AST Unparser Documentation

Release 1.6.0

Simon Percivall

Sep 30, 2018

Contents

An AST unparser for Python.

This is a factored out version of `unparse` found in the Python source distribution; under Demo/parser in Python 2 and under Tools/parser in Python 3.

Basic example:

```
import inspect
import ast
import astunparse

# get back the source code
astunparse.unparse(ast.parse(inspect.getsource(ast)))

# get a pretty-printed dump of the AST
astunparse.dump(ast.parse(inspect.getsource(ast)))
```

This library is single-source compatible with Python 2.6 through Python 3.5. It is authored by the Python core developers; I have simply merged the Python 2.7 and the Python 3.5 source and test suites, and added a wrapper. This factoring out is to provide a library implementation that supports both versions.

Added to this is a pretty-printing `dump` utility function.

The test suite both runs specific tests and also roundtrips much of the standard library.

CHAPTER 1

Extensions and Alternatives

Similar projects include:

- codegen
- astor
- astmonkey
- astprint

None of these roundtrip much of the standard library and fail several of the basic tests in the `test_unparse` test suite.

This library uses mature and core maintained code instead of trying to patch existing libraries. The `unparse` and the `test_unparse` modules are under the PSF license.

Extensions include:

- typed-astunparse: extends astunparse to support type annotations.
- Documentation: http://astunparse.rtfd.org.

CHAPTER 2

Features

- unparses Python AST.
- pretty-prints AST.

CHAPTER 3

Contents:

3.1 Installation

At the command line:

```
$ easy_install astunparse
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv astunparse
$ pip install astunparse
```

3.2 Usage

To use AST Unparser in a project:

```
import astunparse
```

Then use the `unparse()` function to unparse an AST tree.

3.3 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:

3.3.1 Types of Contributions

Report Bugs

Report bugs at https://github.com/simonpercivall/astunparse.

If you are reporting a bug, please include:

- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

Fix Bugs

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

Implement Features

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

Write Documentation

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

Submit Feedback

The best way to send feedback is to file an issue at https://github.com/simonpercivall/astunparse/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 :)

3.3.2 Get Started!

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

1. Check out the repository.

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

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

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

6. Commit and send the patch or create a pull request.

3.3.3 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 2.7, and 3.3.

3.4 Credits

3.4.1 Maintainer

- Simon Percivall <percivall@gmail.com>

3.4.2 Authors

- The Python Software Foundation
- Bogdan Opanchuk
- Vladimir Iakovlev
- Thomas Grainger
- Amund Hov
- Jakub Wilk
- Mateusz Bysiek

3.5 History

Here's the recent changes to AST Unparser.

3.5.1 1.6.0 - 2018-09-30

- Python 3.7 compatibility

3.5.2 1.5.0 - 2017-02-05

- Python 3.6 compatibility
- bugfix: correct argparser option type

3.5.3 1.4.0 - 2016-06-24

- Support for the `async` keyword
- Support for unparsing "Interactive" and "Expression" nodes

3.5.4 1.3.0 - 2016-01-17

- Python 3.5 compatibility

3.5.5 1.2.0 - 2014-04-03

- Python 2.6 through 3.4 compatibility
- A new function `dump` is added to return a pretty-printed version of the AST. It's also available when running `python -m astunparse` as the `--dump` argument.

3.5.6 1.1.0 - 2014-04-01

- `unparse` will return the source code for an AST. It is pretty feature-complete, and round-trips the stdlib, and is compatible with Python 2.7 and Python 3.4.

  Running `python -m astunparse` will print the round-tripped source for any python files given as argument.

CHAPTER 4

Indices and tables

- genindex
- modindex
- search