

# UNO API Info



## Creating Cross-Linked OOo-API-Documentation on the Fly

Rony G. Flatscher, WU Wien



# Overview



- Introduction
  - UNO IDL
- GUI Tool
  - "frontend\_UNO\_API\_info.rxo"
  - Examples
- Using the tool from programs
  - OOo Basic, Java, JavaScript, ooRexx, Python
- Roundup

# Introduction, 1



- UNO IDL
  - Allow you to define types
    - Constants, Enum(erations), Exceptions, Interfaces with Attributes and Methods, Services with Properties and Interfaces
  - Support for UNO IDL for programming languages
    - Allow you to interface with all UNO IDL types
    - Programming languages can be freely mixed



# Introduction, 2



- Huge number of types
- E.g. for Java (on OpenOffice.org 3.2.1)

jar Name	Total of Types	Interfaces	%
juh.jar	47	3	6,4%
ridl.jar	468	223	47,6%
jurt.jar	98	2	2,0%
unoil.jar	2.598	1.368	52,7%
<b>Totals</b>	<b>3.211</b>	<b>1.596</b>	<b>49,7%</b>

- Problems for Programmers
  - Impossible to know each type by heart !

# Introduction, 3



- Problems for Programmers (continued)
  - Which methods are available for Interfaces?
    - What are their signatures?
  - What attributes are available for Interfaces?
    - What are their types?
  - What is the structure of a Service?
    - What Property collection does it have, if any?
    - What Interfaces is it composed of?



# X-Ray



- OOo Basic's "X-Ray" by Bernard Marcellly
  - Allows you to inspect UNO objects at runtime
  - Very helpful for programmers
  - Unfortunately
    - No nicely formatted documents for studying off-line
    - No functionality that would give a structured overview
      - Difficult to gain an overview of the "parts" that constitute an area of programming
    - Not possible to x-ray an UNO IDL type by name only
- X-Ray-Projects for Python and others



# UNO API Info



- Tool got originally created by one of my students, *Nicole Scholz*, at "WU Wien"
- Purpose
  - Alleviate the documentation needs
  - Create on-the-fly structured documents containing links to OOO's documentation
    - Works on UNO objects (via introspection)
    - Works on any UNO IDL type name
  - Allow interactive use via a GUI
  - Allow programmatical use

# UNO API Info – The GUI



UNO API Viewer by Nicole Scholz (103.20100601, using UNO.CLS v135.20100724)

File About

Enter a fully qualified UNO IDL name:  
com.sun.star.embed.Storage Create UNO IDL Documentation

(UNO\_SERVICE) com.sun.star.embed.Storage (mandatory service, mand.+opt. interfaces, properties)

Choose # of layers/levels to analyze: 1 Analyze given UNO IDL only (first layer/level)

Choose the font to use for rendering: Verdana

Choose numbering type for first chapter level: ROMAN\_UPPER 2

Choose numbering type for second chapter level: ARABIC 4 2 Numbering is in Roman numbers with upper case letters as I, II, III, IV

Choose numbering type for third chapter level: ARABIC 4

Determine what to do with the generated documentation: 1 View in writer

Directory/folder to store generated files to: E:\rony\dev\bsf\src\BSF4ooRexx.dev\utilities\OOo\UNO\_API\_info

Path to API documentation directory to hyperlink to: 1 Use the OOo Internet API documentation site

Directory/folder of the local SDK installation: OpenOffice.org\_x.y\_SDK\





# UNO API Info – The GUI

A screenshot of the OpenOffice Writer application window. The title bar reads "UNO\_API\_INFO-com.sun.star.embed.Storage-2010-08-26T20\_11\_03.875000 - OpenOffice.org Writer". The menu bar includes "File", "Edit", "View", "Insert", "Format", "Table", "Tools", "Window", and "Help". The toolbar contains various icons for file operations, editing, and formatting. The status bar at the bottom shows "Page 1 / 3", "Default", "INSRT", "STD", "read-only : Table of Contents1", and "100%". The main document area displays a "Table Of Content" with the following entries:

Table Of Content	
[Storage (com.sun.star.embed) [xref]] .....	2
I. [Storage [xref]] (UNO_SERVICE) by Name.....	2
II. [Storage [xref]] (UNO_SERVICE) by UNO Types.....	2
III. [Storage] (UNO_SERVICE) Graphical.....	3



# UNO API Info – The GUI



The screenshot shows the OpenOffice Writer interface with a document containing UNO API documentation. The document title is "[Storage (com.sun.star.embed) [xref]]". The main heading is "I. [Storage [xref]] (UNO\_SERVICE) by Name". Below this is a table listing members of the storage service.

Strg-click to open hyperlink: <http://api.openoffice.org/docs/common/ref/com/sun/star/embed/Storage>

	Member Name	Type
1	<a href="#">BaseStorage (com.sun.star.embed)</a>	UNO_SERVICE
2	<a href="#">HasEncryptedEntries</a> <java.lang.Boolean>	UNO_PROPERTY
3	<a href="#">HasNonEncryptedEntries</a> <java.lang.Boolean>	UNO_PROPERTY
4	<a href="#">IsRoot</a> <java.lang.Boolean>	UNO_PROPERTY
5	<a href="#">MediaType</a> <java.lang.String>	UNO_PROPERTY
6	<a href="#">MediaTypeFallbackIsUsed</a> <java.lang.Boolean>	UNO_PROPERTY
7	<a href="#">RepairPackage</a> <java.lang.Boolean>	UNO_PROPERTY
8	<a href="#">Version</a> <java.lang.String>	UNO_PROPERTY
9	<a href="#">XEncryptionProtectedSource (com.sun.star.embed)</a>	UNO_INTERFACE
10	<a href="#">XTransactedObject (com.sun.star.embed)</a>	UNO_INTERFACE



# UNO API Info – The GUI



File Edit View Insert Format Table Tools Window Help

Heading 1 Verdana 14 B I U

2 1 1 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

## II. [Storage *xref*] (UNO\_SERVICE) by UNO Types

Strg-click to open hyperlink: <http://api.openoffice.org/docs/common/ref/com/sun/star/embed/Storage-xref.html>

Name	
<b>UNO_INTERFACE:</b>	
1	XEncryptionProtectedSource (com.sun.star.embed) (OPTIONAL)
2	XTransactedObject (com.sun.star.embed) (OPTIONAL)
3	XTransactionBroadcaster (com.sun.star.embed) (OPTIONAL)
<b>UNO_PROPERTY:</b>	
1	HasEncryptedEntries <java.lang.Boolean> (OPTIONAL+READONLY)
2	HasNonEncryptedEntries <java.lang.Boolean> (OPTIONAL+READONLY)
3	IsRoot <java.lang.Boolean> (READONLY)
4	MediaType <java.lang.String>
5	MediaTypeFallbackIsUsed <java.lang.Boolean> (READONLY)
6	RepairPackage <java.lang.Boolean> (OPTIONAL+READONLY)
7	Version <java.lang.String> (OPTIONAL)
<b>UNO_SERVICE:</b>	
1	BaseStorage (com.sun.star.embed) (OPTIONAL)

Page 2 / 3 Default German (Austria) INSRT STD Outline Numbering : Level 1 100%

# UNO API Info – The GUI



The screenshot shows the OpenOffice Writer interface with a document titled "III. [Storage] (UNO\_SERVICE) Graphical". The document content is as follows:

```
graph TD
    A["UNO_SERVICE  
com.sun.star.embed.Storage"]
    B["UNO_INTERFACE  
XEncryptionProtectedSource (com.sun.star.embed)  
XTransactedObject (com.sun.star.embed)  
XTransactionBroadcaster (com.sun.star.embed)"]
    C["UNO_PROPERTY  
HasEncryptedEntries  
HasNonEncryptedEntries  
IsRoot  
MediaType  
MediaTypeFallbackIsUsed  
RepairPackage  
Version"]
    D["UNO_SERVICE  
BaseStorage (com.sun.star.embed)"]
```

The interface includes a menu bar (File, Edit, View, Insert, Format, Table, Tools, Window, Help), a toolbar with various icons, a text area with "Heading 1" and "Verdana" font, and a status bar at the bottom showing "Page 3 / 3", "Default", "German (Austria)", "INSRT", "STD", "Outline Numbering : Level 1", and "100%" zoom.



Overview Module Use Devguide Index

SERVICES' SUMMARY INTERFACES' SUMMARY PROPERTIES' SUMMARY SERVICES' DETAILS INTERFACES' DETAILS PROPERTIES' DETAILS

:: com :: sun :: star :: embed ::

*unpublished* service Storage

Usage Restrictions  
*not published*

Description

This is a service that allows to get access to a package using storage hierarchy.  
A root storage should be retrieved by using **StorageFactory** service. Substorages are created through **XStorage** interface of a parent storage.

Included Services - Summary

<b>BaseStorage</b>	This service describes the base functionality of storages. ( <a href="#">details</a> )
--------------------	--

Exported Interfaces - Summary

<b>XTransactedObject</b>	allows to commit or revert changes that were done for the storage. ( <a href="#">details</a> )
<b>XTransactionBroadcaster</b>	allows to track storage's transaction state.
<b>XEncryptionProtectedSource</b>	allows to set password to a root storage. ( <a href="#">details</a> )



Projects > api

Project tools

- Project home
- Membership
- Announcements
- Mailing lists
- Documents & files
- Version control
- Module structure
- Issue tracker

Developer's Guide

- Content Table
- IDL reference
- IDL Design Guide
- IDL Docu Guide

SDK

- Examples
- Java UNO Reference
- C++ UNO Reference
- Download

Tips 'n' tricks

- FAQ
- Internal OO Spots
- External Resources

Overview Module Use Devguide Index

# uses of service Storage

[back to service Storage](#)

- Services which Include this Service
- Singletons which Instantiate this Service
- References in Developers Guide

[Top of Page](#)

Copyright © 2009 Sun Microsystems, Inc.



- Project home
- Membership
- Announcements
- Mailing lists
- Documents & files
- Version control
- Module structure
- Issue tracker
- Developer's Guide**
- Content Table
- IDL reference
- IDL Design Guide
- IDL Docu Guide
- SDK**
- Examples
- Java UNO Reference
- C++ UNO Reference
- Download
- Tips 'n' tricks**
- FAQ
- Internal OO Spots
- External Resources
- Miscellaneous**
- Developer Projects
- Mailing List Rules
- News Letter Archive
- Search**
- This project
- Go
- Advanced search
- Google search

## uses of interface XResourceId

[back to interface XResourceId](#)

### Derived Interfaces

Synonym Typedefs

[Services which Support this Interface](#)

[Singletons which Support this Interface](#)

[Uses as Return Type](#)

[Uses as Parameter](#)

[Uses as Data Type](#)

[References in Developers Guide](#)

### Services which Support this Interface

[ResourceId](#)

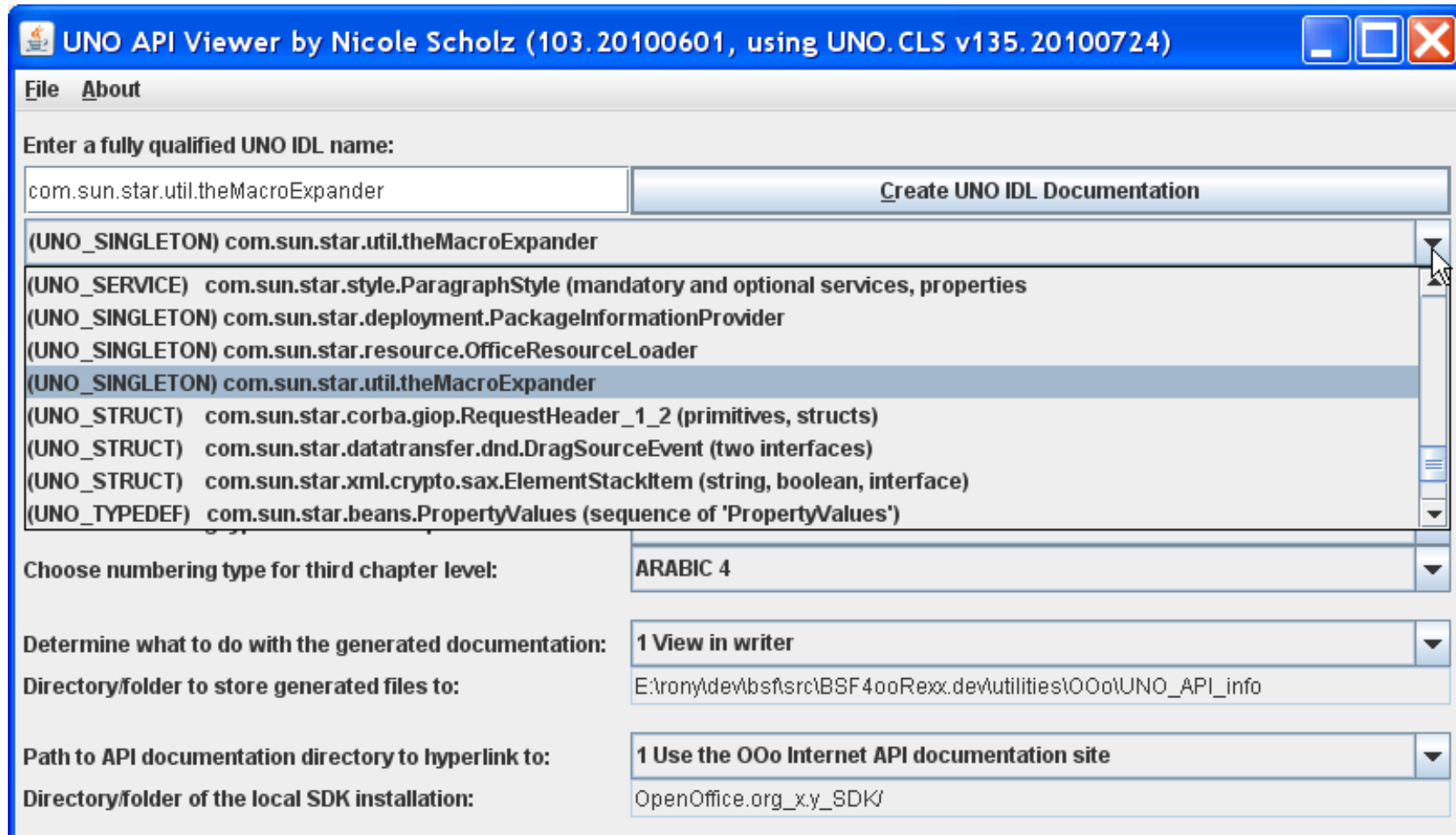
### Uses as Return Type

[XResourceId::clone\(\)](#)  
[XResourceId::getAnchor\(\)](#)  
[XResource::getResourceId\(\)](#)

### Uses as Parameter

[XConfiguration::addResource\(\)](#)  
[XResourceId::compareTo\(\)](#)  
[com.sun.star.drawing.SlideSorter::create\(\)](#)

# UNO API Info – The GUI





# Using "UNO API Info" From Programs



- Tool is an ooRexx application/macro
  - Needs the opensource *ooRexx* from
    - <http://www.oorexx.org/download.html>
  - Needs *BSF4ooRexx* from
    - <http://wi.wu-wien.ac.at/rgf/rexx/bsf4oorexx/current/>
  - Installation all in all: 20 seconds
- Any OOo programming language can use it via the dispatch interface
  - "com.sun.star.frame.XDispatchProvider"



# Using "UNO API Info" From OOO Basic, 1



' demonstrates how to use "UNO\_API\_info.rxo" from OOO Basic

```
Sub testCreateApiInfo
DIM sDispatchHelper AS object, xDispatchProvider AS object ' objects
DIM macroUrl, library, scriptName, langName, location ' variants
DIM args1(0) AS NEW com.sun.star.beans.PropertyValue ' array of type PropertyValue
DIM args2(1), options(6) ' arrays of variants
sDispatchHelper =createUnoService("com.sun.star.frame.DispatchHelper") ' create DispatchHelper service
xDispatchProvider=ThisComponent.CurrentController.Frame ' get dispatch provider interface of current Desktop
' define Rexx dispatch target, library "wu_tools", script name "create_UNO_API_info.rex", location "share"
location = "share" ' case sensitive, other possible values: "user" (current user), "application"
libraryName = "wu_tools" ' case sensitive, name of the Rexx macro library
scriptName = "UNO_API_info.rxo" ' case sensitive, name of the Rexx script
langName = "ooRexx" ' case sensitive, OOO name of the scripting language
' build 'macroUrl' string for the dispatcher
macroUrl = "vnd.sun.star.script:" & libraryName & "." & scriptName & "?language=" & langName & "&location=" _
& location

' ----- use one argument denoting an UNO object from the running program
' define one argument (an UNO object from the running program)
' remark: the array 'args1' is explicitly defined to be of type com.sun.star.beans.PropertyValue,
' hence its element is a PropertyValue object already
args1(0).name="arg1" ' name of the PropertyValue
args1(0).value=sDispatchHelper ' value: UNO object to analyze and document
' dispatching to 'UNO_API_info.rxo' using an UNO object from the running program
sDispatchHelper.executeDispatch(xDispatchProvider, macroUrl, "", 0, args1())
```

# Using "UNO API Info" From OOo Basic, 2



```
' define options; create PropertyValue objects and assign them to the 'options' variant array
options(0)=createProperty("NrOfLayers", 2) ' 2="show two levels deep"
options(1)=createProperty("View", 1) ' 1="view in writer"
options(2)=createProperty("DocumentationSource", 1) ' 1="use Internet" (base url)
options(3)=createProperty("NumberingTypeLevel_1", 0) ' 0="Alpha Uppercase"
options(4)=createProperty("NumberingTypeLevel_2", 4) ' 4="arabic"
options(5)=createProperty("NumberingTypeLevel_3", 3) ' 3="roman lower"
options(6)=createProperty("FontName", "DejaVu Sans Condensed")

' define two arguments (an UNO IDL string and formatting options
' create PropertyValue objects and assign them to the 'args2' variant array
args2(0)=createProperty("arg1", "com.sun.star.frame.Desktop") ' an UNO IDL string
args2(1)=createProperty("arg2", options) ' rendering options
' dispatching to 'UNO_API_info.rxo' using an UNO IDL string and rendering options
sDispatchHelper.executeDispatch(xDispatchProvider, macroUrl, "", 0, args2())
```

# Using "UNO API Info" From Java, 1



```
import com.sun.star.beans.PropertyValue;
import com.sun.star.frame.XDesktop;
import com.sun.star.frame.XDispatchHelper;
import com.sun.star.frame.XDispatchProvider;
import com.sun.star.lang.XMultiComponentFactory;
import com.sun.star.lang.XMultiServiceFactory;
import com.sun.star.uno.UnoRuntime;
import com.sun.star.uno.XComponentContext;
class HowtoCreateApiInfo {
    public static void main (String args[]) {
        // excerpted from "HardFormatting.java" from the OOo development package
        XDesktop xDesktop = null;
        XMultiComponentFactory xMCF = null;
        XMultiServiceFactory xMSF = null;
        try {
            XComponentContext xContext = null;
            xContext = com.sun.star.comp.helper.Bootstrap.bootstrap();// bootstrap the UNO runtime environment
            xMCF = xContext.getServiceManager(); // get the service manager
            xMSF = (XMultiServiceFactory) UnoRuntime.queryInterface(XMultiServiceFactory.class, xMCF);
            if (xMSF!=null)
            {
                System.out.println("Java: connected to the bootstrapped office ...\n---");
                // get XDispatchProvider from XDesktop
                Object oDesktop = xMSF.createInstance("com.sun.star.frame.Desktop");
                xDesktop = (XDesktop) UnoRuntime.queryInterface(XDesktop.class, oDesktop);
                XDispatchProvider xDispatchProvider=(XDispatchProvider)
                    UnoRuntime.queryInterface(XDispatchProvider.class, xDesktop);
                Object sDispatchHelper= xMSF.createInstance("com.sun.star.frame.DispatchHelper");
                XDispatchHelper xDispatchHelper=(XDispatchHelper)
                    UnoRuntime.queryInterface(XDispatchHelper.class, sDispatchHelper);
            }
        }
    }
}
```

# Using "UNO API Info" From Java, 2



```
// invoke the ooRexx script to document the UNO object/IDL
// define Rexx dispatch target, library "wu_tools", script name "create_UNO_API_info.rxo", location "share"
String location    ="share",    // case sensitive, other possible values: "user", "application"
    libraryName="wu_tools",      // case sensitive, name of the Rexx macro library
    scriptName ="UNO_API_info.rxo", // case sensitive, name of the Rexx script
    langName   ="ooRexx";       // case sensitive, OOo name of the scripting language
    // build 'macroUrl' string for the dispatcher
String macroUrl="vnd.sun.star.script:"+libraryName+"."+scriptName+
    "?language="+langName+
    "&location="+location;

// define one argument (an UNO object from the running program)
PropertyValue args1[]={createProperty("arg1", sDispatchHelper) };
System.out.println("Java: dispatching to 'create_UNO_API_info.rxo' using an UNO object from"+
    " the running program...");

// dispatch, supplying arguments
xDispatchHelper.executeDispatch(
    xDispatchProvider, // XdispatchProvider
    macroUrl,          // URL
    "",                // TargetFrameName
    0,                 // SearchFlags
    args1);           // Arguments
```

# Using "UNO API Info" From Java, 3



```
// ===== next dispatch uses an UNO IDL string and options
System.out.println("---");
// define options
PropertyValue options[]= {
    createProperty("NrOfLayers",          new Integer(2)), // 2="show two levels deep"
    createProperty("View",                new Integer(1)), // 1="view in writer"
    createProperty("DocumentationSource",  new Integer(1)), // 1="use Internet" (base url)
    createProperty("NumberingTypeLevel_1", new Integer(0)), // 0="Alpha Uppercase"
    createProperty("NumberingTypeLevel_2", new Integer(4)), // 4="arabic"
    createProperty("NumberingTypeLevel_3", new Integer(3)), // 3="roman lower"
    createProperty("FontName",            "DejaVu Sans Condensed")
};

// define two arguments
PropertyValue args2[]=
{
    createProperty("arg1", "com.sun.star.frame.Desktop"), // analyze UNO IDL name
    createProperty("arg2", options)                       // rendering options
};
System.out.println("Java: dispatching to 'create_UNO_API_info.rxo' using an UNO IDL string +
    " and rendering options...");

// dispatch, supplying arguments
xDispatchHelper.executeDispatch(
    xDispatchProvider, // XdispatchProvider
    macroUrl,          // URL
    "",                 // TargetFrameName
    0,                  // SearchFlags
    args2);             // Arguments
```

# Using "UNO API Info" From Java, 4



... cut ...

```
// utility method to ease creation of PropertyValue objects
static PropertyValue createProperty(String n, Object v)
{
    PropertyValue prop=new PropertyValue();
    prop.Name =n;
    prop.Value=v;
    return prop;
}
```

... cut ...

# Using "UNO API Info" From JavaScript, 1



```
importClass(Packages.com.sun.star.beans.PropertyValue);
importClass(Packages.com.sun.star.frame.XDispatchHelper);
importClass(Packages.com.sun.star.frame.XDispatchProvider);
importClass(Packages.com.sun.star.uno.UnoRuntime);

// utility method to ease creation of PropertyValue objects
function createProperty(n, v)
{
    prop=new PropertyValue();
    prop.Name =n;
    prop.Value=v;
    return prop;
}
```



# Using "UNO API Info" From JavaScript, 2



```
// get important objects from the XSCRIPTCONTEXT
xDispatchProvider = UnoRuntime.queryInterface(XDispatchProvider, XSCRIPTCONTEXT.getDesktop());
ctx = XSCRIPTCONTEXT.getComponentContext();
smgr = ctx.getServiceManager();

// define Rexx dispatch target, library "wu_tools", script name "create_UNO_API_info.rxo", location "share"
location      = "share";
libraryName   = "wu_tools";
scriptName    = "UNO_API_info.rxo";
langName      = "ooRexx";
// build -- macroUrl-- string for the dispatcher
macroUrl = "vnd.sun.star.script:"+libraryName+"."+scriptName+"?language="+langName+"&location="+location;

// create an UNO service object
sdh = smgr.createInstanceWithContext("com.sun.star.frame.DispatchHelper", ctx);
xDispatchHelper = UnoRuntime.queryInterface(XDispatchHelper, sdh);
args = new Array;
args[0]= createProperty("arg1", sdh);
// dispatch the ooRexx macro to document the 'sdh' service object using the default settings
xDispatchHelper.executeDispatch(xDispatchProvider, macroUrl, "", 0, args);
// -----
```

# Using "UNO API Info" From JavaScript, 3



```
// must define this PropertyValue array as a Java array as otherwise the bridge will not be
// able to correctly convert the JavaScript dynamic array in the executeDispatch invocation
options = java.lang.reflect.Array.newInstance(PropertyValue, 7);
options[0]= createProperty("NrOfLayers", "2" ); // 2="show two levels deep"
options[1]= createProperty("View", "1" ); // 1="view in writer"
options[2]= createProperty("DocumentationSource", "1" ); // 1="use Internet" (base url)
options[3]= createProperty("NumberingTypeLevel_1", "0" ); // 0="Alpha Uppercase"
options[4]= createProperty("NumberingTypeLevel_2", "4" ); // 4="arabic"
options[5]= createProperty("NumberingTypeLevel_3", "3" ); // 3="roman lower"
options[6]= createProperty("FontName", "DejaVu Sans Condensed");

// define two arguments
args[0] = createProperty("arg1", "com.sun.star.frame.Desktop"); // analyze UNO IDL name
args[1] = createProperty("arg2", options); // rendering options

// dispatch the ooRexx macro to document the UNO IDL 'com.sun.star.frame.Desktop' using specific settings
xDispatchHelper.executeDispatch(xDispatchProvider, macroUrl, "", 0, args);
```

# Using "UNO API Info" From ooRexx, 1



```
xScriptContext = uno.getScriptContext()      -- get Script context
xComponentContext = xScriptContext~getComponentContext
xDesktop = xScriptContext~getDesktop

-- this macro just works externally, called by rexxj or rexx
-- create DispatchHelper service and query its interface
xMultiServiceFactory = xComponentContext~getServiceManager~XMultiServiceFactory
sDispatchHelper = xMultiServiceFactory~createInstance("com.sun.star.frame.DispatchHelper")
xDispatchHelper = sDispatchHelper~XDispatchHelper
xDispatchProvider = xDesktop~XDispatchProvider -- get dispatch provider interface of current Desktop

-- define Rexx dispatch target, library "wu_tools", script name "create_UNO_API_info.rxo", location "share"
location = "share" -- case sensitive, other possible values: "user" (current user),
"application"
libraryName = "wu_tools" -- case sensitive, name of the Rexx macro library
scriptName = "UNO_API_info.rxo" -- case sensitive, name of the Rexx script
langName = "ooRexx" -- case sensitive, OOo name of the scripting language
-- build -- macroUrl-- string for the dispatcher
macroUrl="vnd.sun.star.script:"libraryName"."scriptName"?language="langName"&location="location

-- define one argument (an UNO object from the running program)
args = uno.CreateArray(.UNO~PROPERTYVALUE, 1) -- array for argument
args[1] = uno.createProperty("arg1", sDispatchHelper)
-- dispatch
res = xDispatchHelper~executeDispatch(xDispatchProvider, macroURL, "", 0, args)
```

# Using "UNO API Info" From ooRexx, 2



```
-- define options
options = uno.CreateArray(.UNO~PROPERTYVALUE, 7) /* define array for options */
options[1] = uno.createProperty("NrOfLayers", 2) -- 2="show two levels deep"
options[2] = uno.createProperty("View", 1) -- 1="view in writer"
options[3] = uno.createProperty("DocumentationSource", 1) -- 1="use Internet" (base url)
options[4] = uno.createProperty("NumberingTypeLevel_1", 0) -- 0="Alpha Uppercase"
options[5] = uno.createProperty("NumberingTypeLevel_2", 4) -- 4="arabic"
options[6] = uno.createProperty("NumberingTypeLevel_3", 3) -- 3="roman lower"
options[7] = uno.createProperty("FontName", "DejaVu Sans Condensed")
-- define two arguments
args=uno.createArray(.uno~propertyValue, 2) -- we have two arguments
args[1]=uno.createProperty("arg1", "com.sun.star.frame.Desktop") -- an UNO IDL string
args[2]=uno.createProperty("arg2", options) -- the options for rendering
-- dispatch
res = xDispatchHelper~executeDispatch(xDispatchProvider, macroURL, "", 0, args)

::requires UNO.CLS -- get the UNO-support (includes BSF.CLS, i.e. Java-support)
```

# Using "UNO API Info" From Python, 1



```
from com.sun.star.beans import PropertyValue
def createApiInfo( ):
    """Uses the ooRexx macro UNO_API_info.rxo to document an OOo service object and a UNO IDL type string"""
# get the OOo Desktop and the ServiceManager
    desktop = XSCRIPTCONTEXT.getDesktop()
    ctx = XSCRIPTCONTEXT.getComponentContext()
    smgr = ctx.ServiceManager
# define Rexx dispatch target, library "wu_tools", script name "create_UNO_API_info.rxo", location "share"
    location = "share"
    libraryName = "wu_tools"
    scriptName = "UNO_API_info.rxo"
    langName = "ooRexx"
# build -- macroUrl-- string for the dispatcher
    macroUrl = "vnd.sun.star.script:"+libraryName+"."+scriptName+"?language="+langName+"&location="+location
# create an UNO service object
    sdh = smgr.createInstance("com.sun.star.frame.DispatchHelper")
    a1 = createPropertyValue("arg1", sdh)
# dispatch the ooRexx macro to document the 'sdh' service object using the default settings
    sdh.executeDispatch(desktop, macroUrl, "", 0, (a1,))
```

# Using "UNO API Info" From Python, 2



```
# --- document an UNO IDL string, change the formatting default values
a1.Value = "com.sun.star.frame.Desktop" # an UNO IDL string
p1=createPropertyValue("NrOfLayers", 2) # 2="show two levels deep"
p2=createPropertyValue("View", 1) # 1="view in writer"
p3=createPropertyValue("DocumentationSource", 1) # 1="use Internet" (base url)
p4=createPropertyValue("NumberingTypeLevel_1", 0) # 0="Alpha Uppercase"
p5=createPropertyValue("NumberingTypeLevel_2", 4) # 4="arabic"
p6=createPropertyValue("NumberingTypeLevel_3", 3) # 3="roman lower"
p7=createPropertyValue("FontName", "DejaVu Sans Condensed")
a2=createPropertyValue("arg2", (p1,p2,p3,p4,p5,p6,p7))
# dispatch the ooRexx macro to document the UNO IDL definitions using the supplied settings
sdh.executeDispatch(desktop, macroUrl, "", 0, (a1,a2))

def createPropertyValue (name, value):
    """Utility function to ease creation of PropertyValues"""
    pv = PropertyValue()
    pv.Name = name      # assign name
    pv.Value = value   # assign value
    return pv          # return PropertyValue object

# list those functions that should be shown in the OOo-UI
g_exportedScripts = createApiInfo,
```

# Roundup



- "UNO\_API\_Info.rxo"
  - A powerful documentation tool
    - Gets installed with "BSF4ooRexx"
    - Can be used interactively using a GUI
    - Can be used programmatically
      - Functionality is available via "com.sun.star.frame.XDispatchProvider"
      - Therefore *all* OOo programming languages can take advantage of it!
  - Allows you to create Writer and PDF documents that are fully linked to the official OOo documentation on the Internet



# Credits



- "UNO\_API\_info.orx" was originally created by Nicole Scholz at WU Wien
  - [http://wi.wu-wien.ac.at/rgf/diplomarbeiten/index.htm#dipl\\_200903](http://wi.wu-wien.ac.at/rgf/diplomarbeiten/index.htm#dipl_200903)
- All helpful OOo developers from the opensource community and Sun/Oracle
- The ooRexx development team
- The Apache Software Foundation's (ASF) Jakarta project "Bean Scripting Framework (BSF)"

