Implementation should add a *list* of *CondaPackage* instances to *self._conda_packages*

Params:

data_str:

str data passed into the Parser

_conda_packages_to_components(*use_purl_bom_ref: bool*) → None

Converts the parsed *CondaPackage* instances into *Component* instances.

class `cyclonedx_py.parser.conda.CondaListJsonParser`(*conda_data: str, use_purl_bom_ref: bool = False, *, debug_message: cyclonedx_py.parser._debug.DebugMessageCallback = quiet*)

Bases: *_BaseCondaParser*

This parser is intended to receive the output from the command `conda list -json`.

_parse_to_conda_packages(*data_str: str*) → None

Abstract method for implementation by concrete Conda Parsers.

Implementation should add a *list* of *CondaPackage* instances to *self._conda_packages*

Params:

data_str:

str data passed into the Parser

class `cyclonedx_py.parser.conda.CondaListExplicitParser`(*conda_data: str, use_purl_bom_ref: bool = False, *, debug_message: cyclonedx_py.parser._debug.DebugMessageCallback = quiet*)

Bases: *_BaseCondaParser*

This parser is intended to receive the output from the command `conda list -explicit` or `conda list -explicit -md5`.

_parse_to_conda_packages(*data_str: str*) → None

Abstract method for implementation by concrete Conda Parsers.

Implementation should add a *list* of *CondaPackage* instances to *self._conda_packages*

Params:

data_str:

str data passed into the Parser

`cyclonedx_py.parser.environment`

Parser classes for reading installed packages in your current Python environment.

These parsers look at installed packages only - not what you have defined in any dependency tool - see the other Parsers if you want to derive CycloneDX from declared dependencies.

The Environment Parsers support population of the following data about Components:

Module Contents

Classes

<i>EnvironmentParser</i>	This will look at the current Python environment and list out all installed packages.
--------------------------	---

```
class cyclonedx_py.parser.environment.EnvironmentParser(use_purl_bom_ref: bool = False, *,
                                                       debug_message: cy-
                                                       clonedx_py.parser._debug.DebugMessageCallback
                                                       = quiet)
```

Bases: `cyclonedx.parser.BaseParser`

This will look at the current Python environment and list out all installed packages.

Best used when you have virtual Python environments per project.

```
static _get_metadata_for_package(package_name: str) → importlib.metadata.PackageMetadata
```

`cyclonedx_py.parser.pipenv`

Module Contents

Classes

<i>PipEnvParser</i>	Helper class that provides a standard way to create an ABC using
<i>PipEnvFileParser</i>	Helper class that provides a standard way to create an ABC using

```
class cyclonedx_py.parser.pipenv.PipEnvParser(pipenv_contents: str, use_purl_bom_ref: bool = False,
                                              *, debug_message:
                                              cyclonedx_py.parser._debug.DebugMessageCallback =
                                              quiet)
```

Bases: `cyclonedx.parser.BaseParser`

Helper class that provides a standard way to create an ABC using inheritance.

```
class cyclonedx_py.parser.pipenv.PipEnvFileParser(pipenv_lock_filename: str, use_purl_bom_ref: bool
                                                  = False, *, debug_message: cy-
                                                  clonedx_py.parser._debug.DebugMessageCallback
                                                  = quiet)
```

Bases: `PipEnvParser`

Helper class that provides a standard way to create an ABC using inheritance.

`cyclonedx_py.parser.poetry`

Module Contents

Classes

<i>PoetryParser</i>	Helper class that provides a standard way to create an ABC using
<i>PoetryFileParser</i>	Helper class that provides a standard way to create an ABC using

```
class cyclonedx_py.parser.poetry.PoetryParser(poetry_lock_contents: str, use_purl_bom_ref: bool =
False, *, debug_message:
cyclonedx_py.parser._debug.DebugMessageCallback =
quiet)
```

Bases: `cyclonedx.parser.BaseParser`

Helper class that provides a standard way to create an ABC using inheritance.

```
class cyclonedx_py.parser.poetry.PoetryFileParser(poetry_lock_filename: str, use_purl_bom_ref: bool
= False, *, debug_message: cy-
clonedx_py.parser._debug.DebugMessageCallback
= quiet)
```

Bases: *PoetryParser*

Helper class that provides a standard way to create an ABC using inheritance.

`cyclonedx_py.parser.requirements`

Module Contents

Classes

<i>RequirementsParser</i>	Helper class that provides a standard way to create an ABC using
<i>RequirementsFileParser</i>	Helper class that provides a standard way to create an ABC using

```
class cyclonedx_py.parser.requirements.RequirementsParser(requirements_content: str,
use_purl_bom_ref: bool = False, *,
debug_message: cy-
clonedx_py.parser._debug.DebugMessageCallback
= quiet)
```

Bases: `cyclonedx.parser.BaseParser`

Helper class that provides a standard way to create an ABC using inheritance.

```
class cyclonedx_py.parser.requirements.RequirementsFileParser(requirements_file: str,
                                                            use_purl_bom_ref: bool = False, *,
                                                            debug_message: cy-
                                                            clonedx_py.parser._debug.DebugMessageCallback
                                                            = quiet)
```

Bases: *RequirementsParser*

Helper class that provides a standard way to create an ABC using inheritance.

cyclonedx_py.utils

Set of utility classes.

Submodules

`cyclonedx_py.utils.conda`

Module Contents

Classes

<code>CondaPackage</code>	Internal package for unifying Conda package definitions to.
---------------------------	---

Functions

<code>conda_package_to_purl(→ ageurl.PackageURL)</code>	pack-	Return the purl for the specified package.
<code>_conda_package_to_pypi_purl(→ ageurl.PackageURL)</code>	pack-	Return the purl for a pip-installed package in a conda environment.
<code>parse_conda_json_to_conda_package(→ Optional[CondaPackage])</code>	Op-	
<code>parse_conda_list_str_to_conda_package(...)</code>		Helper method for parsing a line of output from <i>conda list --explicit</i> into our internal <i>CondaPackage</i> object.
<code>split_package_string(→ Tuple[str, str, str, str])</code>		Helper method for parsing package_name_version_build_string.
<code>split_package_build_string(→ Tuple[str, Optional[int]])</code>	Op-	Helper method for parsing build_string.

```
class cyclonedx_py.utils.conda.CondaPackage
```

Bases: TypedDict

Internal package for unifying Conda package definitions to.

base_url: str

build_number: int | None

```

build_string: str
channel: str
dist_name: str
name: str
platform: str
version: str
package_format: str | None
md5_hash: str | None

```

`cyclonedx_py.utils.conda.conda_package_to_purl(pkg: CondaPackage) → packageurl.PackageURL`

Return the purl for the specified package. See <https://github.com/package-url/purl-spec/blob/master/PURL-TYPES.rst#conda>

`cyclonedx_py.utils.conda._conda_package_to_pypi_purl(pkg: CondaPackage) → packageurl.PackageURL`

Return the purl for a pip-installed package in a conda environment. These packages are listed as if from the pseudo-channel “pypi”. See <https://github.com/package-url/purl-spec/blob/master/PURL-TYPES.rst#pypi>

`cyclonedx_py.utils.conda.parse_conda_json_to_conda_package(conda_json_str: str) → CondaPackage | None`

`cyclonedx_py.utils.conda.parse_conda_list_str_to_conda_package(conda_list_str: str) → CondaPackage | None`

Helper method for parsing a line of output from `conda list -explicit` into our internal `CondaPackage` object.

Params:

conda_list_str:
Line of output from `conda list -explicit`

Returns:

Instance of `CondaPackage` else `None`.

`cyclonedx_py.utils.conda.split_package_string(package_name_version_build_string: str) → Tuple[str, str, str]`

Helper method for parsing `package_name_version_build_string`.

Returns:

Tuple (package_name, build_version, build_string)

`cyclonedx_py.utils.conda.split_package_build_string(build_string: str) → Tuple[str, int | None]`

Helper method for parsing `build_string`.

Returns:

Tuple (build_string, build_number)

6.1.2 Submodules

`cyclonedx_py.__main__`

`cyclonedx_py.client`

Module Contents

Classes

<code>_CLI_OUTPUT_FORMAT</code>	Generic enumeration.
<code>CycloneDxCmd</code>	

Functions

<code>main</code> (→ None)
<code>main_deprecated</code> (→ None)

Attributes

<code>_output_formats</code>
<code>_output_default_filenames</code>

exception `cyclonedx_py.client.CycloneDxCmdException`

Bases: `Exception`

Common base class for all non-exit exceptions.

exception `cyclonedx_py.client.CycloneDxCmdNoInputFileSupplied`

Bases: `CycloneDxCmdException`

Common base class for all non-exit exceptions.

class `cyclonedx_py.client._CLI_OUTPUT_FORMAT`

Bases: `enum.Enum`

Generic enumeration.

Derive from this class to define new enumerations.

XML = `'xml'`

JSON = `'json'`

`cyclonedx_py.client._output_formats`

`cyclonedx_py.client._output_default_filenames`

`class cyclonedx_py.client.CycloneDxCmd(args: argparse.Namespace)`

`_DEBUG_ENABLED: bool = False`

`_arguments: argparse.Namespace`

`_get_output_format() → _CLI_OUTPUT_FORMAT`

`get_output() → cyclonedx.output.BaseOutput`

`execute() → None`

`static get_arg_parser(*, prog: str | None = None) → argparse.ArgumentParser`

`_debug_message(message: str, *args: Any, **kwargs: Any) → None`

`static _error_and_exit(message: str, *args: Any, exit_code: int = 1, **kwargs: Any) → None`

`_get_input_parser() → cyclonedx.parser.BaseParser`

`cyclonedx_py.client.main(*, prog_name: str | None = None, prog_name_instead: str | None = None) → None`

`cyclonedx_py.client.main_deprecated(*, prog_name: str | None = None) → None`

PYTHON MODULE INDEX

C

- `cyclonedx_py`, 29
- `cyclonedx_py.__main__`, 36
- `cyclonedx_py.client`, 36
- `cyclonedx_py.exception`, 29
- `cyclonedx_py.exception.parser`, 29
- `cyclonedx_py.parser`, 29
- `cyclonedx_py.parser._debug`, 29
- `cyclonedx_py.parser.conda`, 30
- `cyclonedx_py.parser.environment`, 31
- `cyclonedx_py.parser.pipenv`, 32
- `cyclonedx_py.parser.poetry`, 33
- `cyclonedx_py.parser.requirements`, 33
- `cyclonedx_py.utils`, 34
- `cyclonedx_py.utils.conda`, 34

Symbols

_BaseCondaParser (class in cyclonedx_py.parser.conda), 30
 _CLI_OUTPUT_FORMAT (class in cyclonedx_py.client), 36
 _DEBUG_ENABLED (cyclonedx_py.client.CycloneDxCmd attribute), 37
 _arguments (cyclonedx_py.client.CycloneDxCmd attribute), 37
 _conda_package_to_pypi_purl() (in module cyclonedx_py.utils.conda), 35
 _conda_packages_to_components() (cyclonedx_py.parser.conda._BaseCondaParser method), 31
 _debug_message() (cyclonedx_py.client.CycloneDxCmd method), 37
 _error_and_exit() (cyclonedx_py.client.CycloneDxCmd static method), 37
 _get_input_parser() (cyclonedx_py.client.CycloneDxCmd method), 37
 _get_metadata_for_package() (cyclonedx_py.parser.environment.EnvironmentParser static method), 32
 _get_output_format() (cyclonedx_py.client.CycloneDxCmd method), 37
 _output_default_filenames (in module cyclonedx_py.client), 37
 _output_formats (in module cyclonedx_py.client), 36
 _parse_to_conda_packages() (cyclonedx_py.parser.conda.CondaListExplicitParser method), 31
 _parse_to_conda_packages() (cyclonedx_py.parser.conda.CondaListJsonParser method), 31
 _parse_to_conda_packages() (cyclonedx_py.parser.conda._BaseCondaParser method), 30

B

base_url (cyclonedx_py.utils.conda.CondaPackage attribute), 34
 build_number (cyclonedx_py.utils.conda.CondaPackage attribute), 34
 build_string (cyclonedx_py.utils.conda.CondaPackage attribute), 34

C

channel (cyclonedx_py.utils.conda.CondaPackage attribute), 35
 conda_package_to_purl() (in module cyclonedx_py.utils.conda), 35
 CondaListExplicitParser (class in cyclonedx_py.parser.conda), 31
 CondaListJsonParser (class in cyclonedx_py.parser.conda), 31
 CondaPackage (class in cyclonedx_py.utils.conda), 34
 cyclonedx_py module, 29
 cyclonedx_py.__main__ module, 36
 cyclonedx_py.client module, 36
 cyclonedx_py.exception module, 29
 cyclonedx_py.exception.parser module, 29
 cyclonedx_py.parser module, 29
 cyclonedx_py.parser._debug module, 29
 cyclonedx_py.parser.conda module, 30
 cyclonedx_py.parser.environment module, 31
 cyclonedx_py.parser.pipenv module, 32
 cyclonedx_py.parser.poetry module, 33
 cyclonedx_py.parser.requirements module, 33

`cyclonedx_py.utils`
module, 34

`cyclonedx_py.utils.conda`
module, 34

`CycloneDxCmd` (class in `cyclonedx_py.client`), 37

`CycloneDxCmdException`, 36

`CycloneDxCmdNoInputFileSupplied`, 36

D

`DebugMessageCallback` (in module `cyclonedx_py.parser._debug`), 30

`dist_name` (`cyclonedx_py.utils.conda.CondaPackage` attribute), 35

E

`EnvironmentParser` (class in `cyclonedx_py.parser.environment`), 32

`execute()` (`cyclonedx_py.client.CycloneDxCmd` method), 37

G

`get_arg_parser()` (`cyclonedx_py.client.CycloneDxCmd` static method), 37

`get_output()` (`cyclonedx_py.client.CycloneDxCmd` method), 37

J

`JSON` (`cyclonedx_py.client._CLI_OUTPUT_FORMAT` attribute), 36

M

`main()` (in module `cyclonedx_py.client`), 37

`main_deprecated()` (in module `cyclonedx_py.client`), 37

`md5_hash` (`cyclonedx_py.utils.conda.CondaPackage` attribute), 35

module

`cyclonedx_py`, 29

`cyclonedx_py.__main__`, 36

`cyclonedx_py.client`, 36

`cyclonedx_py.exception`, 29

`cyclonedx_py.exception.parser`, 29

`cyclonedx_py.parser`, 29

`cyclonedx_py.parser._debug`, 29

`cyclonedx_py.parser.conda`, 30

`cyclonedx_py.parser.environment`, 31

`cyclonedx_py.parser.pipenv`, 32

`cyclonedx_py.parser.poetry`, 33

`cyclonedx_py.parser.requirements`, 33

`cyclonedx_py.utils`, 34

`cyclonedx_py.utils.conda`, 34

N

`name` (`cyclonedx_py.utils.conda.CondaPackage` attribute), 35

P

`package_format` (`cyclonedx_py.utils.conda.CondaPackage` attribute), 35

`parse_conda_json_to_conda_package()` (in module `cyclonedx_py.utils.conda`), 35

`parse_conda_list_str_to_conda_package()` (in module `cyclonedx_py.utils.conda`), 35

`PipEnvFileParser` (class in `cyclonedx_py.parser.pipenv`), 32

`PipEnvParser` (class in `cyclonedx_py.parser.pipenv`), 32

`platform` (`cyclonedx_py.utils.conda.CondaPackage` attribute), 35

`PoetryFileParser` (class in `cyclonedx_py.parser.poetry`), 33

`PoetryParser` (class in `cyclonedx_py.parser.poetry`), 33

Python Enhancement Proposals
PEP 508, 8

Q

`quiet()` (in module `cyclonedx_py.parser._debug`), 30

R

`RequirementsFileParser` (class in `cyclonedx_py.parser.requirements`), 33

`RequirementsParser` (class in `cyclonedx_py.parser.requirements`), 33

S

`split_package_build_string()` (in module `cyclonedx_py.utils.conda`), 35

`split_package_string()` (in module `cyclonedx_py.utils.conda`), 35

V

`version` (`cyclonedx_py.utils.conda.CondaPackage` attribute), 35

X

`XML` (`cyclonedx_py.client._CLI_OUTPUT_FORMAT` attribute), 36