CREATIVITY UNLOCKED: HOW TO MAKE LIBREOFFICE UI ELEMENTS MOVE

CIB.de

Aarhus 2015
CONFERENCE
23rd-25th September

Target audience

- New (UI) hackers
- Core hackers with little UI hacking experience

What will we learn

- Add UNO command
- Add toolbar/sidebar button, menu entry
- … with an icon :)
- Define slot and its interface
- Connect it all together

UNO command

- basic unit of dispatch API
- of a form **.uno:CommandName**
  - e.g. .uno:Print, .uno:CharColor
- configuration (XML) files for commands: *officecfg/registry/data/org/openoffice/Office/UI/*Commands.xcu*
  - common to all modules: *GenericCommands.xcu*
  - app-specific: e.g *WriterCommands.xcu*, *DrawImpressCommands.xcu*

Example UNO command

```
<node oor:name="UserInterface">
  <node oor:name="Commands">
    <node oor:name=".uno:DoSomething" oor:op="replace">
    <prop oor:name="Label" oor:type="xs:string">
         <value xml:lang="en-US">Tooltip text</value>
    </prop>
    <prop oor:name="Properties" oor:type="xs:int">
         <value>1</value>
    </prop>
   </node>
```

Tooltip text

Flags:
Icon/No icon?
CTL sensitive?

UI element

- entry point to executing the command
- toolbar/sidebar button, (context) menu entry

Toolbars

- toolbars (XML) configuration in *$app/uiconfig/$app2/toolbar*
- separate XML files for context-dependent toolbars
- example:

```
<toolbar:toolbaritem xlink:href=".uno:DoSomething"
toolbar:visible="true|false" toolbar:helpid="42"

toolbar:style=dropdown/>
```

Visible by default?

Dropdown?
Split button?
Toggle?

Sidebars

- sidebar definitions are .ui-based
- live in *svx/uiconfig/ui/* (common to all apps) or in *$app/uiconfig/$app2/* (app-specific)
- example:

```
<child>
 <object class="sfxlo-SidebarToolBox" id="font">
   ...
 <child>
   <object class="GtkToolButton" id="fontname">
     <property name="visible">True</property>
     ...
     <property
name="action_name">.uno:CharFontName</property>
   </object>
```

Menus

- menu bar (XML) configuration in *$app/uiconfig/$app2/menubar*
- single config file for app
- example:

```
    <menu:menubar
xmlns:menu="http://openoffice.org/2001/menu"
menu:id="menubar">
    <menu:menu menu:id=".uno:PickList">
        <menu:menupopup>
            <menu:menuitem menu:id=".uno:AddDirect"/>
            <menu:menuitem menu:id=".uno:Open"/>
            <menu:menuitem menu:id=".uno:OpenRemote"/>
            <menu:menuitem menu:id=".uno:RecentFileList"/>
            <menu:menuitem menu:id=".uno:CloseDoc"/>
            <menu:menuseparator/>
            <menu:menu menu:id=".uno:TemplateMenu">
```

Icons

- add 2 icons in .png format to *icon-themes/galaxy/cmd*
  - (including in 'galaxy' theme is compulsory, other icon themes optional)
- UNO commands map to icon names
  - sc | *lc* + command name, lowecase, without .uno prefix + .png
  - **.uno:DoSomething => sc_dosomething.png**
- reuse existing icon
  - *icon-themes/*/links.txt*
  - add 1 entry/line of the form
    - *sc_newcommandicon.png sc_oldcommandicon.png*

Register UI element in application

- nothing to be done for sidebars
- toolbar buttons need to be registered
  - look for [sw|sc|sd|sm]dll.cxx, ::RegisterControllers() method
  - append the following line:

    ```
    MyToolBoxControl::RegisterControl(SID_DO
    _SOMETHING, pMod);
    ```
- dialogs need to be added to dialog factory
  - look for [sw|sc|sd|sm]dlgfact.cxx

Enter slots

- bind functionality to UI elements
- method slots e.g. open a dialog
- state slots (query a state e.g. text colour)
- live in *$app/sdi/* (application-specific) or in e.g in *svx/sdi* (global)

Define slot ID

- .hrc file, associate constant (usually of a form SID_DO_SOMETHING) with numerical ID
- app-specific in $app/inc or global e.g. *include/svx/*.hrc*, *include/editeng/*hrc*
- beware of duplicate IDs
- example:

```
#define SID_DO_SOMETHING 42
```

Add slot definition

- example method slot :

```
SfxVoidItem About SID_ABOUT
()
[
  /* flags: */
  AutoUpdate = FALSE,
  Cachable = Cachable,
  FastCall = FALSE,

  /* config */
  AccelConfig = TRUE,
  MenuConfig = TRUE,
  StatusBarConfig = FALSE,
  ToolBoxConfig = FALSE,
  GroupId = GID_APPLICATION;
]
```

It maps to SID_ABOUT slot

.uno:About command in disguise

This is a menu entry

Add slot definition II

- example property slot：

```
SvxColorItem Color SID_ATTR_CHAR_COLOR
[
  /* flags: */
  AutoUpdate = TRUE,
  Cachable = Cachable,
  FastCall = FALSE,

  /* config: */
  AccelConfig = FALSE,
  MenuConfig = FALSE,
  StatusBarConfig = FALSE,
  ToolBoxConfig = TRUE,
  GroupId = GID_FORMAT;
 ]
```

Return value

This is a toolbar button

Add slot interface

- associate slot IDs with actual function doing heavy-lifting
  - opens a dialog, changes property of an object
- tedious to write huge arrays of C++ function pointers => **svidl** compiler to the rescue
- example (*sd/sdi/_drvwsh.sdi*:

```
SID_ATTR_CHAR_COLOR
[
    ExecMethod = Execute;
    StateMethod = GetAttrState;
]
```

Add slot interface II

- in a subclass of SfxShell (DrawViewShell) the following methods exist:

```
void SomeViewShell::Execute( SfxRequest &rReq );
 void SomeViewShell::GetAttrState( SfxItemSet& rSet );
```

Further reading

- Old OOo wiki on slots and interfaces
- Old OOo wiki on UI XML config files
- Tutorial on creating simple dialog in Impress
- General .ui and widget layout documentation

THANK YOU!

CIB software GmbH
Elektrastraße 6a
81925 München
GERMANY

T +49(0)89 / 1 43 60 – 0
F +49(0)89 / 1 43 60 – 100
vertrieb@cib.de
www.cib.de