

---

# **DDEX Reader for Python Documentation**

*Release 0.1.1*

**Flavian Hautbois**

September 14, 2015



<b>1</b>	<b>DDEX Reader for Python</b>	<b>3</b>
1.1	Features . . . . .	3
1.2	Supported DDEX versions . . . . .	3
1.3	Quickstart . . . . .	4
1.4	How to add more DDEX definitions . . . . .	4
<b>2</b>	<b>Installation</b>	<b>5</b>
<b>3</b>	<b>Usage</b>	<b>7</b>
<b>4</b>	<b>Contributing</b>	<b>9</b>
4.1	Types of Contributions . . . . .	9
4.2	Get Started! . . . . .	10
4.3	Pull Request Guidelines . . . . .	10
<b>5</b>	<b>Credits</b>	<b>11</b>
5.1	Development Lead . . . . .	11
5.2	Contributors . . . . .	11
<b>6</b>	<b>History</b>	<b>13</b>
<b>7</b>	<b>0.1.1 (2015-09-14)</b>	<b>15</b>
<b>8</b>	<b>0.1.0 (2015-01-11)</b>	<b>17</b>
<b>9</b>	<b>Indices and tables</b>	<b>19</b>



Contents:



---

## DDEX Reader for Python

---

This project allows you to read DDEX files into friendly Python data types. XML files are decoded using the PyXB library.

Keep in mind that this is a fairly low level library that only aims at making DDEX files easier to read using Python. Some DDEX data structures expose lists containing only one element, and some value like UpdateIndicator are not cast as booleans.

- Free software: MIT license
- Documentation: <https://ddexreader.readthedocs.org>.
- Repository: <https://github.com/Trax-air/ddexreader>

### 1.1 Features

- Open an XML file into a DDEX data structure generated by pyxb corresponding to the DDEX version.
- Parse this DDEX data structure into a Python dict.

### 1.2 Supported DDEX versions

- 3.1.2
- 3.2 (untested)
- 3.3
- 3.4
- 3.4.1
- 3.5
- 3.5.1
- 3.6

Version 3.7 is causing issues with PyXB.

## 1.3 Quickstart

```
from ddexreader import open_ddex, ddex_to_dict

xml_path = '/path/to/my/ddex_file.xml'
ddex = open_ddex(xml_path)
ddex_dict = ddex_to_dict(ddex)
```

## 1.4 How to add more DDEX definitions

After installing pyxb on your (unix) system, enter:

```
pyxbgen -u [the url to the definition file]
```



---

# Installation

---

At the command line:

```
$ easy_install ddexreader
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv ddexreader  
$ pip install ddexreader
```



---

### Usage

---

To use DDEX Reader for Python in a project:

```
import ddexreader
```

Also, make sure to check out the content of the [IPython notebook example](#).



---

## Contributing

---

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

### 4.1 Types of Contributions

#### 4.1.1 Report Bugs

Report bugs at <https://github.com/traxair/ddexreader/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

#### 4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

#### 4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

#### 4.1.4 Write Documentation

DDEX Reader for Python could always use more documentation, whether as part of the official DDEX Reader for Python docs, in docstrings, or even on the web in blog posts, articles, and such.

#### 4.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/traxair/ddexreader/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

## 4.2 Get Started!

Ready to contribute? Here's how to set up *ddexreader* for local development.

1. Fork the *ddexreader* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/ddexreader.git
```

3. Install your local copy into a virtualenv. Assuming you have *virtualenvwrapper* installed, this is how you set up your fork for local development:

```
$ mkvirtualenv ddexreader
$ cd ddexreader/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass *flake8* and the tests, including testing other Python versions with *tox*:

```
$ flake8 ddexreader tests
$ python setup.py test
$ tox
```

To get *flake8* and *tox*, just *pip* install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

## 4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in *README.rst*.
3. The pull request should work for Python 2.6, 2.7, 3.3, and 3.4, and for PyPy. Check [https://travis-ci.org/traxair/ddexreader/pull\\_requests](https://travis-ci.org/traxair/ddexreader/pull_requests) and make sure that the tests pass for all supported Python versions.

---

## Credits

---

### 5.1 Development Lead

- Flavian Hautbois <[flavian@trax-air.com](mailto:flavian@trax-air.com)>

### 5.2 Contributors

None yet. Why not be the first?





---

## History

---



---

**0.1.1 (2015-09-14)**

---

- Added support for ERN 3.1.2



---

**0.1.0 (2015-01-11)**

---

- First release on PyPI.



---

## Indices and tables

---

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