Setup: The same XHTML5 document is created with different APIs. All output implementations of EcoXiPy are tested as well as "xml.sax" and "xml.dom.minidom". For each of the APIs one test creates its native representation and one test transforms this into an UTF-8 encoded byte string, as mostly XML is serialized this way. The SAX and "ecoxipy.string_output" tests create byte strings in both test types.
Platform: 2.4 GHz Intel Core 2 Duo, 8 GB RAM on Mac OS X 10.9

The following table shows the running times in seconds for the tests for the different Platforms and output types.

Output	Bytes	Bytes	Bytes	Native	Native	Native
Python Platform	PyPy 2.2.1	CPython	CPython	PyPy 2.2.1	CPython	CPython
Python Version	2.7.3	3.3.4	2.7.6	2.7.3	3.3.4	2.7.6
xml.sax	7.183	11.067	15.576	4.664	10.907	15.221
xml.etree	3.970	6.598	6.379	0.195	0.677	0.968
xml.dom.minidom	10.434	22.065	24.756	2.059	10.892	9.928
ecoxipy.dom_output	6.865	26.600	29.270	3.550	15.678	15.551
ecoxipy.etree_output	2.874	12.691	12.654	1.247	6.966	7.701
ecoxipy.pyxom.output	6.549	20.413	20.951	2.338	10.263	11.943
ecoxipy.string_output	3.546	13.228	12.756	3.366	11.799	11.494

Performance Test Results

Platform: 2.4 GHz Intel Core 2 Duo, 8 GB RAM on Mac OS X 10.9

Running Time [seconds]

Output, Platform

Native
CPython
2.7.6
3.3.4
PyPy 2.2.1 2.7.3

Bytes
CPython
2.7.6
3.3.4
PyPy 2.2.1 2.7.3

0 5 10 15 20 25 30 35

- xml.sax
- xml.etree
- xml.dom.minidom
- ecoxipy.dom_output
- ecoxipy.etree_output
- ecoxipy.pyxom.output
- ecoxipy.string_output