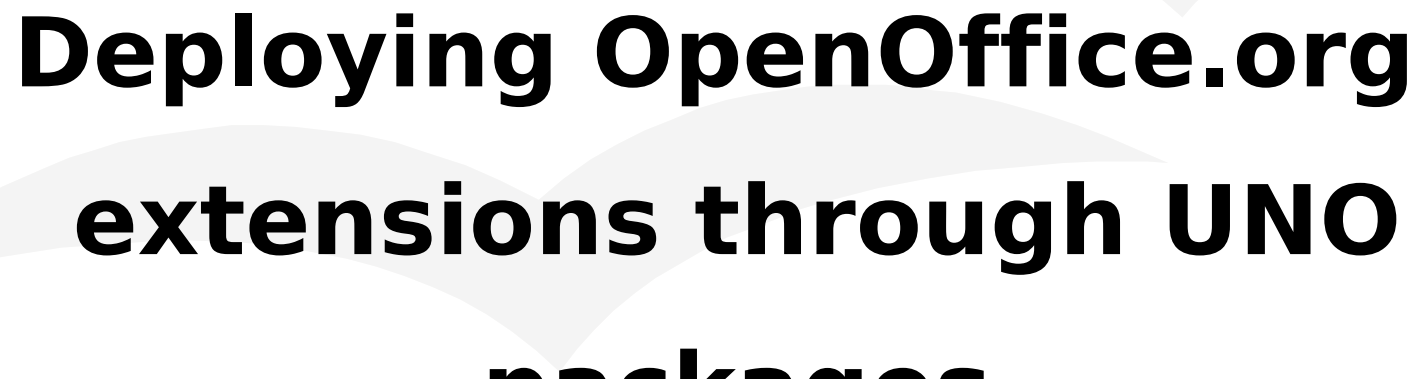




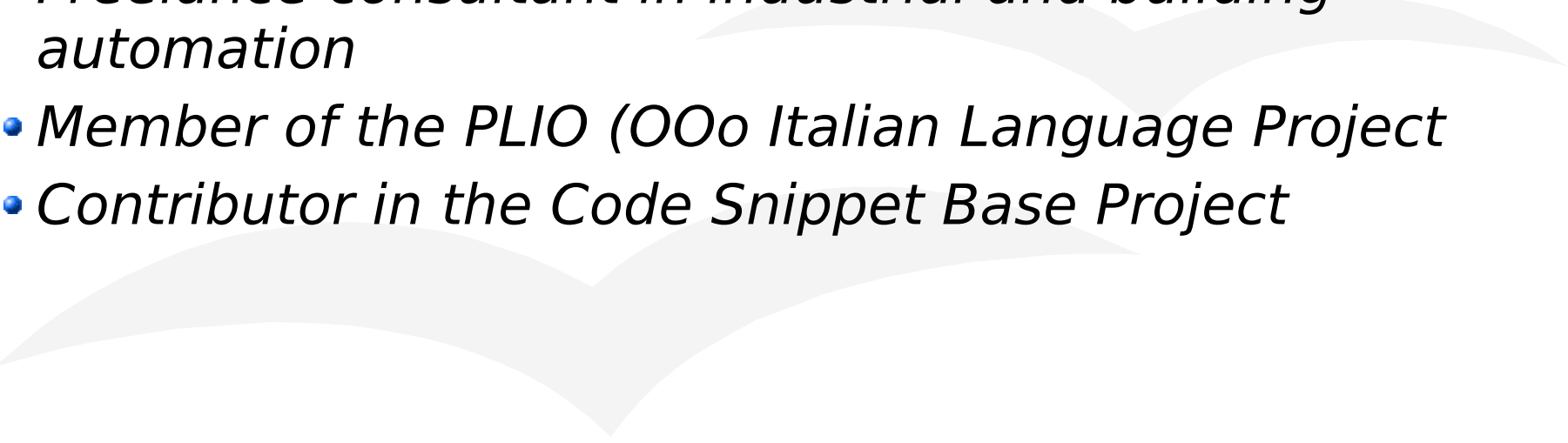
OOoCon 2005



**Deploying OpenOffice.org
extensions through UNO
packages**



Speaker introduction

- I'am Paolo Mantovani
 - *Electrical engineer*
 - *Freelance consultant in industrial and building automation*
 - *Member of the PLIO (OOo Italian Language Project)*
 - *Contributor in the Code Snippet Base Project*
- 



Agenda

- Getting started
- Features
- Anatomy
- A first UNO package
- The Addons.xcu file
- UNO package integrations
- Common problems
- Background: the configuration registry
- Registry Component Schema (xcu)
- Registry update (xcu)
- Accessing to the registry via API
- Macrofied URL's
- Macrofied URL's (example)
- Useful links & resources
- Q & A



Getting started

UNO Packages are zip archives (*.uno.zip)

The UNO Packages mechanism includes:

A command line tool

`<office_path>/program/unopkg`

A dialog window for package management

Special folders in the default setup

Special configuration registry areas



Features

What can be deployed with UNO packages?

- Basic and Dialog Libraries
- Arbitrary UNO components
- Calc Add-Ins
- Chart Add-Ins
- New database drivers
- Other....

Anatomy

Uno Package Bundle

SnippetCreator.zip

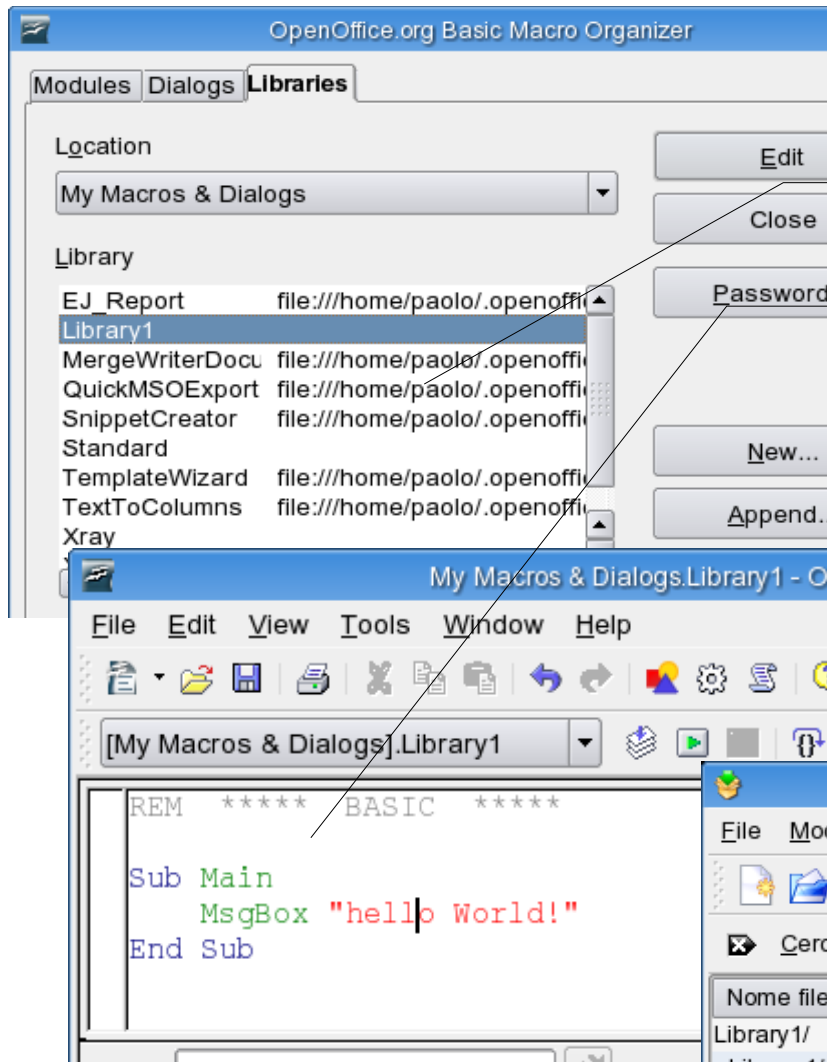
Implementation

Arbitrary files

Configuration files

- SnippetCreator
- images
 - snippetcreator_26.png
 - top.bmp
 - end.bmp
- icons
- help
 - it
 - default
- README
- INSTALL
- COPYING
- Addons.xcu
- AddonConfiguration.xcu
- AddonConfiguration.xcs

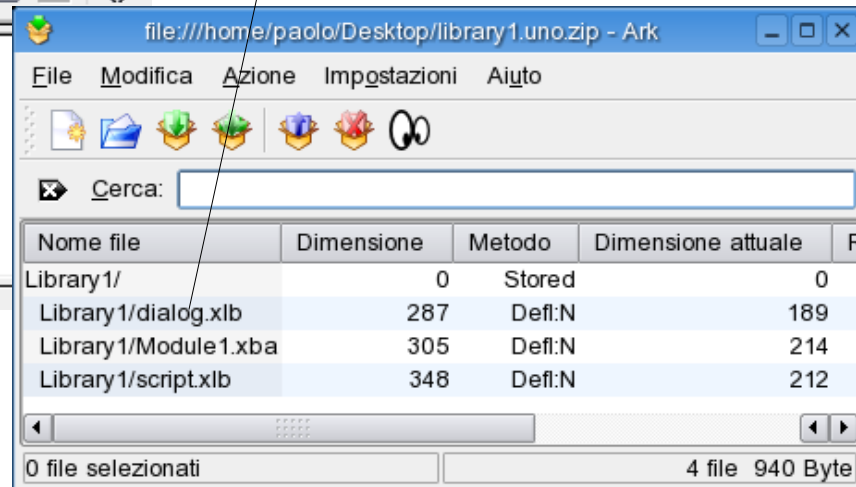
A first UNO package



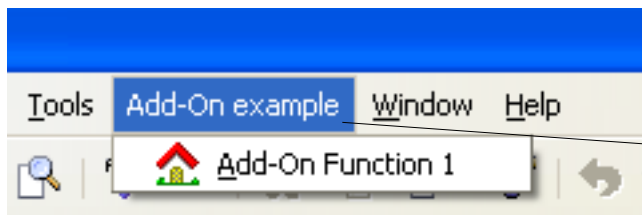
Create a new StarBASIC library

Write some code

Zip everything

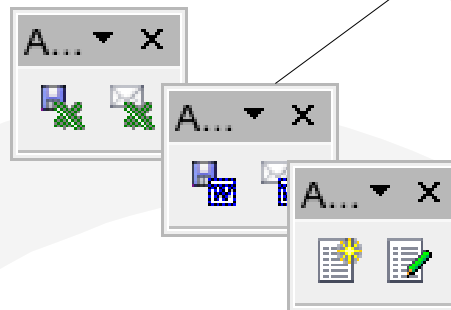


The Addons.xcu file

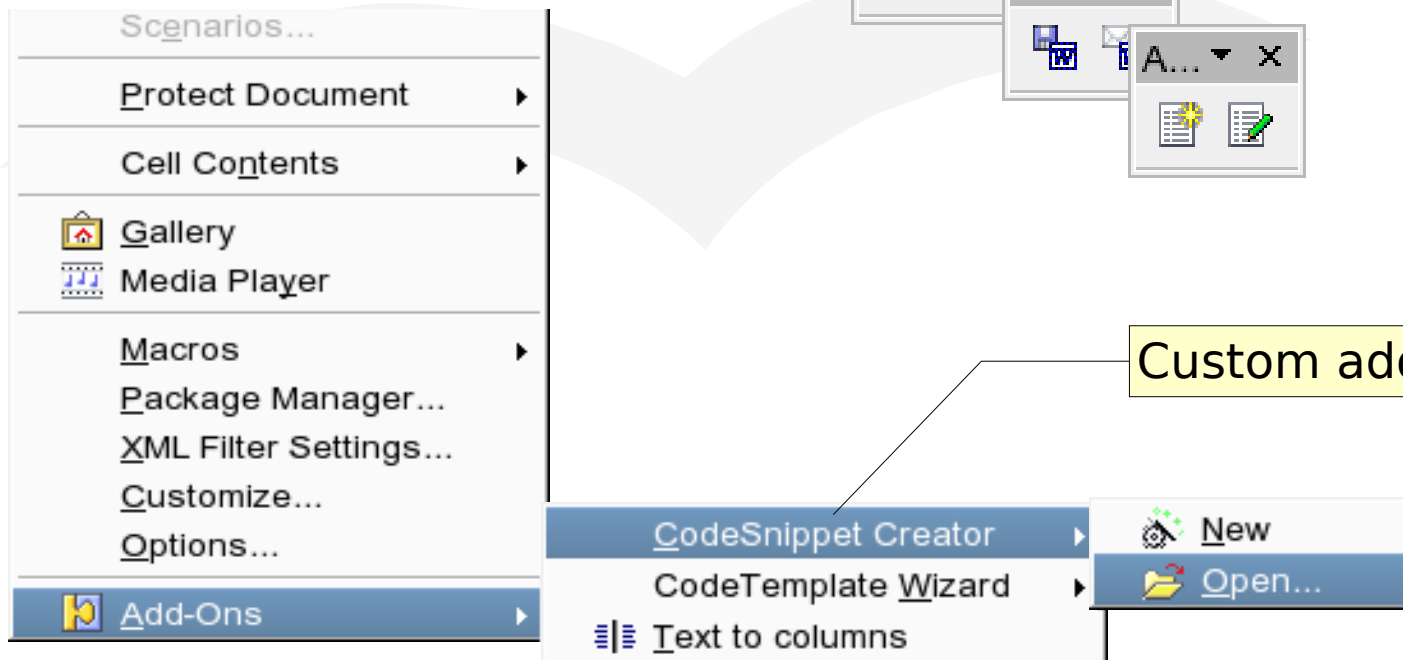


Custom top level menu

Custom toolbars



Custom addon sub menu



UNO package integrations

UI integration



→ Addons.xcu

Automatic execution



→ Jobs.xcu

Dispatch framework



→ ProtocolHandler.xcu

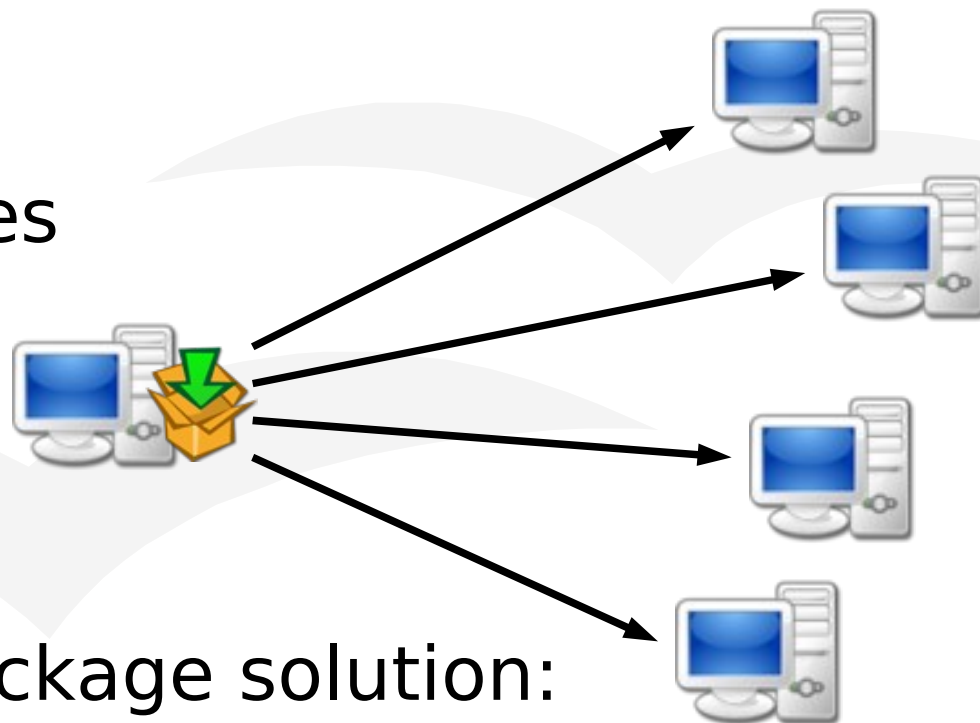
Disabling commands



→ Commands.xcu

Common problems

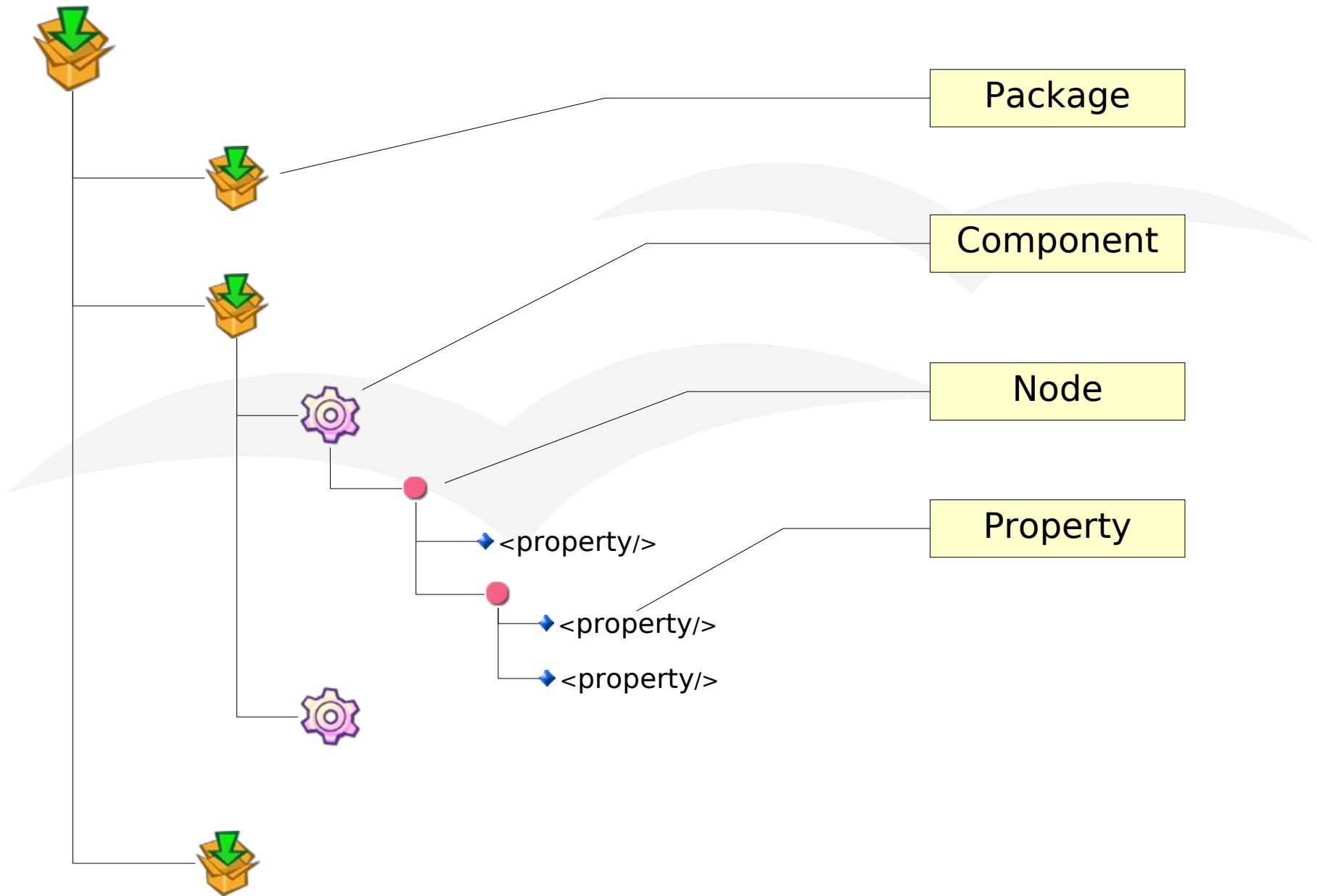
- File resources
- Localization
- User preferences



UNO package solution:

- *High abstraction level*
- *Language agnostic*
- *Multi-platform*

Background: the configuration registry



Registry Component Schema (xcu)

Component name

Package name

```
<?xml version='1.0' encoding='UTF-8'?>

<oor:component-schema
  oor:name="myconfig"
  oor:package="org.openoffice.comp.mypackage"
  xmlns:oor="http://openoffice.org/2001/registry">

<!-- <import/>-->

<templates/>

<component>
  <prop oor:name="ProductName" oor:type="xs:string"/>
  <prop oor:name="ImageLabel" oor:type="xs:string" oor:localized="true"/>
  <prop oor:name="ImageUrl" oor:type="xs:string"/>
</component>

</oor:component-schema>
```

Registry update (xcu)

```
<?xml version='1.0' encoding='UTF-8'?>

<oor:component-data
  oor:name="myconfig"
  oor:package="org.openoffice.comp.mypackage"
  xmlns:oor="http://openoffice.org/2001/registry">

  <prop oor:name="ProductName">
    <value>My First UNO package</value>
  </prop>

  <prop oor:name="ImageLabel">
    <value xml:lang="en-US">This is an image</value>
    <value xml:lang="it">Questa è un'immagine</value>
  </prop>

  <prop oor:name="ImageUrl">
    <value>%origin%/images/MyImage.bmp</value>
  </prop>

</oor:component-data>
```

%origin% variable is used for referencing files inside the package bundle

Accessing to the registry via API

```
'for other languages than StarBasic, you will obtain the configuration  
'Provider service from the local context service manager
```

```
oConfigProvider =  
    CreateUnoService("com.sun.star.configuration.ConfigurationProvider")
```

```
Dim aNodePath(0) As New com.sun.star.beans.PropertyValue  
    aNodePath(0).Name = "nodepath"  
    aNodePath(0).Value = "org.openoffice.mypackage.myconfig"
```

```
'obtaining an read-only access
```

```
oRegKey = oConfigProvider.CreateInstanceWithArguments( _  
    "com.sun.star.configuration.ConfigurationAccess", _  
    aNodePath() )
```

```
'retrieving values:
```

```
Print oRegKey.GetByName("ImageLabel")
```



Macrofied URL's

Management of file resources is a 2 steps process:

1. UNOPKG resolves the **%origin%** placeholder:

vnd.sun.star.expand:**\$UNO_USER_PACKAGES_CACHE**/uno_packages/...

2. The MacroExpander service replaces

\$UNO_PACKAGES_CACHE

with the actual cache URL

Macrofied URL's (example)

```
'cut the "vnd.sun.star.expand:" protocol from the macrofied Url retrieved
'from the registry
sImageUrl = Right(sImageUrl, Len(sImageUrl)-20)

'get the default context
oCtx = getProcessServiceManager().getDefaultContext

'get the macro expander from the default context
oMacroExpander = _
oCtx.getValueByName("/singletons/com.sun.star.util.theMacroExpander")

'expand the macrofield expression
sImageUrl = oMacroExpander.ExpandMacros(sImageUrl)

' At this point you should de-escape the resulting URL
' but OoBasic does not offer a function for that.
' Anyway this is a small problem if you avoid accented chars or spaces
' in filenames you use inside your package

Print sImageUrl      'this is the absolute Url of your image
```




Useful links & resources

Official documentation

- *Developer's Guide – 4 Writing UNO Components*
- *Utilities Project - Documentation - Registry*

From <http://oomacros.org>

- *Add On Tool, a nice document from Bernard Marcellly*
- *Various UNO packages*

Other:

- *Kohei Yoshida has nice uno packages!*
<http://kohei.us/index.html>
- *Thessalonica by Alexej Kryukov*
<http://www.thessalonica.org.ru>
- *SnippetCreator by ...me*
<http://api.openoffice.org/SDK/snippets/>



Q & A

- Do you have any questions?