From extension to core

- Bjoern Michaelsen

- Member, Board of Directors, The Document Foundation
  bjoern.michaelsen@canonical.com

- Ubuntu LibreOffice package maintainer
  bjoern.michaelsen@canonical.com
I have no idea what I'm doing
Interfaces, Services and Implementations

A quick race through some UNO concepts ...
Interfaces, Services and Implementations

- **Interfaces** say what you can do with a UNO-thing
  - usually named com::sun::star...
  - and then a name prefixed with X

- **Services** promise to provide a set of Interfaces
  - usually named com::sun::star...
  - and then a name not prefixed with X

- **Implementations** can provide a service or a set of Interfaces
  - usually named … inconsistently
  - have some some real core code behind it
XCargoStorage

XTransport
Lorry service

- still abstract
- implements XTransport and XCargoStorage
Lorry implementation

- concrete
- implements a service
- has some real code behind it
UNO Runtime Reflection

How to find out what kind of UNO thing you are holding ...
XInterface

XInterface Interface Reference

base interface of all UNO interfaces More...

import "XInterface.idl";

Inheritance diagram for XInterface:

Public Member Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>More...</th>
</tr>
</thead>
<tbody>
<tr>
<td>any queryInterface ([in] type aType)</td>
<td>queries for a new interface to an existing UNO object.</td>
<td></td>
</tr>
<tr>
<td>void acquire ()</td>
<td>increases the reference counter by one.</td>
<td></td>
</tr>
<tr>
<td>void release ()</td>
<td>decreases the reference counter by one.</td>
<td></td>
</tr>
</tbody>
</table>
XTypeProvider

**XTypeProvider Interface Reference**

Interface to get information about the types (usually interface types) supported by an object. More...

```java
import "XTypeProvider.idl";
```

Inheritance diagram for XTypeProvider:

- `XInterface` → `XTypeProvider`

**Public Member Functions**

1. `sequence< type > getTypes ()`
   - Returns a sequence of all types (usually interface types) provided by the object. More...

2. `sequence< byte > getImplementationId ()`
   - Obsolete unique identifier. More...
XServiceInfo

XServiceInfo Interface Reference

Provides information regarding the implementation: which services are implemented and the name of the implementation. More...

import "XServiceInfo.idl";

Inheritance diagram for XServiceInfo:

```
  XInterface
    XServiceInfo
    XSelectionFunction
```

Public Member Functions

<table>
<thead>
<tr>
<th>Type</th>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>getImplementationName ()</td>
<td>Provides the implementation name of the service implementation. More...</td>
</tr>
<tr>
<td>boolean</td>
<td>supportsService ([in] string ServiceName)</td>
<td>Tests whether the specified service is supported, i.e. More...</td>
</tr>
<tr>
<td>sequence&lt;</td>
<td>getSupportedServiceNames ()</td>
<td>Provides the supported service names of the implementation, including also indirect service names. More...</td>
</tr>
</tbody>
</table>
Introducing the css.script.theServiceDocumenter singleton
Getting to the UNO documentation from your extension or script quickly ...
TheServiceDocumenter.showServiceDocs(…)

- [attribute] string ServiceBaseUrl;
- void showServiceDocs( [in] com::sun::star::lang::XServiceInfo xService );

- opens documentation for all services supported by this implementation in a browser
TheServiceDocumenter.showInterfaceDocs(...) 

- [attribute] string ServiceBaseUrl;
- void showServiceDocs( [in] com::sun::star::lang::XTypeProvider xTypeProvider );

- opens documentation for all interfaces (types) supported by this implementation in a browser
Example Ad-hoc StarBasic

```starbasic
TheServiceDocumenter = GetDefaultContext().getValueByName("/singletons/com.sun.star.util.theServiceDocumenter")

' old style reflection helpers
MsgBox(TheServiceDocumenter.Dbg_Methods)
MsgBox(TheServiceDocumenter.Dbg_Properties)
MsgBox(TheServiceDocumenter.Dbg_SupportedInterfaces)

' using the Service Documenter
TheServiceDocumenter.showServiceDocs(ThisComponent)
'TheServiceDocumenter.showInterfaceDocs(ThisComponent)
```
WIP: using the css.script.theServiceDocumenter singleton for finding core implementations

Getting to the C++ documentation of your implementation from your extension or script quickly ...
The old manual ways:

- ad-hoc: Just search for the implementation name on http://opengrok.libreoffice.org
- lots of clutter from testcode etc. though

- somewhat more reliable: finding implementation in .component files, then find the implementation in the cxx files of that library

- best solution: leak the information with a clang plugin triggering on getImplementationName(..)
Finding implementation via .component files

```
for f in `git ls-tree -r --name-only HEAD|grep '.component$'`
do
    grep 'implementation name' $f|nl -s "","$f"
done > componentgrep.csv
```

- In the second row, we then find the path of the component file
- In the third row, we then find the implementation name
- thus we can find the `{module}/Library_${libname}.mk` for each implementation and the `cxx` files where it could be implemented
clang, the Ultimate:

- best solution: leak the information with a clang plugin triggering on getImplementationName(..)

- There might be some cases where the implementation name is dynamically generated
- That likely a bad idea anyway, so fix that then
- (or worst case: fallback on the scripted solution via component files)
Example Ad-hoc StarBasic

```
TheServiceDocumenter = GetDefaultContext().getValueByName("/singletons/com.sun.star.util.theServiceDocumenter")

' old style reflection helpers
MsgBox(TheServiceDocumenter.Dbg_Methods)
MsgBox(TheServiceDocumenter.Dbg_PROPERTIES)
MsgBox(TheServiceDocumenter.Dbg_SupportedInterfaces)

' using the Service Documenter
TheServiceDocumenter.CoreBaseUrl = "http://people.canonical.com/~bjoern/implref/5.1/"
TheServiceDocumenter.showCoreDocs(ThisComponent)
```

- WIP: only works for SwXTextDocument for now
How should the Core Documentation Reference work in the end?

- It just opens
  "{$CoreBaseUrl}/{$ImplementationName}" in a browser

- and the URLs at {$CoreBaseUrl} are just statically generated redirects to http://docs.libreoffice.org

- (possibly we could also link to github or opengrok) later
Thank you!

- @Sweet5hark
- See you at the LibreOffice Aarhus Conference
- Find out more at http://conference.libreoffice.org

All text and image content in this document is licensed under the Creative Commons Attribution-Share Alike 3.0 License (unless otherwise specified). "LibreOffice" and "The Document Foundation" are registered trademarks. Their respective logos and icons are subject to international copyright laws. The use of these therefore is subject to the trademark policy.
Default Slide Example

- Here is space for your content …
- Some hints:
  - being the basis for the default slides
  - It would be useful to add your presentation title to the footer. Add it via

- Use the LibreOffice (libreoffice.soc) delivered with LibreOffice.
Pre-defined Shapes

Here are some pre-defined shapes for your convenience: copy the shapes, copy their formatting, or use the LibO styles.

Green Dark  ->  Green Bright
Blue Dark    ->  Blue Bright
Orange Dark  ->  Orange Bright
Purple Dark  ->  Purple Bright
Yellow Dark  ->  Yellow Bright

You may add your code examples, XML statements, or debug output here ;-)
Section Header Example

You may add additional text here ...