XLST With Python

XSLT Is Awesome

- XSLT is better than your code.
  - libxslt is old, fast, and completely debugged.
- XSLT is a standard, people know XSLT
  - They do not know your code.
- XSLT is faster then your code.
  - Yes, it is.
  - A lot of problems have already been solved.

Python is Awesome

- This needs no explanation.

XSLT 1.0 Limitations

- XSLT has some string manipulation methods
  - But it lacks some others, such as replace.
- XSLT doesn't do state.
  - It is a template / transform solution!
- XSLT knows nothing about dates.
  - Dates are the bane of our existence.

Transform!

```
from lxml import etree

rfile = open('anxmlfile.xml', 'rb')
wfile = open('thenewfile.xml', 'wb')
tfile = open('thetemplate.xslt', 'rb')

#Read the source file
source = etree.parse(rfile)

#Create the transform
xslt = etree.XSLT( etree.XML(tfile.read()))

'''
LXML docs say the correct thing to do to get the result of the transform
is to call "str( XLST-apply )".  These seems *TOTALLY* dork-wad and almost
certainly means the entire result exists in memory at least momentarily
We love LXML & E-Tree, but really?!?!  This is hackish, it just stinks.
See http://lxml.de/api/lxml.etree.XSLT-class.html if you don't believe me.
As the song says, "Life goes on, long after the belief in beautiful code is
gone.
'''
wfile.write(str(xslt(source)))
wfile.close()
```

Extending XSLT

```
oie_extentions = OIEXSLTExtensionPoints( context=self._ctx,
                                         process=self.process,
                                         scope=self.scope_stack,
                                         ctxids=self._ctx_ids )

extensions = etree.Extension(
    oie_extentions,
    (
     'sequencereset',      # Reset/set the value of a named sequence
     'sequencevalue',      # Retrieve the value of a named sequence
     'sequenceincrement',  # Increment a sequence, returning the new value
     'messagetext',        # Retrieve the content of a message by label
     'searchforobjectid',  # Search for an objectId by criteria
     'tablelookup',        # Performa table lookup
     'reformatdate',       # Reformat a StandardXML date to some other format
     ...
    ),
    ns='http://www.opengroupware.us/oie')
```

The Extension Object

```
class OIEXSLTExtensionPoints:

    def __init__(self, process, context, scope, ctxids):
        self.process = process
        self.context = context
        self.scope   = scope
        self.ctxids  = ctxids
        self.tables  = { }


….
    def reformatdate(self, _, value, format):
        value = self._make_date( value, 'reformatdate', 'date' )
        return value.strftime(format)
```

Declare Your Namespace

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0"
                xpath-default-namespace="http://www.w3.org/1999/xhtml"
                xmlns="http://www.w3.org/1999/xhtml"
                xmlns:oie="http://www.opengroupware.us/oie"
                xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
   <xsl:output method="text"/>
```

Use Your Extension Method

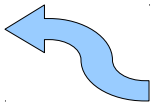
```
.....
  <xsl:template match="row">
    <xsl:value-of select="vendor_id"/>
    <xsl:text>,</xsl:text>
    <xsl:value-of select="invoice_number"/>
    <xsl:text>,</xsl:text>
    <xsl:value-of select="oie:reformatdate(string(invoice_date), '%m/%d/%Y')"/>
    <xsl:text>,</xsl:text>
    <xsl:value-of select="net_amount"/>
    <xsl:text>,</xsl:text>
    <xsl:text>,</xsl:text>
    <xsl:text>,</xsl:text>
    <xsl:text>,</xsl:text>
    <xsl:value-of select="check_number"/>
    <xsl:text>,</xsl:text>
    <xsl:value-of select="oie:reformatdate(string(check_date), '%m/%d/%Y')"/>
    <xsl:text>,</xsl:text>
    <xsl:value-of select="oie:replace(string(vendor_name), ',', '')"/>
    <xsl:text>&#xA;</xsl:text>
  </xsl:template>

</xsl:stylesheet>
```

Getting some state!

```
class OIEXSLTExtensionPoints:

    def __init__(self, process, context, scope, ctxids):
        self.process = process
        self.context = context
        self.scope   = scope
        self.ctxids  = ctxids
        self.tables  = { }x
```



We can pass stuff to the extension object.

```
def sequenceincrement(self, _, scope, name, increment):
    value = self._get_sequence_value(scope, name)
    value += increment
    self._set_sequence_value(scope, name, value)
    return unicode(value)
```

```
<xsl:value-of
    select="oie:sequenceincrementt('route', 'masternumber, 1)"/>
```

More State

Provide a way to pass some state[like variables] into a templates processing.

```
def xattrvalue(self, _, name):
    name = name.strip().lower()
    prop = self.context.property_manager.get_property(
      self.process,
          'http://www.opengroupware.us/oie',
           'xattr_{0}'.format(name))
    if prop:
        return prop.get_string_value())
    return u''
```

```
<company dataType="string">
    <xsl:value-of select="oie:xattrvalue('company')"/>
</company>
```