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 2.1.0	CPython	CPython	PyPy 2.1.0	CPython	CPython
Python Version	2.7.3	3.3.3	2.7.6	2.7.3	3.3.3	2.7.6
xml.sax	5.364	10.595	15.110	5.042	10.575	14.913
xml.etree	3.427	6.423	6.261	0.203	0.664	0.987
xml.dom.minidom	10.279	21.691	24.848	2.034	10.367	10.411
ecoxipy.dom_output	6.739	25.315	28.852	3.856	14.917	15.333
ecoxipy.etree_output	2.742	11.763	12.174	1.189	6.131	7.498
ecoxipy.pyxom.output	6.107	19.795	20.783	2.091	9.721	11.641
ecoxipy.string_output	3.471	10.981	11.316	3.328	10.952	11.278

ECoXiPy Peformance Test Results - Graph

Performance Test Results

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

Running Time [seconds]

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