

OpenOffice.org



Ad-hoc-Generierung der API- Dokumentation

Prof. Dr. Rony G. Flatscher, WU Wien
(37. Tagung, GSE WG Rexx plus ISPF, Frankfurt, 2010-09-21)



Überblick



- Einleitung
 - UNO IDL
- GUI Tool
 - "frontend_UNO_API_info.rxo"
 - Beispiele
- Nutzung des Rexx-Tools von anderen Programmiersprachen heraus
 - OOo Basic, Java, JavaScript, ooRexx, Python
- Zusammenfassung

Einleitung, 1



- UNO IDL
 - Erlaubt die Definition von Datentypen
 - Konstante ("Constant"), Aufzählungen ("Enum"), Ausnahmen ("Exception"), "Interface" mit "Attribute(n)" und "Method(en)", "Service" mit Eigenschaften ("Property") und "Interfaces"
 - Programmiersprachen, die UNO IDL unterstützen
 - Können alle UNO IDL Datentypen nutzen
 - Programmiersprachen können frei kombiniert werden



Einleitung, 2



- Riesige Anzahl an Datentypen
- Z.B. für Java (auf OpenOffice.org 3.2.1)

jar-Name	Summe Datentypen	Davon "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%
Summen	3.211	1.596	49,7%

- Problem für Programmierer
 - Unmöglich, jeden Datentyp auswändig zu kennen!

Einleitung, 3



- Problem für Programmierer
 - Welche Methoden sind in welchen Interfaces verfügbar?
 - Welche Signaturen haben diese?
 - Welche Attribute sind für Interfaces verfügbar?
 - Welchen Datentyp tragen sie?
 - Wie ist die Struktur eines Service?
 - Über welche "Property"-Sammlung verfügt es, wenn es denn überhaupt eine gibt?
 - Aus welchen Interfaces ist es zusammengesetzt?



X-Ray



- OOo Basic's "X-Ray" von Bernard Marcellly
 - Erlaubt den Einblick in UNO-Objekte zur Laufzeit
 - Sehr hilfreich für Programmierer
 - Allerdings
 - Erzeugt keine schön formatierten Dokumente zum Off-line-Studium
 - Kann keine Überblicksdokumentation erstellen
 - Daher schwierig, einen Überblick über die Zusammenhänge zu gewinnen
 - Keine namentliche Analyse von UNO-IDL-Datentypen
- X-Ray-Projekte für Python et.al.



UNO API Info



- Tool, das ursprünglich von einer meiner Studentinnen, *Nicole Scholz* (WU Wien), entwickelt wurde
 - Zweck
 - Bedarf an Dokumentation stillen helfen
 - Erstellung von ad-hoc-Dokumentationen, die mit den offiziellen Ooo-HTML-Seiten verlinkt sind
 - Dokumentiert UNO-Objekte zur Laufzeit
 - Dokumentiert namentlich UNO-IDL-Datentypen
 - Bereitstellung über interaktives GUI
 - Erlaube Benutzung von beliebigen Ooo-Programmen aus



UNO API Info – Das GUI



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

File About

Enter a fully qualified UNO IDL name:

(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 – Das 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 application window. The main content area displays the following text:

[Storage (com.sun.star.embed) [xref]]

I. [Storage [xref]] (UNO_SERVICE) by Name

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

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

The application window 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 2 / 3", "Default", "German (Austria)", "INSRT", "STD", "Outline Numbering : Level 1", and "100%" zoom.

UNO API Info – Das GUI



File Edit View Insert Format Table Tools Window Help

Heading 1 Verdana 14 B I U

2 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	
UNO_INTERFACE:	
1	XEncryptionProtectedSource (com.sun.star.embed) (OPTIONAL)
2	XTransactedObject (com.sun.star.embed) (OPTIONAL)
3	XTransactionBroadcaster (com.sun.star.embed) (OPTIONAL)
UNO_PROPERTY:	
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)
UNO_SERVICE:	
1	BaseStorage (com.sun.star.embed) (OPTIONAL)

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

UNO API Info – Das GUI



File Edit View Insert Format Table Tools Window Help

Heading 1 Verdana 14 B I U

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

III. [Storage] (UNO_SERVICE) Graphical

UNO_SERVICE
com.sun.star.embed.Storage

UNO_INTERFACE
XEncryptionProtectedSource (com.sun.star.embed)
XTransactedObject (com.sun.star.embed)
XTransactionBroadcaster (com.sun.star.embed)

UNO_PROPERTY
HasEncryptedEntries
HasNonEncryptedEntries
IsRoot
MediaType
MediaTypeFallbackIsUsed
RepairPackage
Version

UNO_SERVICE
BaseStorage (com.sun.star.embed)

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



Service Storage - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://api.openoffice.org/docs/common/ref/com/sun/star/embed/Storage

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

BaseStorage	This service describes the base functionality of storages. (details)
--------------------	--

Exported Interfaces - Summary

XTransactedObject	allows to commit or revert changes that were done for the storage. (details)
XTransactionBroadcaster	allows to track storage's transaction state.
XEncryptionProtectedSource	allows to set password to a root storage. (details)



the free and open productivity suite

Search

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.



Service Storage
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 [v]
[] [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](#)

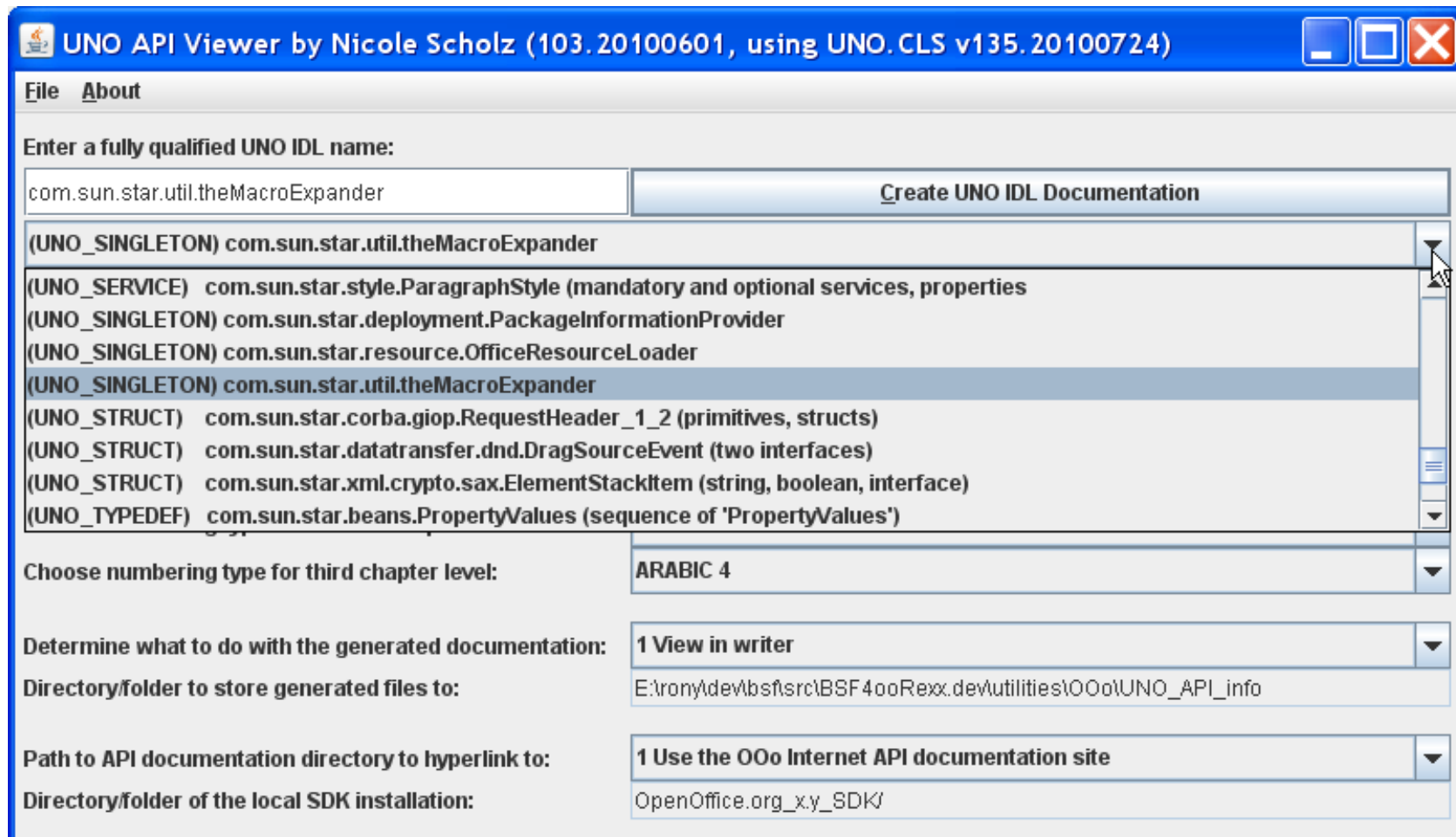
Uses as Return Type

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

Uses as Parameter

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

UNO API Info – The GUI



"UNO API Info" von anderen Programmen aus



- Tool ist in ooRexx implementiert
 - Benötigt daher *ooRexx*
 - <http://www.oorexx.org/download.html>
 - *Benötigt BSF4ooRexx*
 - <http://wi.wu-wien.ac.at/rgf/rexx/bsf4oorexx/current/>
 - Installationszeit, etwa 20 Sekunden
- Jede OOo-Programmiersprache kann das Tool über das OOo-Dispatch-Interface benutzen
 - "com.sun.star.frame.XDispatchProvider"



"UNO API Info"

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())
```

"UNO API Info"

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())
```

"UNO API Info"

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);
            }
        }
    }
}
```

"UNO API Info"

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
```

"UNO API Info"

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
```

"UNO API Info"

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 ...

"UNO API Info" 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;
}
```


"UNO API Info" 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);
// -----
```

"UNO API Info" 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);
```

"UNO API Info"

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)
```

"UNO API Info"

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)
```

"UNO API Info" 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,))
```

"UNO API Info"

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,
```

Zusammenfassung



- "UNO_API_Info.rxo"
 - Erzeugt Writer- bzw. PDF-Dokumente, die mit der OOo-Internet-Doku verlinkt sind!
 - Ein mächtiges Dokumentations-Tool
 - Wird mit "BSF4ooRexx" installiert
 - Kann interaktiv mit einem GUI benutzt werden
 - Menü "*BSF4ooRexx* → *Utilities*",
"*OOo/UNO_API_info/frontend_UNO_API_info.rxo*"
 - Kann von Programmen aus benutzt werden
 - Aufruf mit Hilfe des OOo-Interfaces
"*com.sun.star.frame.XDispatchProvider*"
 - Daher können *alle* OOo-Programmiersprachen dieses Tool für die Programmierung benutzen!



Credits



- "UNO_API_info.rxo" wurde ursprünglich von Nicole Scholz, WU Wien, erstellt
 - http://wi.wu-wien.ac.at/rgf/diplomarbeiten/index.htm#dipl_200903
- Alle hilfreichen OOo-Entwickler aus der Opensource- und Sun/Oracle-Welt
- Das ooRexx-Entwicklerteam
- Das Apache Software Foundation (ASF) "Jakarta"-Projekt namens "Bean Scripting Framework (BSF)"

