



# OOoPy: OpenOffice.org Dokumente in anderen Applikationen

Dr. Ralf Schlatterbeck  
Open Source Consulting

Email: [office@runtux.com](mailto:office@runtux.com)  
Web: <http://www.runtux.com>  
Tel. +43/650/621 40 17



## Inhalt

OOoPy Features	3
OOoPy Anwendungen	5
Aufbau eines OOo Dokuments	6
API: General	8
API: Pure OOoPy	9
Transformer API	10
Transform API	11
Attribute Access API	12
Voraussetzungen, Download	13
Roadmap	14



## OOoPy Features

- Unabhängig von OpenOffice.org (OOo)
- Lesen, Schreiben, Ändern von OOo Dokumenten
- Bibliothek für Python → OOoPy
- Plattformunabhängigkeit
- Framework für Transformationen
- Basierend auf **ElementTree**



## OOoPy Features (2)

- Existierende Transformationen:
  - Verändern von „Feldbefehlen“ (“Fields” in english version)
  - Serienbrief (Mailmerge)
  - Zusammenhängen von Dokumenten
  - Viele kleine Utilities
- Transformationen derzeit nur für OOo 1.X
- Framework funktioniert auch für 2.X
- Patch auf **SourceForge** für 2.X nicht integriert



## OOoPy Anwendungen

- Applikations-Integration
- serverseitige Web-Anwendungen
- Reportgeneratoren
- Generieren von OOo Dokumenten aus anderen Formaten
  - Als Text (OOo Writer)
  - OOo Calc
  - OOo Presenter
  - ...



## Aufbau eines OOo Dokuments

- ZIP-File, ähnlich wie Java .jar Format
- Strukturierte Information in XML
- `mimetype: application/vnd.sun.xml.writer`
- Inhaltsverzeichnis: `META-INF/manifest.xml`
- Einstellungen von OOo: `settings.xml`
- Vorlagen (Styles): `styles.xml`



## Aufbau eines OOo Dokuments (2)

- Meta-Information: `meta.xml`
  - Seiten- Absatz- Wort- Zeichenzahl, ...
  - Versionsinformation
  - Benutzerdefinierte Felder
- Inhalt: `content.xml`
  - Ad-hoc Vorlagen
  - Textkörper
- Jedes Objekt (Bild, ...) in eigener Datei



## API: General

- Apply `ElementTree` primitives (e. g. for document contents generation)
- ... or Transforms (e. g. Mailmerge)
- OOoPy gets input and output files  
→ Zip module can't modify a file in place
- Files can be filenames or open files
- e. g., in-memory files using `StringIO`
- a single Transformer can contain multiple Transforms to be applied to a document



## API: Pure OOoPy

```

from ooopy.OOoPy import OOoPy
from ooopy.Transformer import OOo_Tag
from elementtree.ElementTree import SubElement
o = OOoPy (infile = 'test.sxw', outfile = 'out.sxw')
e = o.read ('content.xml')
body = e.find ('.//' + OOo_Tag ('office', 'body'))
p = SubElement (body, OOo_Tag ('text', 'p'))
p.text = 'Paragraph text'
e.write ()
o.close ()

```



## Transformer API

```

sio = StringIO ()
o = OOoPy (infile = 'test.sxw', outfile = sio)
t = Transformer \
    ( get_meta
      , Concatenate ('test.sxw', 'rechng.sxw')
      , Attribute_Access
        ( ( renumber_frames
            , renumber_sections
            , renumber_tables
          ) )
      , set_meta
    )
t.transform (o)
o.close ()

```



## Transform API

- constructor should accept prio argument
- Transforms applied in prio order by Transformer
- either define `apply` and `filename`  
→ transform works with only one file
- or use `apply_all`  
→ need to access several files,  
e. g., `contents.xml` and `meta.xml`
- Transforms must not retain state after `apply_all`



## Attribute Access API

- read or change attributes of nodes in an OOo xml document
- only one pass over the document
- renumbering
- re-anchoring (e. g., text boxes to different pages)
- reading/updating attributes (e. g. page-count)



## Voraussetzungen, Download

Benötigte Software:

- [OpenOffice Version 1.X]
- mindestens Python 2.3
- ElementTree XML-Library von Fredrik Lundh  
<http://effbot.org/zone/element-index.htm>

OOoPy:

- <http://runtux.com/ooopy.html>
- <http://ooopy.sourceforge.net/>



## Roadmap

- current version 0.2 with mailmerge and concatenation of documents
- **SourceForge** activity percentile 89.17%
- post 0.2: New OOo 2.X Oasis document format
- post 0.2: Transformations for OOo Calc documents
- idea: Generation of presentations, see also Paul Barry, Linux Journal **04/2005**:  
*Building Impress and Powerpoint Slides with  $\LaTeX$  and Perl*