

Incorporando macros Python aos arquivos ODF

Por Júlio César Eiras Melanda

Devido ao suporte ainda pequeno à programação de macros em Python no BrOffice/LibreOffice, muitas pessoas acabam preferindo utilizar o OOBASIC, uma variação da linguagem Basic com funções especiais para trabalhar com documentos BrOffice. Porém, será mostrado agora que criar macros em Python e deixá-las prontas para distribuição, seja num arquivo de template ou num pacote de extensão, é muito mais fácil do que parece.

Esta é uma série de dois artigos, sendo que este trata da criação de macros embutidas nos arquivos e o próximo tratará do empacotamento da macro.

Abra o BrOffice e crie um arquivo odt vazio. Agora, use sua ferramenta de compactação preferida para descompactá-lo. Os arquivos do BrOffice são na verdade conjuntos de arquivos zipados contendo as informações de dados e conteúdo dos arquivos.

Veja como é a estrutura de pastas que você encontrará:

```
Configurations2/
  accelerator/
    current.xml
  floater/
  images/
    Bitmaps/
  menubar/
  popupmenu/
  progressbar/
  statusbar/
  toolbar/
  toolpanel/
META-INF/
  manifest.xml
Thumbnails/
  thumbnail.png
manifest.rdf
mimetype
content.xml
meta.xml
settings.xml
styles.xml
```

Macros com Python

Por Júlio César Eiras Melanda

Cada um destes arquivos e pastas é indispensável para um arquivo de texto do BrOffice, e sua falta no arquivo resultará num arquivo corrompido.

Mas, porque estamos vendo isto?

Porque precisamos alterar elementos da estrutura do arquivo para incorporar a macro em Python.

Vamos começar criando os diretórios necessários que o arquivo ainda não possui. Na raiz do arquivo, onde se encontra o diretório **META-INF**, crie um diretório vazio chamado **Scripts**. Tenha muita atenção à capitalização das letras dos nomes dos diretórios.

Dentro do diretório **Scripts**, crie outro chamado **python**.

Finalmente, dentro deste novo diretório, crie um arquivo de texto em branco chamado *minhamacro.py*. Este arquivo pode ser criado com o seu editor de textos preferido.

Obs.: As suítes de escritório geralmente possuem processadores de texto e não editores de texto, então BrOffice Writer, MS Word, e tantos outros não devem ser usados para criar o arquivo *minhamacro.py*. Este arquivo será o módulo Python que conterá nossa macro.

Abra agora o arquivo *minhamacro.py* e coloque o seguinte conteúdo, respeitando rigorosamente os recuos iniciais de cada linha (identação).

As linhas que iniciam com # são comentários, com exceção da primeira que avisa o interpretador Python qual é a codificação de caracteres que será usada.

Para efeito de simplificação do exemplo, esta macro simplesmente pega nosso arquivo em branco e acrescenta o texto “**Olá esta é minha primeira macro com python!**”.

Você pode observar que nela ajustamos o tamanho das letras para 16 e o tipo de letra para Itálico antes de inserirmos o texto.

Salve e feche o arquivo.

Agora, a estrutura do seu arquivo já está pronta, porém, se recriarmos o arquivo odt compactando o conteúdo destes diretórios, o BrOffice não saberá o que fazer com o novo diretório e considerará o arquivo como corrompido. Assim, precisamos modificar o arquivo manifest.xml, para avisar ao programa que os novos diretórios e arquivos fazem parte do arquivo, e não devem ser considerados um erro.

```
# -*- coding:utf8 -*-
#importa a biblioteca UNO do BrOffice
import uno
def HelloWorld():
    #cria um objeto documento a partir do contexto do documento local
    documento = XSCRIPTCONTEXT.getDocument()
    texto = documento.getText()
    dadosTexto = texto.getEnd()
    dadosTexto.CharHeight = 16.0
    dadosTexto.CharPosture = uno.getConstantByName(
        "com.sun.star.awt.FontSlant.ITALIC")
    dadosTexto.setString('Olá esta é minha primeira macro com python!')
    return None
```

Macros com Python

Por Júlio César Eiras Melanda

Vá então até o diretório META-INF e abra o arquivo *manifest.xml* num editor de textos. Coloque antes da última linha (</manifest:manifest>) o seguinte conteúdo:

```
<manifest:file-entry manifest:media-type="" manifest:full-path="Scripts/python/minhamacro.py"/>
<manifest:file-entry manifest:media-type="application/binary"
manifest:full-path="Scripts/python/" />
<manifest:file-entry manifest:media-type="application/binary"
manifest:full-path="Scripts/" />
```

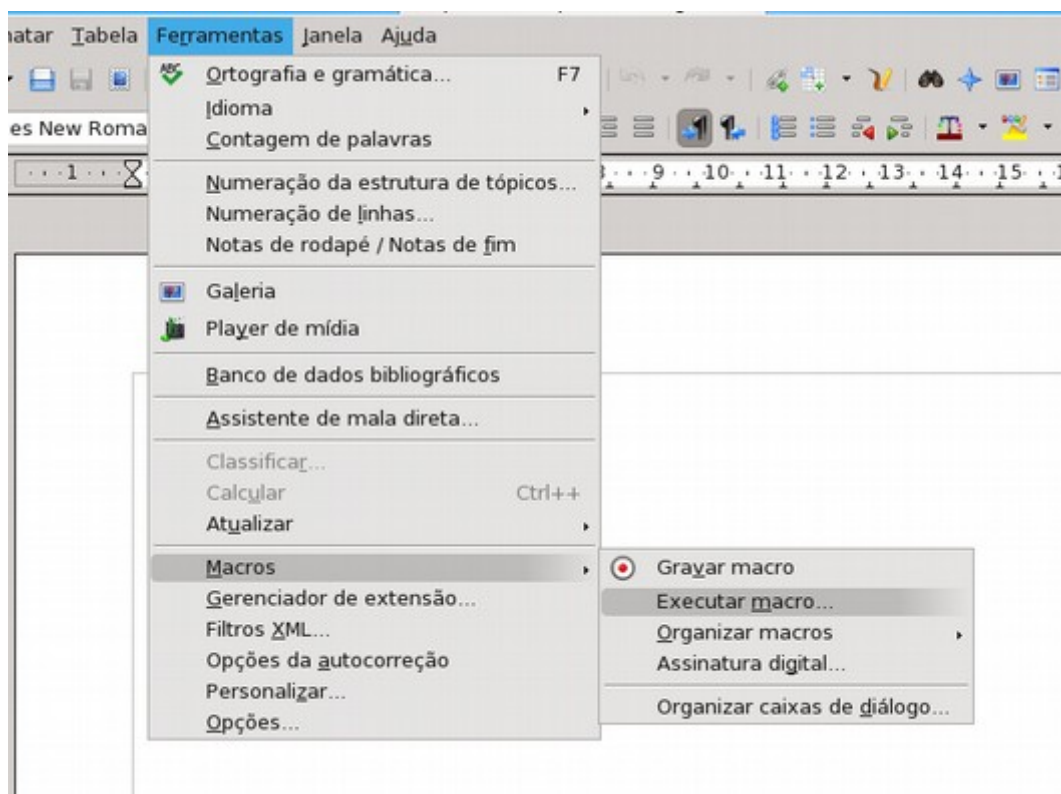
Estas linhas avisam ao BrOffice que temos um script python dentro do arquivo e onde este se encontra.

Antes de gerar o arquivo ODT, verifique se não ficou algum arquivo oculto ou de backup nos diretórios dos arquivos que foram modificados, pois isto poderia gerar arquivos ODT corrompidos.

Agora, navegue de volta para a raiz do arquivo, onde criamos a pasta Script e empacote todos os arquivos e pastas como um único arquivo zip. É importante que seja como zip, não 7zip nem rar ou qualquer outro formato de compactação para funcionar.

Renomeie o arquivo zip gerado, trocando a extensão por odt. Pronto! Agora você já tem seu arquivo ODT com uma macro python embutida!

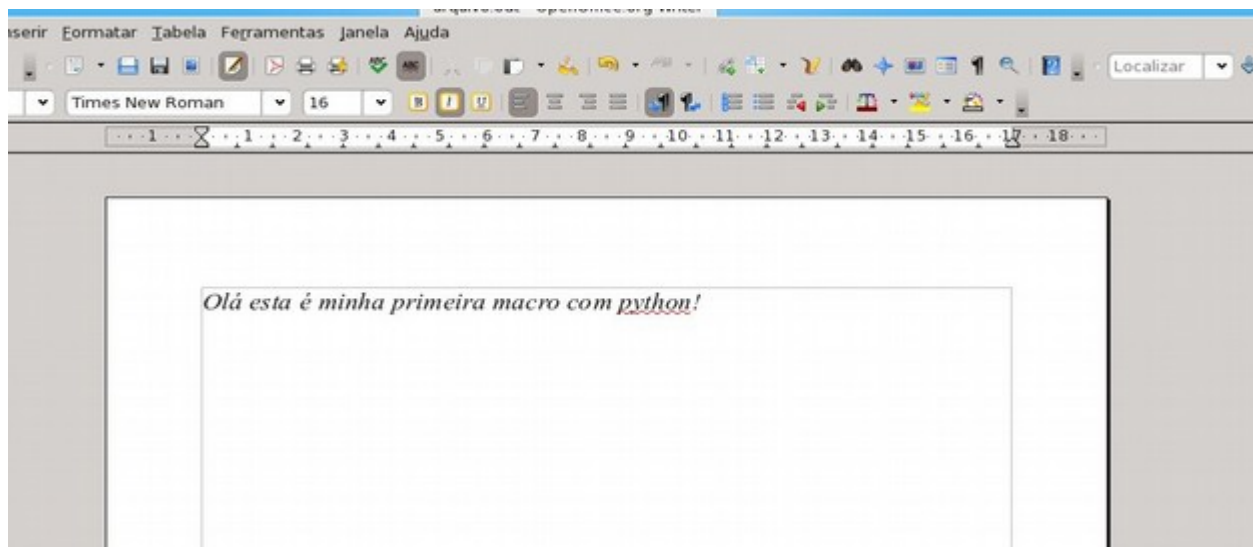
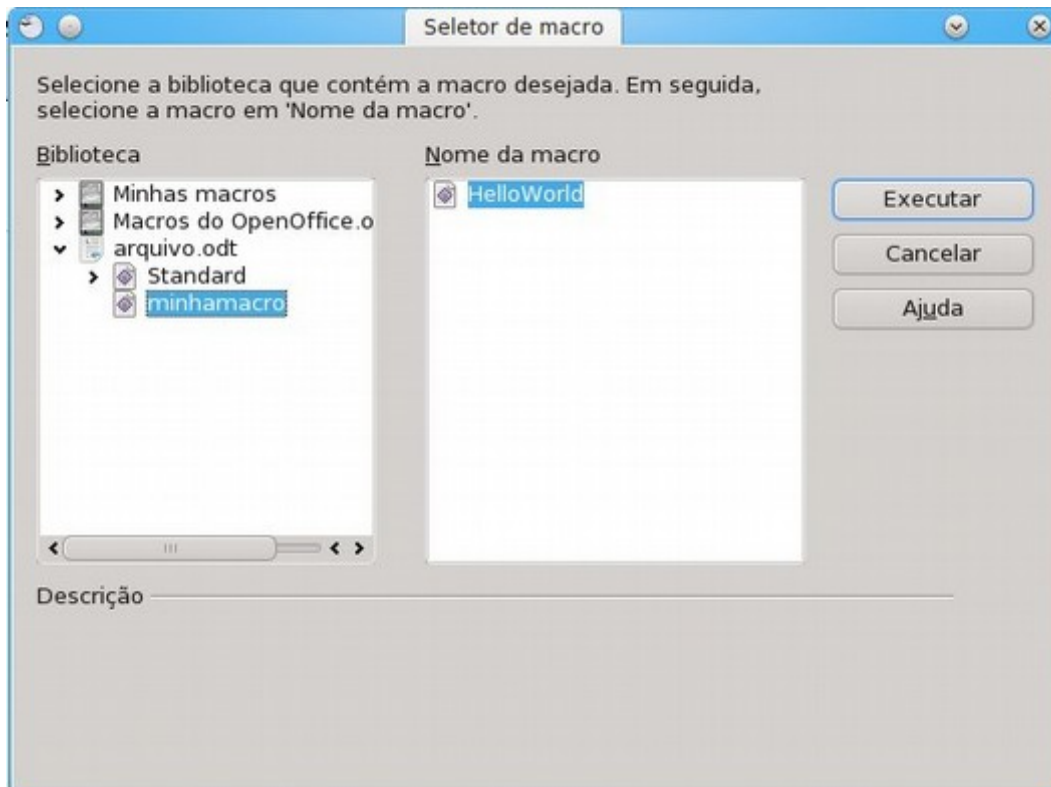
Para testar, abra o arquivo ODT no BrOffice, vá até o menu **Ferramentas**, clique em **Macros** e **Executar macro...**



Macros com Python

Por Júlio César Eiras Melanda

Será aberto um diálogo com duas caixas de seleção. Selecione na primeira à esquerda o seu documento (último item da lista), selecione *minhamacro*. Na outra caixa, selecione *HelloWorld* e clique no botão executar.



No próximo artigo, criaremos um pacote para instalarmos nossa macro como uma extensão para o BrOffice!

Até a próxima! ✓