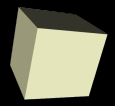
ElementFlow

Whitemice Consulting

Author: Ivan Sagalaev <Maniac@SoftwareManiacs Org>
License: BSD variant
Source: http://github.com/ryanmcgrath/twython
Home Page:<https://launchpad.net/elementflow>
Dependencies: textwrap, codecs



Description

Elementflow is a Python library for generating XML as a stream. Some existing XML producing libraries (like ElementTree, lxml) build a whole XML tree in memory and then serialize it. It might be inefficient for moderately large XML payloads (think of a content-oriented Web service producing lots of XML data output). Python's built-in xml.sax.saxutils.XMLGenerator is very low-level and requires closing elements by hand.

Also, most XML libraries, to be honest, suck when dealing with namespaces.
<http://pypi.python.org/pypi/elementflow>

Example

```
file = open('text.xml', 'w') # can be any  object with .write() method

with elementflow.xml(file, u'root') as xml:
    xml.element(u'item', attrs={u'key': u'value'}, text=u'text')
    with xml.container(u'container', attrs={u'key': u'value'}):
        xml.text(u'text')
        xml.element(u'subelement', text=u'subelement text'
```

- Raises an exception if an undefined namespace prefix is used.
- Namespaces may be defined for any container.

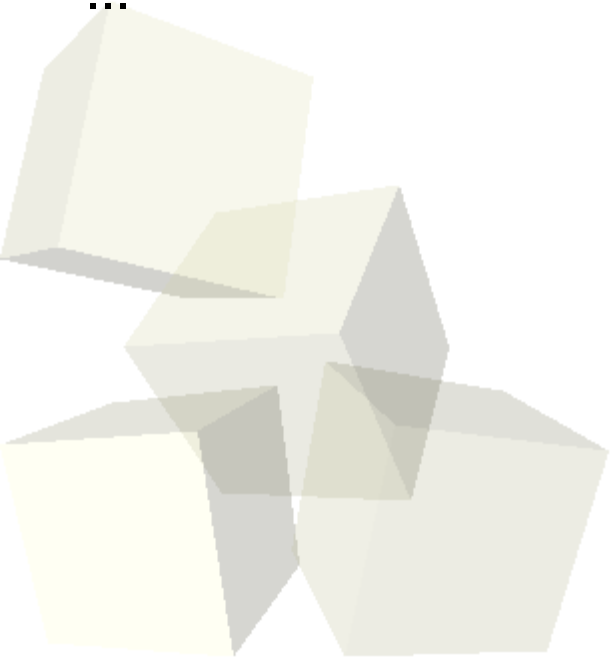
Namespaces

```
with elementflow.xml(file, 'root', namespaces={'': 'urn:n', 'n1': 'urn:n1'}) as xml:
    xml.element('item')
    with xml.container('container', namespaces={'n2': 'urn:n2'):
        xml.element('n1:subelement', text='YoYo Mama')
        xml.element('n2:subelement')
```

- Raises an exception if an undefined namespace prefix is used.
- Namespaces may be defined for any container.

Pretty Print

```
with elementflow.xml(file, 'root', indent=True):
  ...
```



As a generator

```
def g():
    xml = elementflow.xml(elementflow.Queue(), 'root')
    with xml:
        for item in collection:
            xml.element(...)
            if len(xml.file) > BUFSIZE:
                yield xml.file.pop() #Yield only when the buffer exceeds a given size
    yield xml.file.pop()
```

- elementflow.Queue is a buffer that is cleared by calling pop()

Write to the stream

```
with elementflow.xml(stream, u'D:multistatus', indent=True,
                                                    namespaces=ns) as response:
    for resource in resources:
        with response.container('D:response'):
            response.file.write(u'<D:href>{0}</D:href>'.format(resource.webdav_url))
            ...
```

- Of course you looks auto-indenting, tag closure, and namespace verification.