



Extensões PyUNO

por **Noelson Duarte**

No segundo artigo desta série, escrevemos macros para aumentar e reduzir o tamanho da fonte do texto selecionado no Writer. Hoje, vamos transformar estas macros numa extensão - suplemento - do BrOffice.org.

Um suplemento é um programa interpretado ou compilado com uma maior integração ao aplicativo, principalmente com a interface gráfica. Normalmente, é composto de código, arquivos de configuração e recursos como ícones. Todos estes elementos devem ser empacotados num arquivo zip, que pode ser distribuído para instalação com o Gerenciador de Extensões. Nossa extensão será bem simples, contendo apenas código, ícones e o arquivo de configuração da interface gráfica.

O código fonte

Para transformar as nossas macros numa extensão, faremos algumas alterações no código original.

Note, abaixo, que não definimos a variável `g_exportedScripts` e, as macros nela relacionadas, foram transformadas em classes. Também, todas as referências a `XSCRIPTCONTEXT` foram excluídas, por isto passamos um parâmetro adicional, `oDoc`, para a função `AlteraTamanhoFonte`.

```
# tamanhofonte.py
# extensao PyUNO para o Open / BrOffice.org

# modulos pyuno
import uno
import unohelper
# interfaces uno
from com.sun.star.task import XJobExecutor

def AlteraTamanhoFonte(oDoc, valor):
    oSel = oDoc.getCurrentSelection()
    if (oSel.supportsService("com.sun.star.text.TextRanges")):

for i in range(oSel.getCount()):
    oCur = oSel.getByIndex(i)
    oEnum = oCur.createEnumeration()
    while (oEnum.hasMoreElements()):
        oTxt = oEnum.nextElement()
        if (oTxt.supportsService("com.sun.star.text.Paragraph")):
            oEnum2 = oTxt.createEnumeration()
            while (oEnum2.hasMoreElements()):
                oTxtParte = oEnum2.nextElement()
                tamFonte = oTxtParte.getPropertyValue("CharHeight") + valor
                oTxtParte.setPropertyValue("CharHeight", tamFonte)

# implementa um component uno
class IncrementaTamanhoFonte(unohelper.Base, XJobExecutor):
    """Aumenta o tamanho da fonte em uma unidade"""
    def __init__(self, ctx):
        self.ctx = ctx
```



```
def trigger(self, args):
    smgr = self.ctx.ServiceManager
    desktop = smgr.createInstanceWithContext("com.sun.star.frame.Desktop", self.ctx)
    doc = desktop.getCurrentComponent()
    AlteraTamanhoFonte(doc, 1)

# implementa um component uno
class DecrementaTamanhoFonte(unohelper.Base, XJobExecutor):
    """Diminui o tamanho da fonte em uma unidade"""
    def __init__(self, ctx):
        self.ctx = ctx

    def trigger(self, args):
        smgr = self.ctx.ServiceManager
        desktop = smgr.createInstanceWithContext("com.sun.star.frame.Desktop", self.ctx)
        doc = desktop.getCurrentComponent()
        AlteraTamanhoFonte(doc, -1)

# registra os componentes
g_ImplementationHelper = unohelper.ImplementationHelper()
g_ImplementationHelper.addImplementation(\
    IncrementaTamanhoFonte,
    "org.openoffice.comp.pyuno.nad.Incrementa",
    ("com.sun.star.task.Job",),)

g_ImplementationHelper.addImplementation(\
    DecrementaTamanhoFonte,
    "org.openoffice.comp.pyuno.nad.Decrementa",
    ("com.sun.star.task.Job",),)
```

Inicialmente, importamos os módulos PyUNO e interfaces UNO usadas pelo nosso componente e, a seguir, definimos a já conhecida função `AlteraTamanhoFonte`.

As classes são bem simples, elas possuem apenas dois métodos - *init* e *trigger*. O argumento *ctx* é passado automaticamente para o construtor (método *init*). É o objeto utilizado para obter o `ServiceManager`, responsável pela criação de instâncias de outros componentes.

O método *trigger*, derivado da interface `XJobExecutor`, encarrega-se de disparar o código na ocorrência de algum evento na interface gráfica, por exemplo, um clique sobre um ícone. Ainda, estas classes derivam da classe `unohelper.Base`. Um componente UNO precisa implementar algumas interfaces básicas e esta classe encarrega-se desta tarefa. Finalmente, adicionamos as classes principais ao registro do BrOffice.org, criando a variável estática `g_ImplementationHelper` e chamando o método `addImplementation` para cada uma delas. Este método requer três argumentos: o nome da classe; um identificador único para a implementação, este nome será utilizado para chamar o componente; e, uma tupla com os serviços uno implementados, em nosso caso, apenas `com.sun.star.task.Job`, necessário para a execução da tarefa ao ocorrer o evento. Uma boa regra para criar o identificador único é adotar um esquema de nome qualificado, por exemplo:
`org.openoffice.comp.NomeEmpresa.NomeAplicativo.NomeClasse`

Ainda, evite o uso de caracteres especiais (por ex: acentuados) e não use espaços.

Criando a interface gráfica

O modo como uma extensão deve ser integrada na interface gráfica é definido através do arquivo de configuração `addons.xcu`. Os elementos e atributos configuráveis estão descritos no arquivo XMLSchema `addons.xcs` (não altere), na pasta:

```
<broo_install>/share/registry/schema
```

Além desta fonte, o projeto Programação [1] e o The Developer's Guide [2] contém informações e exemplos sobre estas configurações.

Os ícones

Ícones para os suplementos do BrOffice.org são arquivos bmp, tamanhos grande (26x26) e pequeno (16x16) e contrastes normal e alto. Podem, ainda, ser codificados em sua forma



hexadecimal no próprio arquivo de configuração. Utilizaremos ícones de contraste normal, nos dois tamanhos, ambos com 256 cores. Para facilitar, serão nomeados com um sufixo indicando o seu tamanho e contraste. Eis uma descrição:

Tamanho (pixels)	Contraste	Arquivo	Classe
16x16	normal	incr_16.bmp	IncrementaTamanhoFonte
26x26	normal	incr_26.bmp	IncrementaTamanhoFonte
16x16	normal	decr_16.bmp	DecrementaTamanhoFonte
26x26	normal	decr_26.bmp	DecrementaTamanhoFonte

O arquivo de configuração

Em nosso exemplo, criaremos uma opção de menu, com submenus, em *Ferramentas | Suplementos* e uma barra de ferramentas com seus próprios botões. Desejamos que a extensão esteja ativa apenas no módulo Writer. O arquivo XML *addons.xcu* deve ser codificado como UTF-8 e os nomes dos elementos indicativos das opções de menu e dos botões na barra de ferramentas devem ser únicos. Eis o arquivo:

```
<?xml version="1.0" encoding="UTF-8"?>
<oor:node xmlns:oor="http://openoffice.org/2001/registry"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  oor:name="Addons" oor:package="org.openoffice.Office">
<node oor:name="AddonUI">
<node oor:name="AddonMenu">
<node oor:name="nad.tfonte.m00" oor:op="replace">
<prop oor:name="Context" oor:type="xs:string">
  <value>com.sun.star.text.TextDocument</value>
</prop>
<prop oor:name="URL" oor:type="xs:string">
</prop>
<prop oor:name="Target" oor:type="xs:string">
  <value>_self</value>
</prop>
<node oor:name="Submenu">
<node oor:name="nad.tfonte.m01" oor:op="replace">
  <prop oor:name="Title" oor:type="xs:string">
    <value xml:lang="en-US">Increase</value>
    <value xml:lang="pt-BR">Aumentar</value>
  </prop>
  <prop oor:name="URL" oor:type="xs:string">
    <value>service:org.openoffice.comp.pyuno.nad.Incrementa?insert</value>
  </prop>
  <prop oor:name="Target" oor:type="xs:string">
    <value>_self</value>
  </prop>
  <prop oor:name="ImageIdentifier" oor:type="xs:string">
    <value>%origin%/icones/incr</value>
  </prop>
</node>
<node oor:name="nad.tfonte.m02" oor:op="replace">
  <prop oor:name="Title" oor:type="xs:string">
    <value xml:lang="en-US">Decrease</value>
    <value xml:lang="pt-BR">Diminuir</value>
  </prop>
  <prop oor:name="URL" oor:type="xs:string">
    <value>service:org.openoffice.comp.pyuno.nad.Decrementa?insert</value>
  </prop>
  <prop oor:name="Target" oor:type="xs:string">
    <value>_self</value>
  </prop>
  <prop oor:name="ImageIdentifier" oor:type="xs:string">
    <value>%origin%/icones/decr</value>
  </prop>
</node>
</node>
</node>
</node>
</node>
```



```

<node oor:name="OfficeToolBar">
<node oor:name="nad.tfonte.tb00" oor:op="replace">
<node oor:name="nad.tfonte.btn01" oor:op="replace">
  <prop oor:name="Context" oor:type="xs:string">
    <value>com.sun.star.text.TextDocument</value>
  </prop>
  <prop oor:name="Title" oor:type="xs:string">
    <value xml:lang="en-US">Increase</value>
    <value xml:lang="pt-BR">Aumentar</value>
  </prop>
  <prop oor:name="URL" oor:type="xs:string">
    <value>service:org.openoffice.comp.pyuno.nad.Incrementa?insert</value>
  </prop>
  <prop oor:name="Target" oor:type="xs:string">
    <value>_self</value>
  </prop>
  <prop oor:name="ImageIdentifier" oor:type="xs:string">
    <value>%origin%/icones/incr</value>
  </prop>
</node>
<node oor:name="nad.tfonte.btn02" oor:op="replace">
<prop oor:name="Context" oor:type="xs:string">
  <value>com.sun.star.text.TextDocument</value>
</prop>
<prop oor:name="Title" oor:type="xs:string">
  <value xml:lang="en-US">Decrease</value>
  <value xml:lang="pt-BR">Diminuir</value>
</prop>
<prop oor:name="URL" oor:type="xs:string">
  <value>service:org.openoffice.comp.pyuno.nad.Decrementa?insert</value>
</prop>
  <prop oor:name="Target" oor:type="xs:string">
    <value>_self</value>
  </prop>
  <prop oor:name="ImageIdentifier" oor:type="xs:string">
    <value>%origin%/icones/decr</value>
  </prop>
</node>
</node>
</node>
</oor:node>

```

Agora, há uma breve descrição dos principais elementos:

AddonUI: é o elemento raiz.

AddonMenu: indica uma opção de menu em *Ferramentas* | *Suplementos*, com ou sem um comando associado;

Submenu: representa um submenu subordinado a *AddonMenu*. Aqui, cada elemento *nad.tfonte.m0x*, refere-se a uma opção no submenu.

OfficeToolBar: indica a existência de uma barra de ferramentas *nad.tfonte.tb00*, com seus próprios botões de comando *nad.tfonte.btn0x*;

Context: indica os módulos do aplicativo nos quais a interface estará ativa. Para mais de um contexto, separe os valores da tabela abaixo com uma vírgula.

Módulo	Valor
Writer	com.sun.star.text.TextDocument
Calc	com.sun.star.sheet.SpreadsheetDocument
Impress	com.sun.star.presentation.PresentationDocument
Draw	com.sun.star.drawing.DrawingDocument
Editor de Fórmulas	com.sun.star.formula.FormulaProperties

Title: é o texto da opção de menu ou a dica do botão da barra de ferramenta, note que podemos definir textos para diversos idiomas;

ImageIdentifier: associa uma imagem privada ou definida pelo usuário ao comando. Note que omitimos a parte final – sufixo e extensão – do nome do arquivo e que os ícones estão numa pasta *icones*. A variável *%origin%* refere-se ao caminho onde a extensão será instalada.

URL: é o comando a ser executado quando o elemento a que se refere for selecionado. É formada pelo qualificador *service*: seguido do nome da implementação (como registrado no BrOffice.org), e um argumento *?insert* para o método trigger (o argumento pode ser qualquer cadeia de caracteres).

Criando o pacote

Um suplemento é um arquivo zip, contendo todos os componentes e recursos. Então vamos criar o arquivo *tfonte.zip*, com a estrutura ao lado:

```
tamanhofonte.py
addons.xcu
/icones
  incr_16.bmp
  incr_26.bmp
  decr_16.bmp
  decr_26.bmp
```

Instalando a extensão

A instalação pode ser feita para um usuário – Minhas Extensões – ou para todos os usuários – Extensões do BrOffice.org.

No primeiro caso, selecione Ferramentas | Gerenciador de Extensões, comande Adicionar, selecione o arquivo *tfonte.zip* e comande Abrir.

No segundo caso, rode o Gerenciador de Extensões como administrador, a partir da linha de comando:

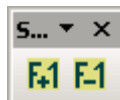
```
<broo_install>/program/unopkg gui
```

Selecione Extensões do Broffice.org, comande Adicionar, selecione o arquivo *tfonte.zip* e comande Abrir.

Após a instalação, atualize a interface gráfica criando um novo arquivo e você verá uma nova entrada sob o menu Ferramentas.



E, também, uma nova barra de ferramentas:



Conclusão

Nos três artigos desta série, apresentamos os passos básicos para o desenvolvimento de macros e suplementos do BrOffice.org, com a linguagem Python.

Nosso exemplo, além de simples, é uma extensão “à moda antiga”. Os desenvolvedores do OpenOffice.org ampliam, a cada versão, o conceito de extensão. Recursos como reconhecimento, atualização, dependência de versão e licenciamento estão disponíveis. Apesar de nenhum deles ter sido apresentado, uma boa fonte de informações é o The Developer's Guide [2] e os aplicativos no projeto Extensões do OpenOffice.org [3].

Finalmente, por questão de espaço, removi a macro *DialogoAlterar* do suplemento. Deixei como um exercício “de fim de curso” a transformação desta macro num componente, o desenho dos ícones e a sua inclusão no arquivo de configuração. ✈

Referências

- [1] http://www.openoffice.org.br/files/addons1_1_br.zip
- [2] <http://api.openoffice.org/DevelopersGuide/DevelopersGuide.html>
- [3] <http://extensions.services.openoffice.org/project>