



Creating an OpenOffice.org Extension

Adding On

Dmitri shows how to create an OpenOffice.org extension from scratch and introduces a useful extension packaging tool. *By Dmitri Popov*

So, you've written a nifty OpenOffice.org macro and want to share it with the world. You can, of course, publish the code on your blog or email it to your colleagues. But if you want to make it easier for people to install and use your macro, the best way is to turn it into an OpenOffice.org extension. Packaging your code as an extension requires some work, but a couple of tools can make the process less laborious. This tutorial will guide you through the process of creating an OpenOffice.org extension and introduce a useful extension packaging tool along with couple of nifty tricks.

Starting from Scratch

For this project, we'll use a simple macro that inserts dummy text into the currently opened OpenOffice.org Writer document (Listing 1).

If you're wondering how this macro works, here is a brief explanation. To begin, the macro uses the `CreateUnoService("com.sun.star.util.PathSubstitution")` service to determine the path to the OpenOffice.org user directory and locate the `dummy.txt` plaintext file (you have to create the file manually and put

it in the OpenOffice.org user directory). Then, the macro prompts the user to enter the number of paragraphs it should insert into the document. Finally, the macro opens `dummy.txt` for reading, and inserts text from it into the current Writer document.

Before you add the macro to OpenOffice.org, you should create a separate library for it so you can easily make an extension out of it later. To create a new library, first choose *Tools | Macros | Organize Macro | OpenOffice.org Basic*. Second, press the *Organizer* button, switch to the *Libraries* section, and press *New*. Next, give the new library a name (e.g., *DummyText*) and press *OK*. Finally, hit the *Edit* button to open the library for editing and enter the macro code in the empty module, *Module1*. Now you are ready to build an extension.

Creating an Extension

An OpenOffice.org extension is just a plain ZIP archive that contains a macro library and a few XML-based files. Theoretically, it's possible to create the required XML files in a text editor and pack the extension manually, but why bother when the `BasicAddonBuilder` [1] can do the heavy lifting for you? Using `BasicAddonBuilder`, you can create an extension by simply following the extension builder wizard. Because `BasicAddonBuilder` itself is just a regular OpenOffice.org extension, you can use it without leaving the convenience of your favorite productivity suite. Once installed, `BasicAddonBuilder` adds a new toolbar with a single button in it. Press that to open the wizard, which will guide you through the entire process.

To start, choose the library containing the macro you want to add to the extension (in this case, it's *DummyText*) (Figure 1). Next, you need to specify the way the extension integrates into OpenOffice.org. You have several options here: Add a customized toolbar, add a top-level menu, add items under *Tools | Add-ons*, or add items under the Help menu. Usually, a separate toolbar and a top-level menu are perfect for extensions containing multiple macros (and thus several menu items). *Tools | Add-Ons* or the Help menu works best for extensions with one or two macros.

For this example, you'll want a top-level menu. Tick the *Add top-level menu*

DMITRI POPOV

Dmitri Popov holds a degree in Russian language and computer linguistics. He has been writing exclusively about Linux and open source software for several years, and his articles have appeared in Danish, British, North American, German, and Russian magazines and websites.

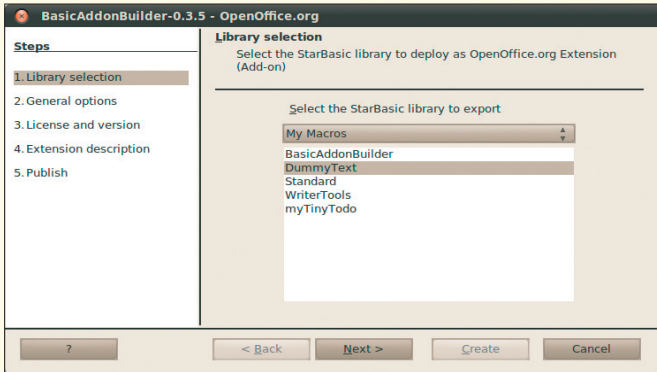


Figure 1: BasicAddonBuilder simplifies the process of creating an extension.

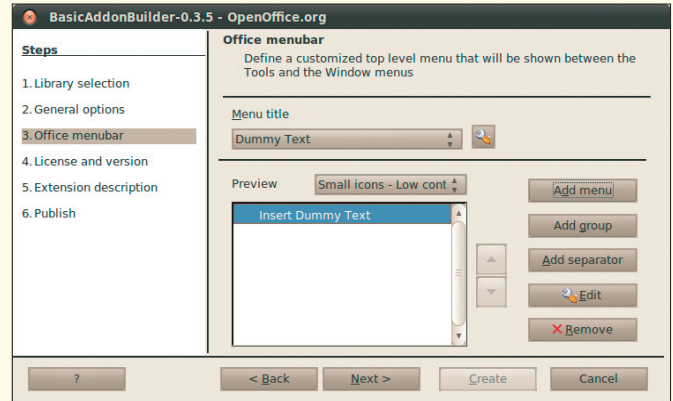


Figure 2: Adding menu entries.

option and press the *Next* button to move to the next window, where you build the menu structure of the extension. To create a top-level menu, first press the *Edit* button next to the *Menu title* field (Figure 2). In the dialog window that appears, enter the desired menu title in the *Item label* field. Also, you have to specify a so-called Office context, or OpenOffice.org applications in which the top-level menu should be visible.

The Dummy Text extension is designed to work with Writer documents, so tick the *Writer* checkbox. This ensures that the specified top-level menu is available only in OpenOffice.org Writer. BasicAddonBuilder provides support for multiple languages, so you can specify the menu title label in different languages. To do this, press the *More Languages* button next to the *Item label* field and add entries for different languages. When the user installs the extension, the system will automatically pick the appropriate language depending on the locale settings of OpenOffice.org.

Next, you need to specify a menu item that launches the InsertDummyText macro. To do this, press the *Add menu* button. This opens a dialog window similar to the previous one, but it has a few more options to specify. Besides the menu item label,

LISTING 1: Insert Dummy Text Macro

```

01 Sub InsertDummyText()
02 Dim ParaNo as Integer
03 ThisDoc=ThisComponent
04 ThisText=ThisDoc.Text
05 TheCursor=ThisText.createTextCursor
06 On Error GoTo ErrHndl
07 SubstService = CreateUnoService("com.sun.star.util.PathSubstitution")
08 UserPath = SubstService.substituteVariables("${user}", true)
09 DummyTxt = UserPath + "/dummy.txt"
10 DummyTxtDir=Right(UserPath, Len(UserPath)-7)
11
12 f1 = FreeFile()
13 Open DummyTxt for Input as #f1
14 ParaNo=InputBox("Number of Paragraphs:", "Input")
15 ParaCounter=0
16 Do while ParaCounter<ParaNo
17     Line Input #f1, s
18     TheCursor.String=s+Chr(13)+Chr(13)
19     TheCursor.collapseToEnd
20     ParaCounter=ParaCounter+1
21 Loop
22 Close #f1
23 Exit Sub
24 ErrHndl:
25 MsgBox("Copy the dummy.txt file supplied with this extension to "
        +DummyTxtDir, 16, "Error")
26 End Sub
    
```

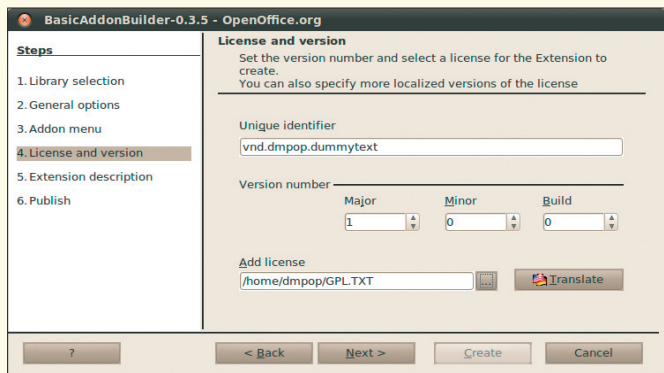


Figure 3: Specifying the extension's unique identifier and version number.

you must link the item to a specific macro (in this case, Insert-DummyText). Press the button next to the *Command* field, select the appropriate library from the drop-down list, choose the desired module in the left pane, and then pick the macro you want in the right pane.

In the *Icons* section, you can add icons to the menu item. To do this, click in the empty icon square and select the desired icon of the right size. Repeat this step for other icon sizes, if necessary. Here, too, you can use the *More Languages* button to specify the name of the menu entry in other languages. Next, tick the *Writer* checkbox in the *Office context* section and press *OK*.

Hit *Next* to move to the *License and version* section, where you should specify a unique identifier for your extension, a version number, and a license (Figure 3). In most cases, the de-

fault identifier generated by BasicAddOnBuilder will do just fine, but you are free to replace it with your own string.

Finally, the *Publish* section lets you specify a local directory in which BasicAddOnBuilder saves the created extension (Figure 4). Additionally, you can configure BasicAddOnBuilder to push the created extension to the specified FTP server. To make this feature work, you must configure FTP connection settings by pressing the *Configure* button next to the *FTP address* field to open the *FTP connection settings* dialog window. Fill out the required fields and press the *Connect* button to test the connection. If everything works properly, press the *OK* button to save the settings and close the window.

OpenOffice.org can automatically update installed extensions by polling a special XML update file containing the latest version number at the link to the most recent *.oxt* file. If the version number in the update file is greater than the version number of the installed extension, OpenOffice.org will automatically grab the latest *.oxt* package and install it. This handy mechanism ensures that the user always runs the latest version of the extension and completely automates the update process. If you want to enable this feature in your extension, tick the *Add auto-update information* checkbox in BasicAddOnBuilder's *Publish* section and enter the path that OpenOffice.org will use to fetch the latest version of the extension.

Finally, hit the *Create* button and, once BasicAddOnBuilder has finished its job, you will find a freshly baked *.oxt* package in the specified directory (and on the FTP server, provided you enabled and configured the FTP feature). If you enabled the auto-update feature, you'll need to do one more thing before you can release your extension into the wild.

Although BasicAddOnBuilder adds the required update information to the extension, it doesn't generate an update file, so you have to create it manually. To do that, use your archive manager of choice to extract the *description.xml* file, then open it in a text editor and locate the following line (where *EXTENSION_URL* is the actual download URL and *EXTENSION_ID* is the extension's unique identifier):

```
<update-information>
  <src xlink:href="http://EXTENSION_URL/.EXTENSION_ID_update.xml" />
</update-information>
```

Note the *EXTENSION_URL* and *EXTENSION_ID* values, create a new text file, and insert the XML code from Listing 2 in it.

Replace *EXTENSION_ID* with your extension's identifier and *DOWNLOAD_URL* with the direct link to the *.oxt* file (e.g., <http://dmpop.homelinux.com/DummyText/DummyText.oxt>). Save the file as *.EXTENSION_ID_update.xml* (don't forget to replace the *EXTENSION_ID* placeholder with the extension's identifier) and upload it to the *EXTENSION_URL* location.

Next time you have to push an update, upload a new version of the extension to the download URL specified in the update file. Then open the update file in a text editor and replace the *version* value with the new version number (e.g., `<version value="1.1.0" />`). Upload the modified file to the specified location, and you are done.

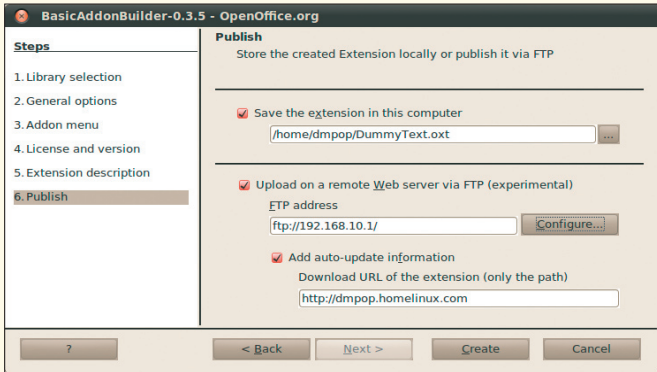


Figure 4: Configuring publishing settings.

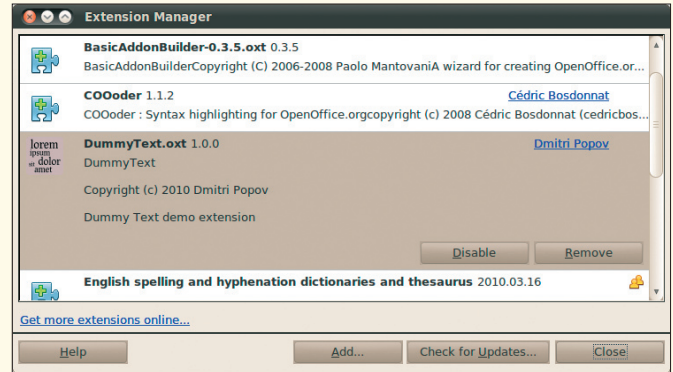


Figure 5: The finished extension in all its beauty.

To spice up your extension, you might want to add an icon. To do this, create a 42x42-pixel icon in your favorite graphics editor and save the result in PNG format. Open the `.oxt` package with an archive manager and add the icon to it. Extract the `description.xml` file, open it in a text editor, and add the following code above the `<update-information>` node, replacing the `ICON` placeholder with the actual file name:

```
<icon>
  <default xlink:href="ICON.png" />
  <high-contrast xlink:href="ICON.png" />
</icon>
```

While you are at it, you might want to specify the publisher property, which displays the name of the extension's creators linked to a URL of your choice. To do this, add the following code to the `description.xml` file:

```
<publisher>
  <name xlink:href="http://dmpop.homelinux.com" lang="en">Dmitri Popov</name>
</publisher>
```

Save the modified `description.xml` file and add it to the `.oxt` package.

Now you are ready to install and test the extension, but before you do that, remove the `DummyText` library; otherwise, it will conflict with the extension, preventing it from working properly. Once you've done that, install the extension with the `Tools | Extension Manager` command (Figure 5); restart OpenOffice.org, and you should see the `Dummy Text` menu in the main menubar.

Final Word

With the `BasicAddOnBuilder` tool and some fine-tuning, you can turn your OpenOffice.org Basic macros into neatly packaged extensions. Of course, the best way to learn the intricacies is to deconstruct an existing package, so feel free to grab the `Dummy Text` demo extension [2] and examine it. Note that this version of the extension adds the `Insert Dummy Text` command to `Tools | Add-Ons`. ■■■

LISTING 2: XML Code

```
01 <?xml version='1.0' encoding='UTF-8'?>
02 <description
03   xmlns="http://openoffice.org/extensions/update/2006"
04   xmlns:xlink="http://www.w3.org/1999/xlink">
05   <identifier value="EXTENSION_ID" />
06   <version value="1.0.0" />
07   <update-download>
08     <src xlink:href="http://DOWNLOAD_URL" />
09   </update-download>
10 </description>
```

INFO

- [1] `BasicAddOnBuilder`:
<http://extensions.services.openoffice.org/en/project/BasicAddOnBuilder>
- [2] `Dummy Text` demo extension:
<http://code.google.com/p/writertools/downloads/detail?name=DummyText.zip>