Search projects

Help Donate Log in Register

lxml 4.4.0

Latest version

```
pip install lxml
```

Last released: Jul 27, 2019

Powerful and Pythonic XML processing library combining libxml2/libxslt with the ElementTree API.

Navigation

Project description

Release history

Download files

Project links

Homepage

Statistics

Project description

lxml is a Pythonic, mature binding for the libxml2 and libxslt libraries. It provides safe and convenient access to these libraries using the ElementTree API.

It extends the ElementTree API significantly to offer support for XPath, RelaxNG, XML Schema, XSLT, C14N and much more.

To contact the project, go to the project home page or see our bug tracker at https://launchpad.net/lxml

In case you want to use the current in-development version of lxml, you can get it from the github repository at https://github.com/lxml/lxml . Note that this requires Cython to build the sources, see the build instructions on the project home page. To the same end, running `easy_install lxml==dev` will install lxml from https://github.com/lxml/lxml/tarball/master#egg=lxml-dev if you have an appropriate version of Cython installed.

View statistics for this project via Libraries.io, or by using Google BigQuery

Meta

License: BSD License (BSD)

Author: lxml dev team

Maintainer: lxml dev team

Maintainers


faassen


scoder


zope.winegg builder

Classifiers

Development Status
5 - Production/Stable

Intended Audience
Developers
Information Technology

License
OSI Approved :: BSD License

After an official release of a new stable series, bug fixes may become available at https://github.com/lxml/lxml/tree/lxml-4.4 . Running `easy_install lxml==4.4bugfix` will install the unreleased branch state from https://github.com/lxml/lxml/tarball/lxml-4.4#egg=lxml-4.4bugfix as soon as a maintenance branch has been established. Note that this requires Cython to be installed at an appropriate version for the build.

4.4.0 (2019-07-27)

Features added

- `Element.clear()` accepts a new keyword argument `keep_tail=True` to clear everything but the tail text. This is helpful in some document-style use cases.
- When creating attributes or namespaces from a dict in Python 3.6+, lxml now preserves the original insertion order of that dict, instead of always sorting the items by name. A similar change was made for ElementTree in CPython 3.8. See https://bugs.python.org/issue34160
- Integer elements in `lxml.objectify` implement the `__index__()` special method.
- GH#269: Read-only elements in XSLT were missing the `nsmap` property. Original patch by Jan Pazdziora.
- ElementInclude can now restrict the maximum inclusion depth via a `max_depth` argument to prevent content explosion. It is limited to 6 by default.
- The `target` object of the XMLParser can have `start_ns()` and `end_ns()` callback methods to listen to namespace declarations.
- The `TreeBuilder` has new arguments `comment_factory` and `pi_factory` to pass factories for creating comments and processing instructions, as well as flag arguments `insert_comments` and `insert_pis` to discard them from the tree when set to false.

Operating System
OS Independent

Programming Language
C
Cython
Python :: 2
Python :: 2.7
Python :: 3
Python :: 3.4
Python :: 3.5
Python :: 3.6
Python :: 3.7

Topic
Software Development :: Libraries :: Python Modules
Text Processing :: Markup :: HTML
Text Processing :: Markup :: XML

- A C14N 2.0 implementation was added as `etree.canonicalize()`, a corresponding `C14NWriterTarget` class, and a `c14n2` serialisation method.

Bugs fixed

- When writing to file paths that contain the URL escape character '%', the file path could wrongly be mangled by URL unescaping and thus write to a different file or directory. Code that writes to file paths that are provided by untrusted sources, but that must work with previous versions of lxml, should best either reject paths that contain '%' characters, or otherwise make sure that the path does not contain maliciously injected '%XX' URL hex escapes for paths like '../'.
- Assigning to Element child slices with negative step could insert the slice at the wrong position, starting too far on the left.
- Assigning to Element child slices with overly large step size could take very long, regardless of the length of the actual slice.
- Assigning to Element child slices of the wrong size could sometimes fail to raise a ValueError (like a list assignment would) and instead assign outside of the original slice bounds or leave parts of it unreplaced.
- The `comment` and `pi` events in `iterwalk()` were never triggered, and instead, comments and processing instructions in the tree were reported as `start` elements. Also, when walking an ElementTree (as opposed to its root element), comments and PIs outside of the root element are now reported.
- LP#1827833: The RelaxNG compact syntax support was broken with recent versions of `rnc2rng`.
- LP#1758553: The HTML elements `source` and `track` were added to the list of empty tags in `lxml.html.defs`.
- Registering a prefix other than "xml" for the XML namespace is now rejected.
- Failing to write XSLT output to a file could raise a misleading exception. It now raises `IOError`.

Other changes

- Support for Python 3.4 was removed.
- When using `Element.find*()` with prefix-namespace mappings, the empty string is now accepted to define a default namespace, in addition to the previously supported `None` prefix. Empty strings are more convenient since they keep all prefix keys in a namespace dict strings, which simplifies sorting etc.
- The `ElementTree.write_c14n()` method has been deprecated in favour of the long preferred `ElementTree.write(f, method="c14n")`. It will be removed in a future release.