

# OpenOffice

December 8, 2008

## Contents

<b>1</b>	<b>Install</b>	<b>2</b>
<b>2</b>	<b>OpenOffice and Python</b>	<b>2</b>
2.1	Start openoffice . . . . .	2
2.2	Connect to OpenOffice . . . . .	2
2.3	Modify Content . . . . .	3
2.3.1	Load Services . . . . .	3
2.3.2	Load Current Document . . . . .	3
2.3.2.1	set cursor . . . . .	3
2.3.3	Load New Document . . . . .	3
2.3.4	Load Existing Document . . . . .	4
2.3.5	Modifying text . . . . .	4
2.3.5.1	Line breaks . . . . .	4
2.3.5.2	Indent . . . . .	4
2.3.5.3	Change Font property . . . . .	4
2.3.6	Create Table . . . . .	5
2.4	Save . . . . .	5
2.4.1	Save Changes . . . . .	5
2.4.2	Save file as . . . . .	6
2.4.3	Exit the file . . . . .	6
2.5	Manipulate . . . . .	6
2.5.1	Find and Replace . . . . .	6
2.5.2	Save to PDF . . . . .	7
<b>3</b>	<b>Examples</b>	<b>7</b>
3.1	Create Document . . . . .	7
3.2	Find and Replace . . . . .	7
3.3	SqlAlchemy and OpenOffice . . . . .	8
<b>4</b>	<b>References</b>	<b>13</b>
	Keywords: OpenOffice, Python, Uno, Templates, Document Generation, <a href="#">FoxPro</a> Forms, vb Forms, Word Forms,	

# 1 Install

- Install OpenOffice

```
aptitude update  
aptitude install openoffice.org
```

- Install python uno (pyuno)

```
aptitude install python-uno
```

- Done.

## 2 OpenOffice and Python

### 2.1 Start openoffice

- Start openoffice so it listens on a port.

```
openoffice -accept="socket,host=localhost,port=2002;urp;"
```

### 2.2 Connect to OpenOffice

- Open UNO. These are python bindings to openoffice api.
- Start python:

```
python
```

- Then import uno.

```
import uno
```

- Now import the OpenOffice component context.

```
local = uno.getComponentContext()
```

- Now access the [UnoUrlResolver](#) service. This will allow you to connect to OpenOffice.org program.

```
resolver = local.ServiceManager.createInstanceWithContext(" ↔  
com.sun.star.bridge.UnoUrlResolver", local)
```

- Now load the context and you are now connected. You can access OpenOffice via its API mechanism.

```
context = resolver.resolve("uno:socket,host=localhost,port ↔  
=2002;urp;StarOffice.ComponentContext")
```

## 2.3 Modify Content

### 2.3.1 Load Services

- There is a service responsible for the current document which is called desktop. This service can be access by the name "com.sun.star.frame.Desktop".
- We will be using that service so we need to create it/load it.

```
desktop = context.ServiceManager.createInstanceWithContext(" ↔  
com.sun.star.frame.Desktop", context)
```

- We now have access to the Desktop service. We now can load and manage documents.

### 2.3.2 Load Current Document

- To get the current document we call *getCurrentComponent* method:

```
document = desktop.getCurrentComponent()
```

#### 2.3.2.1 set cursor

- We need to create a cursor that we will use to write documents.

```
cursor = document.Text.createTextCursor()
```

### 2.3.3 Load New Document

- If you want to create another page you do the following:

```
document = desktop.loadComponentFromURL("private:factory/ ↔  
swriter", "_blank", 0, ())  
cursor = document.Text.createTextCursor()
```

### 2.3.4 Load Existing Document

- To load existing document do:

```
document = desktop.loadComponentFromURL("file:///home/lucas/ ↵  
myfile.odt" , "_blank", 0, ())  
or  
document = desktop.loadComponentFromURL("file:///c:"+filepath ↵  
+".odt" , "_blank", 0, ())
```

### 2.3.5 Modifying text

- Lets add some text:

```
document.Text.insertString(cursor, "This text is being added ↵  
to openoffice using python and uno package.", 0)
```

- Your text should show up on you screen.

#### 2.3.5.1 Line breaks

- Line breaks :

```
document.Text.insertString(cursor, "\n\nThis is a new ↵  
paragraph.", 0)
```

- The text is entered right after the previous sentence.

#### 2.3.5.2 Indent

- To indent a add \t to a file:

```
document.Text.insertString(cursor, "\n\n\tAnd this is another ↵  
new paragraph.", 0)
```

#### 2.3.5.3 Change Font property

- You can manipulate the cursor just like you would in OpenOffice Writer. You can change a porperty of the cursor, example:font size.
- Lets get the current font size:

```
cursor.getPropertyValue("CharHeight")
```

- Change Font Size, Font Name, Font Weight:

```
cursor.setPropertyValue("CharHeight", 20)  
cursor.setPropertyValue("CharFontName", "Arial")  
cursor.setPropertyValue("CharWeight", 150)
```

### 2.3.6 Create Table

- To create table we do the following. We will call our table object mytable:

```
mytable = document.createInstance("com.sun.star.text. ↵  
TextTable")
```

- Now lets set the number of rows and columns in the table.

```
mytable.initialize(6,2)
```

- Now you need to insert the table into document:

```
document.Text.insertTextContent(cursor, mytable, 0)
```

- Now you need to fill the table. Each cell is being referenced by its name. First row is named "A", and the first column is named "1". Second row is named "B" and first column is "1", second column is "2".....

```
mytable.getCellByName("A1").setString("This is first column")  
mytable.getCellByName("B1").setString("This is second column ↵  
")
```

- You fill the rest of the clumns in a similar way.

```
mytable.getCellByName("A2").setString("Row 2 is here")  
mytable.getCellByName("B2").setString("Here is column 2")  
mytable.getCellByName("A3").setString("This manual")  
mytable.getCellByName("B3").setString("You see")  
mytable.getCellByName("A4").setString("was")  
mytable.getCellByName("B4").setString("how easy")  
mytable.getCellByName("A5").setString("Made in")  
mytable.getCellByName("B5").setString("it is ")  
mytable.getCellByName("A6").setString("Chicago")  
mytable.getCellByName("B6").setString("to do things like this ↵  
!")
```

## 2.4 Save

### 2.4.1 Save Changes

- Save changes

```
document.store()
```

## 2.4.2 Save file as

- To save file do:

```
document.storeAsURL("file:///home/lucas/myfile2.odt", ())
```

## 2.4.3 Exit the file

- To exit from the file do:

```
document.dispose()
```

## 2.5 Manipulate

### 2.5.1 Find and Replace

- If you want to find and replace text you can use below code to do it. We use python string module to do the work:

```
import string
#Create Search Descriptor
search = document.createSearchDescriptor()
#What to search for
search.SearchString = u"Lucas"
#Found string
found = document.findFirst( search )

while found:
    found.String = string.replace( found.String, u"Lucas", u" ←
        NewLucas" )
    found = document.findNext( found.End, search)
#Save
document.store()
```

- Some other search option include:

```
search = document.createSearchDescriptor()
search.SearchString = "search for"
search.SearchCaseSensitive = True
search.SearchWords = True
or
search = document.createSearchDescriptor()
search.SearchRegularExpression = True
search.SearchString = "\\<(k|s|v|z|o|u|i|a) "
```

## 2.5.2 Save to PDF

- You can save the file as pdf by doing:

```
from com.sun.star.beans import PropertyValue
property = (
    PropertyValue( "FilterName" , 0, "writer_pdf_Export" , 0 ←
    ),
)
document.storeToURL("file:///home/lucas/myfile2.pdf",property ←
)
```

# 3 Examples

## 3.1 Create Document

## 3.2 Find and Replace

```
"""This program will do 'find and replace', and save an ←
output to a different file"""
#You should have openoffice listening on specified port ←
already.
#Load necessary items
import uno
local = uno.getComponentContext()
resolver = local.ServiceManager.createInstanceWithContext(" ←
com.sun.star.bridge.UnoUrlResolver", local)
context = resolver.resolve("uno:socket,host=localhost,port ←
=2002;urp;StarOffice.ComponentContext")
desktop = context.ServiceManager.createInstanceWithContext(" ←
com.sun.star.frame.Desktop", context)

#Load file template
document = desktop.loadComponentFromURL("file:///home/lucas/ ←
TemplateLetter.odt" , "_blank", 0, ())
cursor = document.Text.createTextCursor()

#-----Start doing stuff-----
import string

#Create Search Descriptor
print 'starting a search'
search = document.createSearchDescriptor()

def findandreplace(document=document, search=search, find=None, ←
replace=None):
```

```

"""This function searches and replaces. Create search, ↵
    call function findFirst, and finally replace what we ↵
    found."""
#What to search for
search.SearchString = unicode(find)
#search.SearchCaseSensitive = True
#search.SearchWords = True
found = document.findFirst( search )
if found:
    print 'Found %s' % find
while found:
    found.String = string.replace( found.String, unicode( ↵
        find),unicode(replace))
    found = document.findNext( found.End, search)

#Create data structure of what will get replaced. I assume ↵
    that "$FirstName", etc. are already in you odt file.
data={}
data['$FirstName']='Lucas'
data['$LastName']='Mylastname'
data['$Address']='123 Main St.'
data['$City']='Chicago'
data['$State']='IL'
data['$Zipcode']='60645'
data['$TodaysDate']='20080821'
data['$DOB']='My Date of Birth'

#Do a loop of the data and replace the content.
for find,replace in data.items():
    findandreplace(document,search,unicode(find),unicode( ↵
        replace))
    print find,replace

#Save document
document.storeAsURL("file:///home/lucas/letter2.odt",())
#Close
document.dispose()

```

### 3.3 SqlAlchemy and OpenOffice

- This program will get records using Sqlalchemy and Fill in the OpenOffice Fields.
- You might need to change the database name, Table name, and fields names. I use Addressbook.

- You also need to create an odt file that has "\$FirstName"... in it. Find and Replace function cannot search inside of the text fields.
- If you are using vim, set the paste mode on before pasting in this code.

```
set paste
```

```
"""This program Gets records from sqlalchemy and replaces ↵
    templates fields inside OpenOffice
Copyright Lukasz Szybalski szybalski@gmail.com
License: GNU General Public License 2 (GPL2)"""
#Following is if we are importing a file, or setting it up in ↵
    a cronjob. This will switch folder to where the .py is ↵
    located at.
import os
workfolder=os.path.dirname(__file__)
if workfolder:
    print __file__
    os.chdir(workfolder)

#Start Logging error
import logging
logging.basicConfig()
#Setup config
log = logging.getLogger("MyApp")
log.setLevel(logging.DEBUG) #set verbosity to show all ↵
    messages of severity >= DEBUG
#log.setLevel(logging.INFO) #set verbosity to show all ↵
    messages of severity >= DEBUG
log.info("Starting my app")

import sqlalchemy
#Mysql
e = sqlalchemy.create_engine('mysql://user:pass@hostname/ ↵
    db_name')
#Postgres
#e = sqlalchemy.create_engine('postgres://user:pass@hostname/ ↵
    db_name')
#Mssql
#e = sqlalchemy.create_engine("mssql://user:pass@hostname ↵
    :1433/db_name?driver=TDS&odbc_options='TDS_Version=8.0' ")
#e.echo=True
e.echo=False
metadata=sqlalchemy.MetaData(e)
log.info('Connection Set')

from sqlalchemy.orm import sessionmaker
Session = sessionmaker(bind=e, autoflush=True, autocommit= ↵
    False)
```

```

session = Session()
#Connected to a database
log.info("Connected to a Database")

#----Table Definition and Python object that maps to it ←
-----
from sqlalchemy.orm import mapper
#Addressbook master File
addressbookrecords_table = sqlalchemy.Table('Addressbook', ←
    metadata, autoload=True)
#Python Object
class AddressbookRecords(object):
    pass
#Mapper
mapper(AddressbookRecords,addressbookrecords_table)
log.info('Connected to a database, and have all the field ←
    names')

#-----Start of OpenOffice Section-----
"""This progrma will do find and replace and save an output ←
    to a different file"""
#Load necessary items
import uno
import sys
#You need to set few variables when using OpenOffice
local = uno.getComponentContext()
resolver = local.ServiceManager.createInstanceWithContext(" ←
    com.sun.star.bridge.UnoUrlResolver", local)
#One of the variables is context which connects to openoffice ←
    . We test here if open office is listenting, if not we ←
    start it.
try:
    #Am I able to connect?
    context = resolver.resolve("uno:socket,host=localhost, ←
        port=2002;urp;StarOffice.ComponentContext")
    log.debug('Connecting to OpenOffice')

except Exception ,e :
    #If I got the error
    if 'connect to socket' in str(e):
        import os
        #Start it
        os.system('"/usr/bin/openoffice -accept="socket,host ←
            =localhost,port=2002;urp;"')
        import time
        #Wait 3 seconds for openoffice to load.
        time.sleep(3)
        #See if we can connect again

```

```

        context = resolver.resolve("uno:socket,host=localhost ↵
        ,port=2002;urp;StarOffice.ComponentContext")
        log.debug('OpenOffice was not opened. I have opened ↵
        it and connected')
#Rest of the necessary variables.
desktop = context.ServiceManager.createInstanceWithContext(" ↵
        com.sun.star.frame.Desktop", context)

log.debug('Done Setting up all OpenOffice Variables')

#-----Start manipulating openoffice document-----
import string

#We will use this function to find and replace things on the ↵
page.
def findandreplace(document=None,search=None,find=None, ↵
replace=None):
    """This function searches and replaces. Create search, ↵
    call function findFirst, and finally replace what we ↵
    found."""
    #What to search for
    search.SearchString = unicode(find)
    #search.SearchCaseSensitive = True
    #search.SearchWords = True
    found = document.findFirst( search )
    if found:
        #log.debug('Found %s' % find)
        pass
    while found:
        found.String = string.replace( found.String, unicode( ↵
        find),unicode(replace))
        found = document.findNext( found.End, search)

#This will be our mini file definition. Here we can specify ↵
which fileds get replaced.
from datetime import datetime
def replace_data(record=None):
    """Key and Value for replacement. Keys are defined by you ↵
    values come from sqlalchemy. Here we define what ↵
    $FirstName in odt file will be replaced with."""
    data={}
    data['$FirstName']=record.FIRSTNAME
    data['$LastName']=record.LASTNAME
    data['$Address']=record.ADDRESS_1
    data['$City']=record.CITY.strip()
    data['$State']=record.STATE
    data['$Zipcode']=record.ZIP_CODE
    data['$TodaysDate']=datetime.now().date()
    data['$User']=record.USER_NO
    data['$CaseID']=str(record.CASE_NUMBER)

```

```

return data

#Lets import page break as we will be doing a loop
from com.sun.star.style.BreakType import PAGE_BEFORE, ←
    PAGE_AFTER

#Lets create a new document (document2 and cursor2) that we ←
    will use to save our replaced pages to.
document2 = desktop.loadComponentFromURL("private:factory/ ←
    swriter", "_blank", 0, ())
cursor2 = document2.Text.createTextCursor()

#Save the document.
document2.storeAsURL("file:///home/lucas/ReminderPages.odt ←
    ", ())
log.info('Created New Document: ReminderPages.odt')

#Selecting a date for my letters. All records created 35 Days ←
    ago.
from datetime import timedelta
days35=timedelta(days=35)
day35=datetime.now().date()-days35
import time
day35a=time.strftime('%Y%m%d',day35.timetuple())

#Select the records from the Database.
allrecords=session.query(AddressbookRecords).filter( ←
    AddressbookRecords.CASE_OPEN_DATE==day35a).order_by( ←
    AddressbookRecords.USER_NO).all()

#Print how many records we have
log.info('We have %s Records from the database.' % len( ←
    allrecords))

#Lets start the loop
log.debug('Starting to Loop Through Records')
for record in allrecords:
    #Load file template
    document = desktop.loadComponentFromURL("file:///home/ ←
        lucas/ReminderPagesTemplate.odt", "_blank", 0, ())
    #We don't need a cursor since we will be searching and ←
        replacing only
    #cursor = document.Text.createTextCursor()

    #Create Search Descriptor
    search = document.createSearchDescriptor()

    #Pass in a sqlalchemy record to our replace_data which ←
        will fill in the values for each field we want. We ←

```

```

will use these values to replace inside the odt.
data=replace_data(record)

#Do a loop of the data and replace the content.We pass in ←
a dictionary data which has keys and values. We find ←
keys and replace them with values.
for find,replace in data.items():
    findandreplace(document,search,unicode(find),unicode( ←
    replace))
    log.debug(str(find)+' '+str(replace))
#Save replaced document as a temporary file.
#Trying to find out how to skip this part but for now we ←
do it.
document.storeAsURL("file:///home/lucas/temp.odt", ())
#Close File
document.dispose()

#We now append our temp file to our document that will ←
hold all pages.
cursor2.gotoEnd(False)
cursor2.BreakType = PAGE_BEFORE
cursor2.insertDocumentFromURL("file:///home/lucas/temp. ←
odt", ())
document2.store()
#2nd copie
#cursor2.gotoEnd(False)
#cursor2.BreakType = PAGE_BEFORE
#cursor2.insertDocumentFromURL("file:///home/lucas/temp. ←
odt", ())
#document2.store()

#Exit
document2.dispose()
log.info('Done.I am exiting')

```

## 4 References

1. <http://www.devshed.com/c/a/Python/Python-and-OpenOfficeorg/>
2. <http://api.openoffice.org/docs/common/ref/com/sun/star/module-ix.html>
3. <http://api.openoffice.org/docs/common/ref/com/sun/star/uno/module-ix.html>