Programming Apache OpenOffice

The Universal Network Object (UNO) Framework

Rony G. Flatscher

FOSDEM'13

Overview

- Introduction
  - UNO IDL
  - Bird eye's view
- GUI based on-the-fly documentation tool
  - "frontend_UNO_API_info.rxo"
  - Examples
- Using the tool from programs
  - AOO Basic, Java, JavaScript, ooRexx, Python
- Roundup

Introduction

- Cf. presentation “OOo Budapest 2010”
- Works for LibreOffice (LO) as well
- UNO IDL
  - Allows 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

Bird Eye's View, 1

- Set of *services* that may contain *interfaces* with *attributes*, other *services*, *structs* and *properties*
- All common functionality of all types of documents is extracted and organized as a set of *interfaces* that define *methods* and possibly *attributes*
  - E.g. loading, saving, printing documents, …
- *Services* are created and get managed by *service managers*

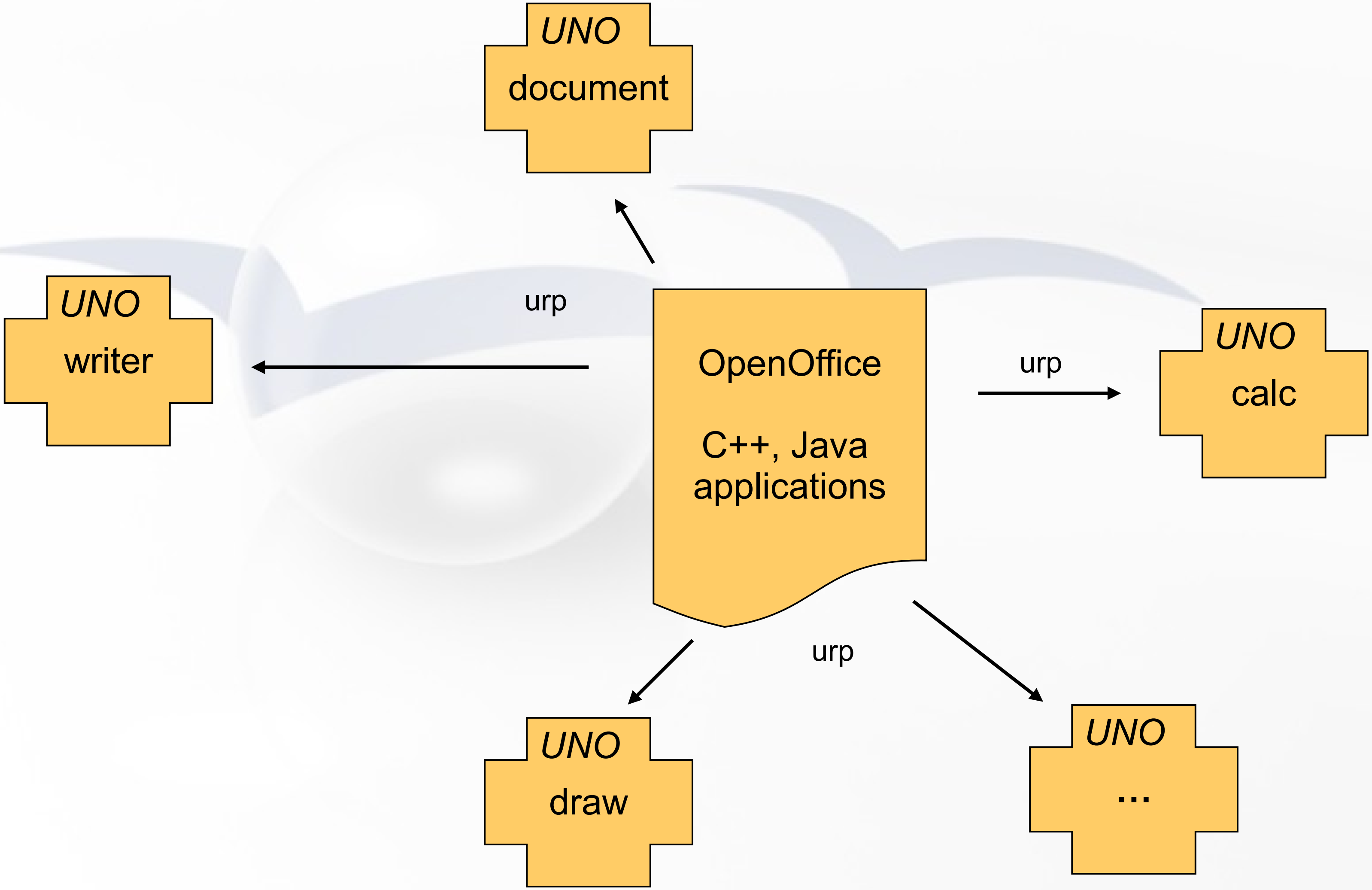
Bird Eye's View, 2

- Client-/Server-Architecture
  - Communication via TCP/IP
  - Employing distributable components (“UNO”)
    - Server can run on any computer in the world!
    - Operating systems of the server and the client are irrelevant for the purpose of communication!
  - Client may run on the same machine as the server
    - Default installation and configuration

Bird Eye's View, 3

- “UNO”
  - **U**niversal **N**etwork **O**bjects
  - Distributable, interconnected infrastructure
  - All functionality is organized in the form of classes (“UNO classes”)
  - UNO classes (types) get defined in an IDL (Interface Description Language)
- “urp”
  - **U**NO **r**emote **p**rotocol
  - CORBA-like

Bird Eye's View, 4



UNO
document

UNO
writer

urp

OpenOffice

C++, Java
applications

urp

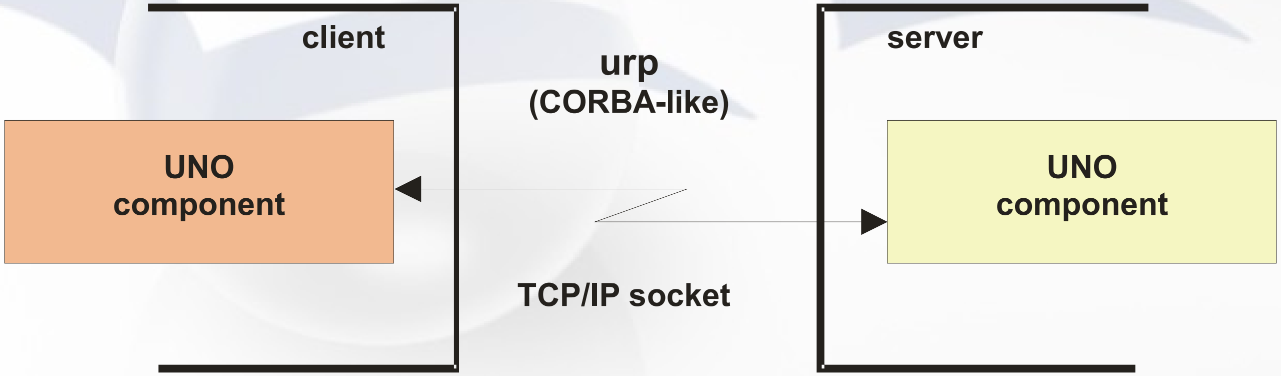
UNO
calc

urp

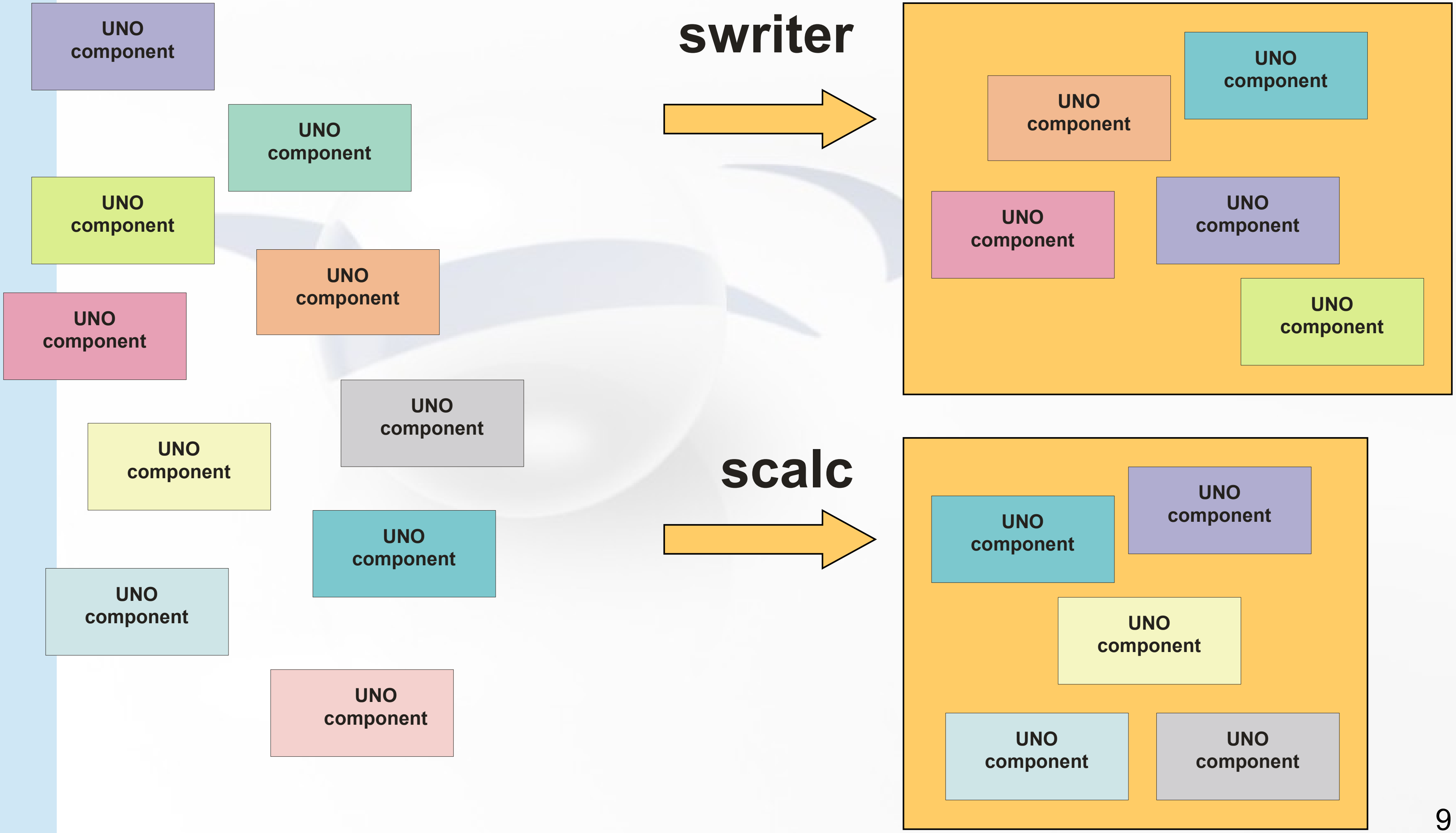
UNO
draw

UNO
…

Bird Eye's View, 5



Bird Eye's View, 6



Bird Eye's View, 7

- “Service Managers” (a.k.a. “factories”)
  - Supplied by servers
    - Also cf. XComponentContext.getServiceManager()
  - Can be used to request/create *services*
  - Returned *service* allows access to a part of the "office" functionality, e.g.
    - com.sun.star.frame.*Desktop*
    - com.sun.star.configuration.*ConfigurationProvider*
    - com.sun.star.sdb.*DatabaseContext*

Bird Eye's View, 8

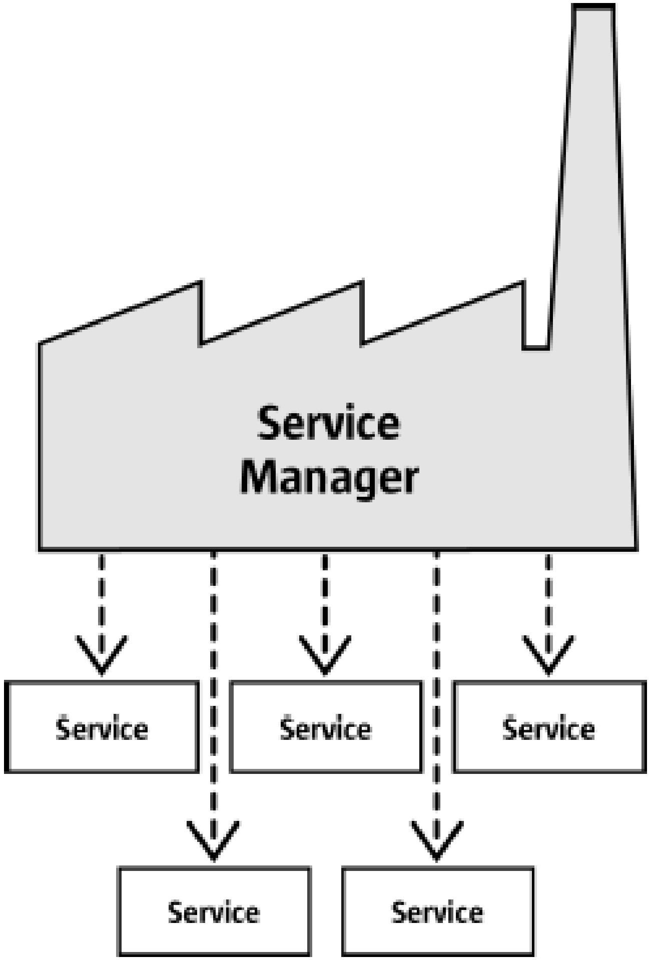


Illustration 2.1: Service manager

Bird Eye's View, 9

- “Services”
  - Can be comprehensive
  - May contain
    - "Interfaces" (group of *methods* and *attributes*)
    - Other "Services"
    - “properties” (com.sun.star.beans.*PropertyValue*)
  - Depending on the desired task you need to query (request) the appropriate interface, e.g.
    - com.sun.star.view.*XPrintable*
    - com.sun.star.frame.*XStorable*
    - com.sun.star.text.*XTextDocument*

Bird Eye's View, 10

- An example
  - Two services with seven interfaces
    - "OfficeDocument"
      - Four interfaces
    - "TextDocument"
      - Three interfaces

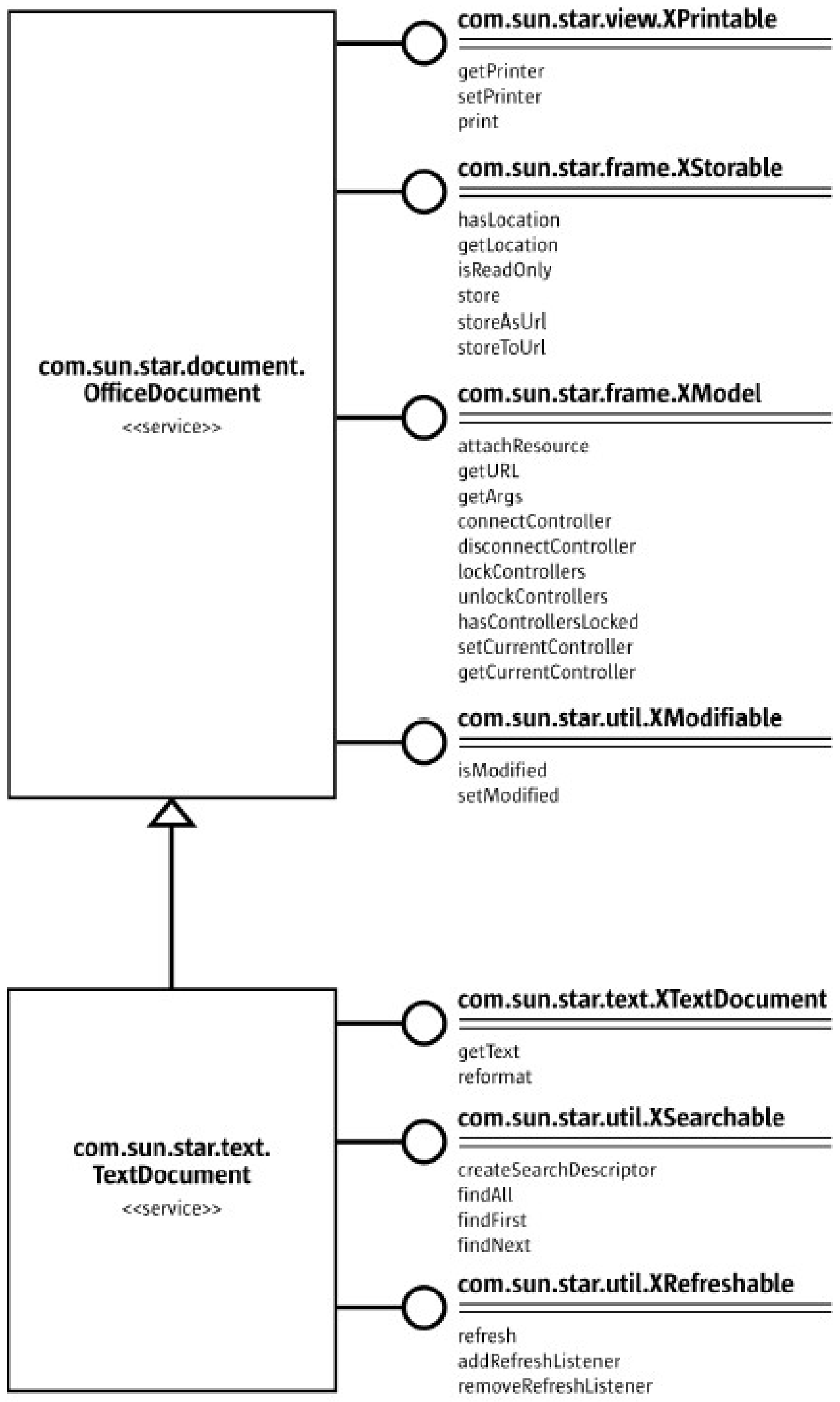


Illustration 2.3: Text Document

Bird Eye's View, 11

- **Huge** number of types
  - E.g. for Java (Apache OpenOffice 3.4.1

jar Name	Total of Types	Interfaces	%
juh.jar	47	3	6,4%
ridl.jar	469	224	47,8%
jurt.jar	98	2	2,0%
unoil.jar	2.694	1.422	52,8%
Totals	***3.308***	***1.651***	***49,9%***

Bird Eye's View, 12

- Problems for Programmers
  - *Impossible* to know each type by heart !
  - 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 Services and Interfaces is it composed of?

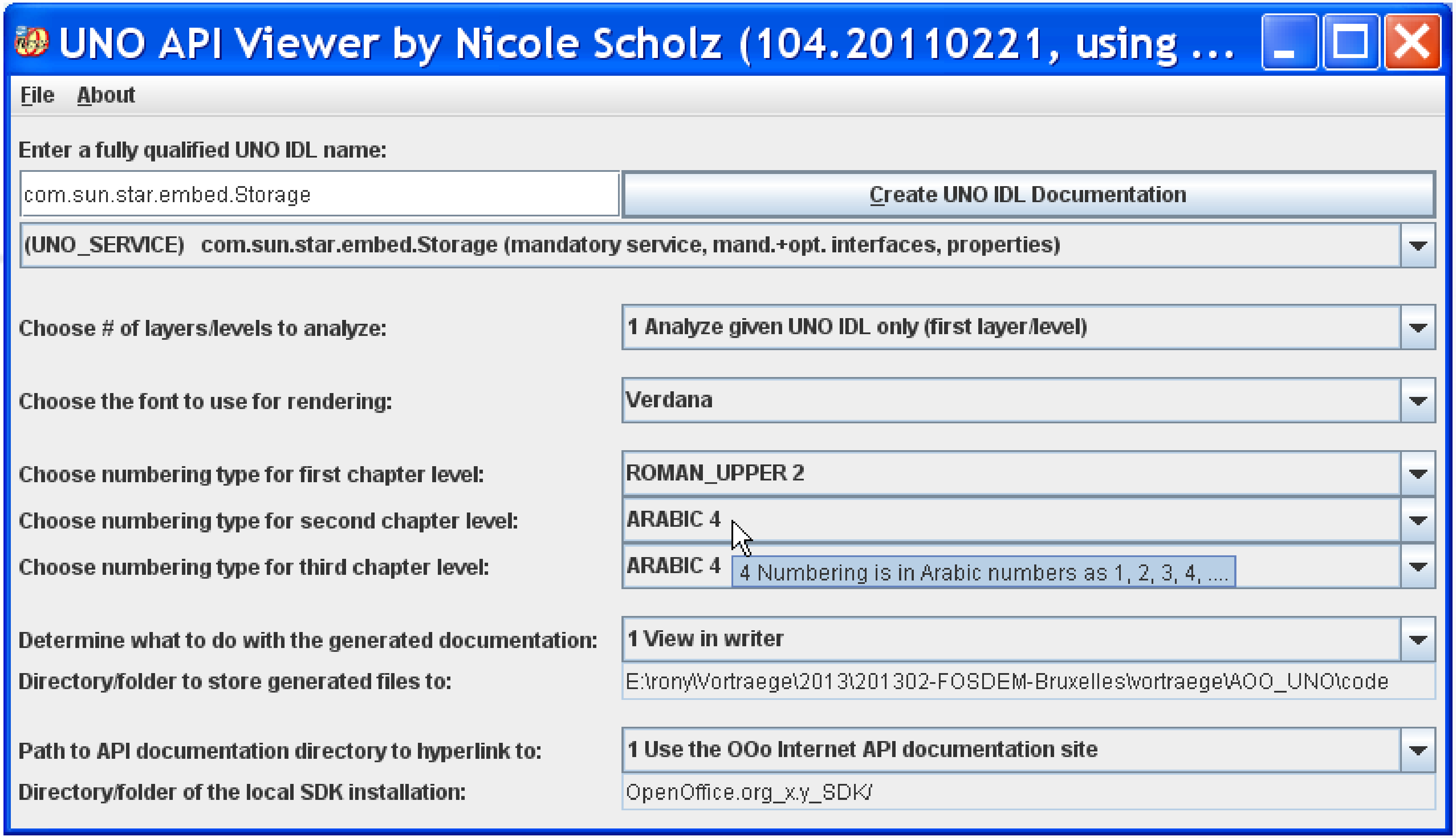
X-Ray

- AOO Basic's "X-Ray" by Bernard Marcelly
  - 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 AOO'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, 1



UNO API Viewer by Nicole Scholz (104.20110221, using ...
File About
Create UNO IDL Documentation

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

(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

Choose numbering type for third chapter level: ARABIC 4

4 Numbering is in Arabic numbers as 1, 2, 3, 4, ...

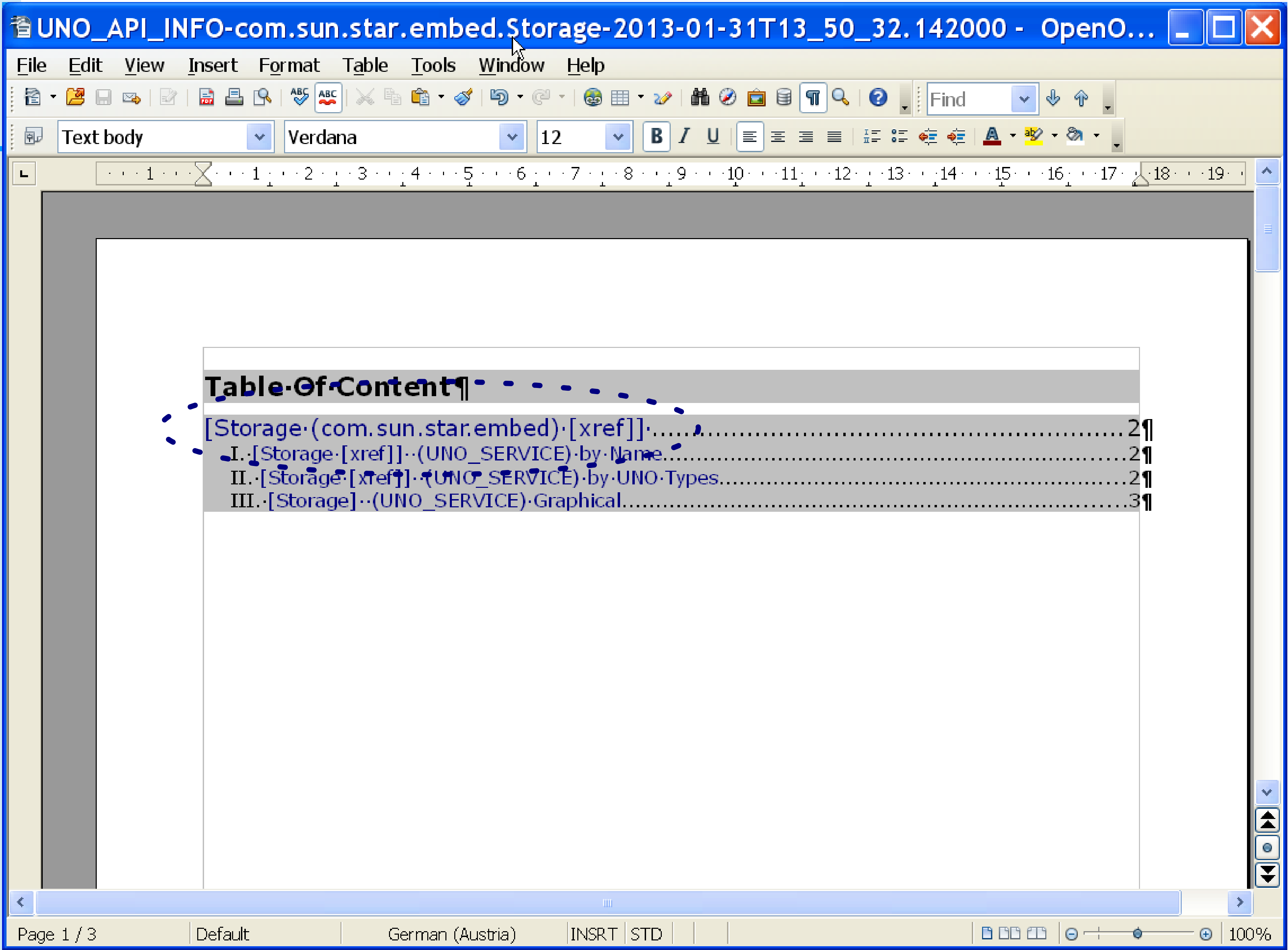
Determine what to do with the generated documentation: 1 View in writer

Directory/folder to store generated files to: E:\rony\Vortraege\2013\201302-FOSDEM-Bruxelles\vortraege\AOO_UNO\code

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 Result, 1

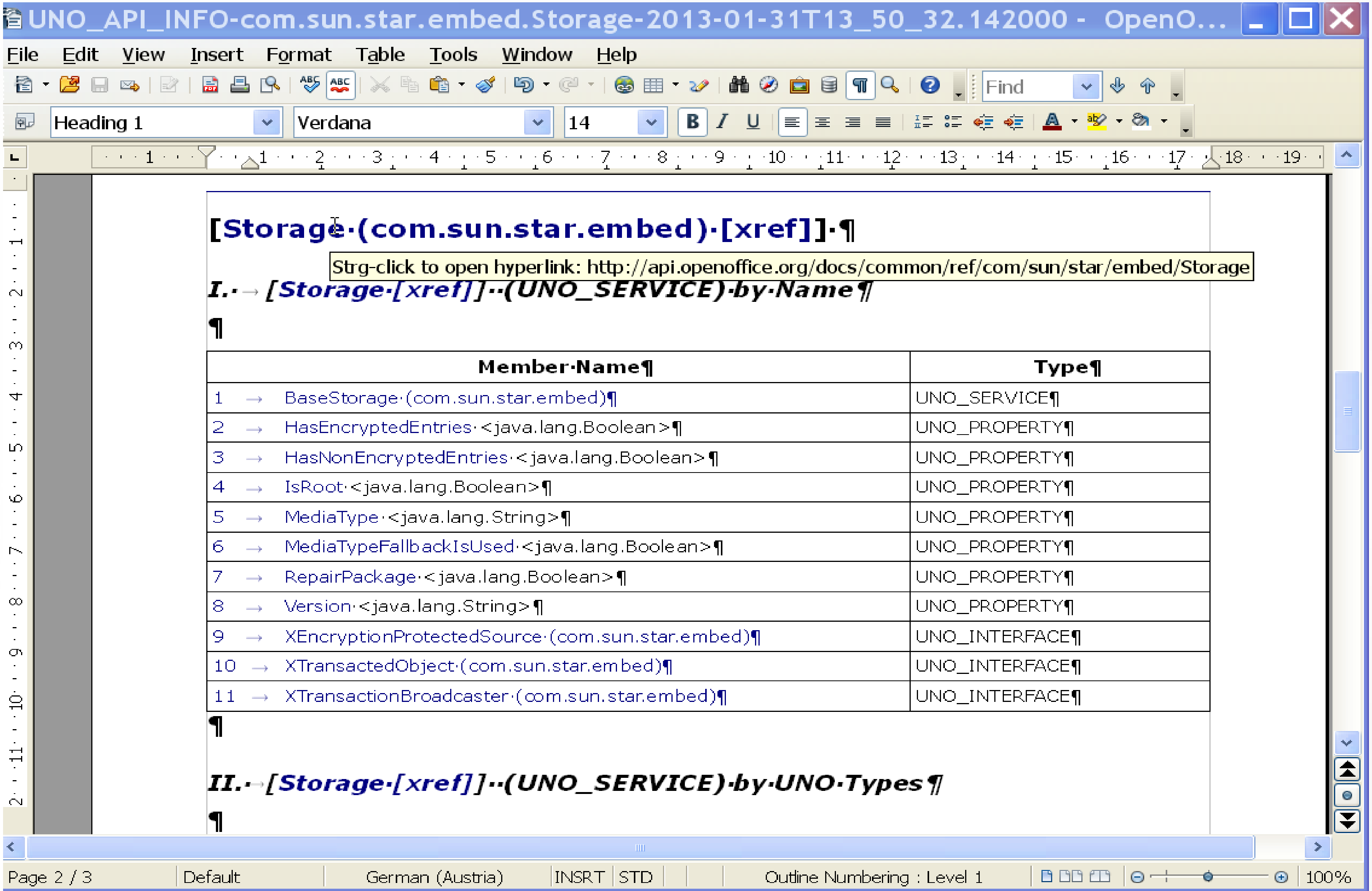


UNO_API_INFO-com.sun.star.embed.Storage-2013-01-31T13_50_32.142000 - OpenO...

Table·Of·Content¶

Page 1 / 3 | Default | German (Austria) | INSRT | STD | 100%

UNO API Info – The Result, 2



UNO_API_INFO-com.sun.star.embed.Storage-2013-01-31T13_50_32.142000 - OpenO...
File Edit View Insert Format Table Tools Window Help
Find
Heading 1
Verdana
14
Page 2 / 3
Default
German (Austria)
INSRT STD
Outline Numbering : Level 1
100%

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

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

I.· → [Storage·[xref]]··(UNO_SERVICE)·by·Name¶

¶

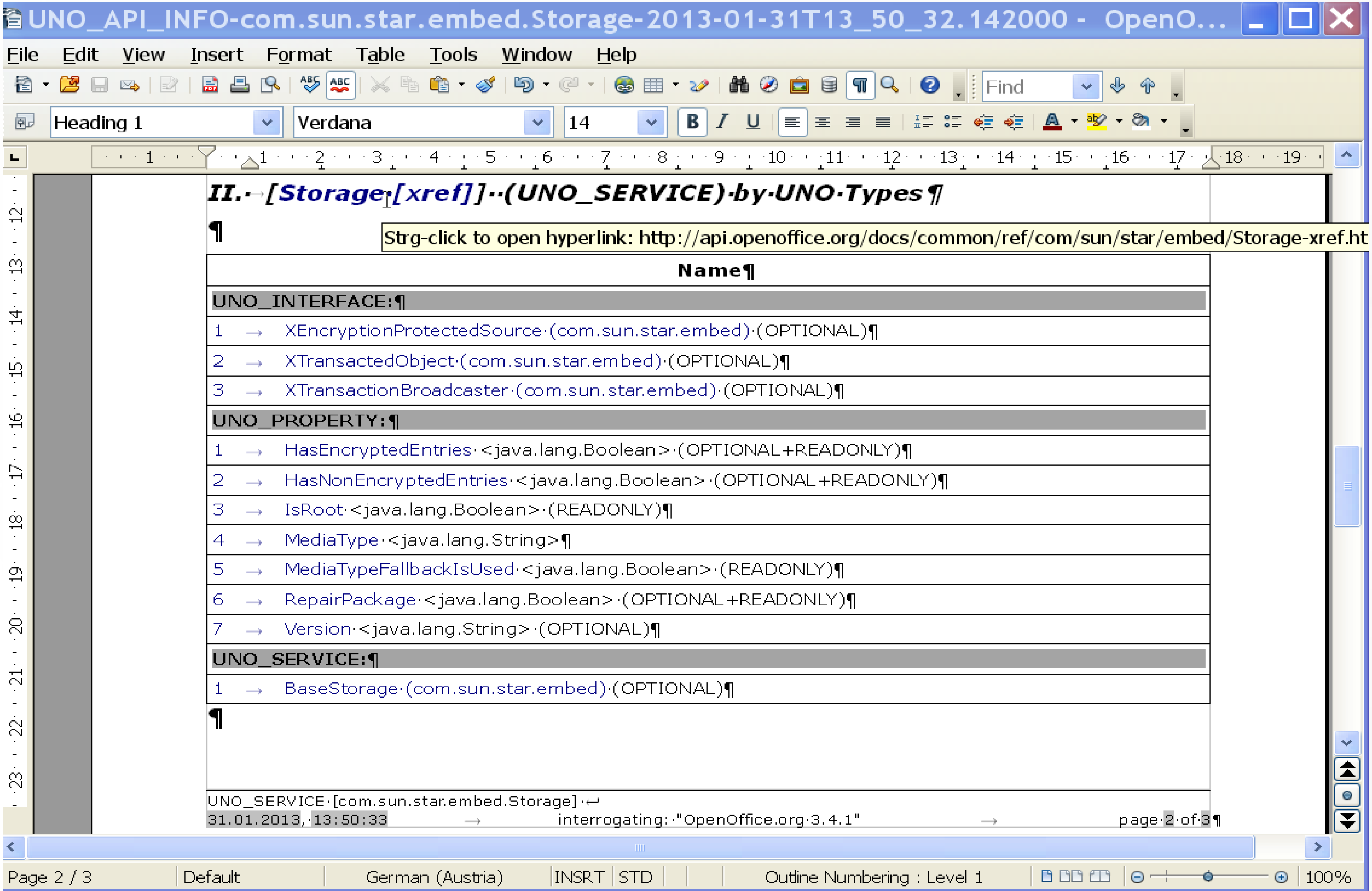
Member·Name¶	Type¶
1 → BaseStorage·(com.sun.star.embed)¶	UNO_SERVICE¶
2 → HasEncryptedEntries·<java.lang.Boolean>¶	UNO_PROPERTY¶
3 → HasNonEncryptedEntries·<java.lang.Boolean>¶	UNO_PROPERTY¶
4 → IsRoot·<java.lang.Boolean>¶	UNO_PROPERTY¶
5 → MediaType·<java.lang.String>¶	UNO_PROPERTY¶
6 → MediaTypeFallbackIsUsed·<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¶
11 → XTransactionBroadcaster·(com.sun.star.embed)¶	UNO_INTERFACE¶

¶

II.· → [Storage·[xref]]··(UNO_SERVICE)·by·UNO·Types¶

¶

UNO API Info – The Result, 3



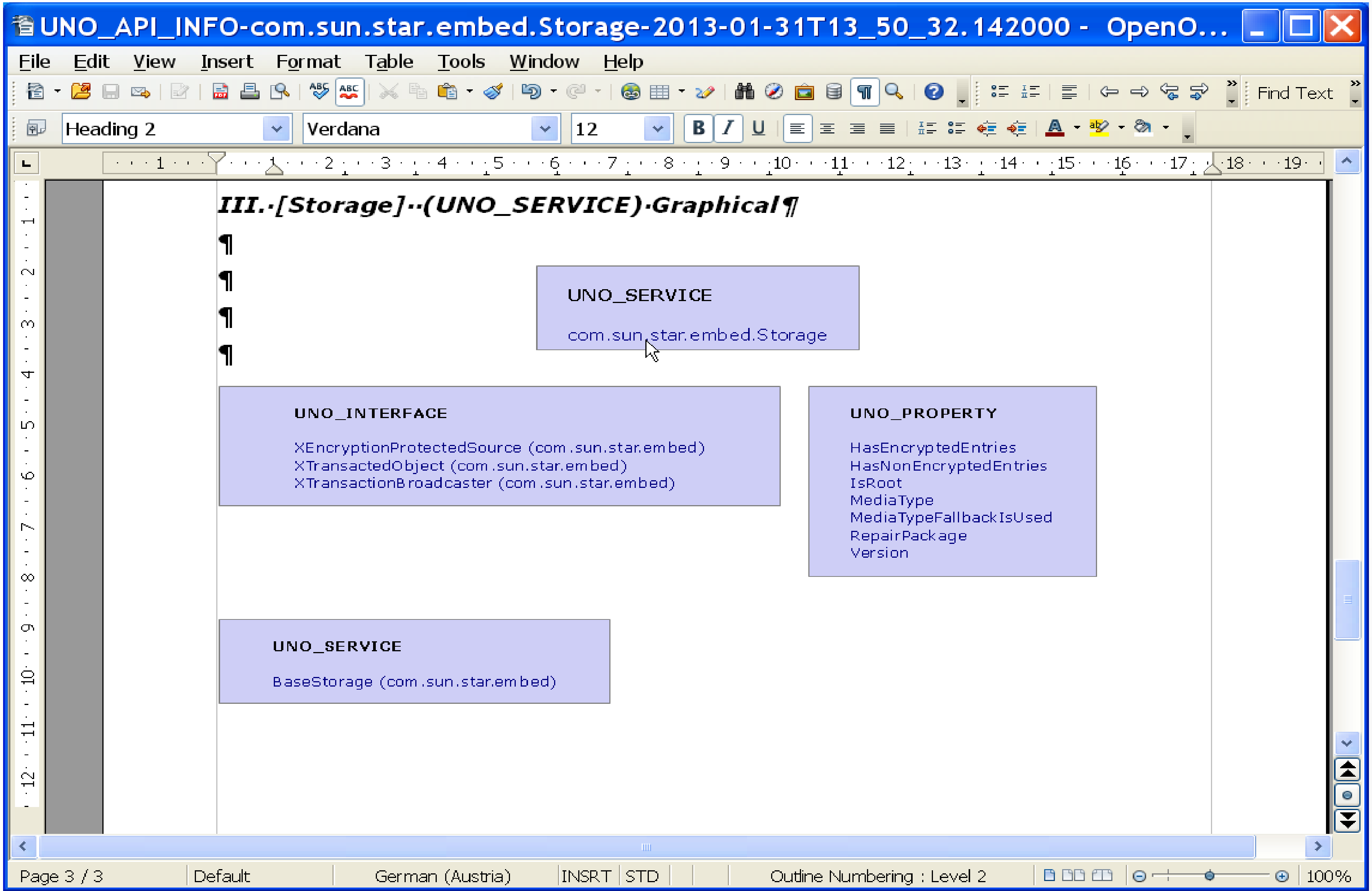
II. [Storage [xref]] (UNO_SERVICE) by UNO Types

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

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)

UNO_SERVICE [com.sun.star.embed.Storage]
31.01.2013, 13:50:33 → interrogating: "OpenOffice.org 3.4.1" → page 2 of 3

UNO API Info – The Result, 4



UNO_API_INFO-com.sun.star.embed.Storage-2013-01-31T13_50_32.142000 - OpenO...
File Edit View Insert Format Table Tools Window Help
Find Text
Heading 2
Verdana
12
Page 3 / 3
Default
German (Austria)
INSRT STD
Outline Numbering : Level 2
100%

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)

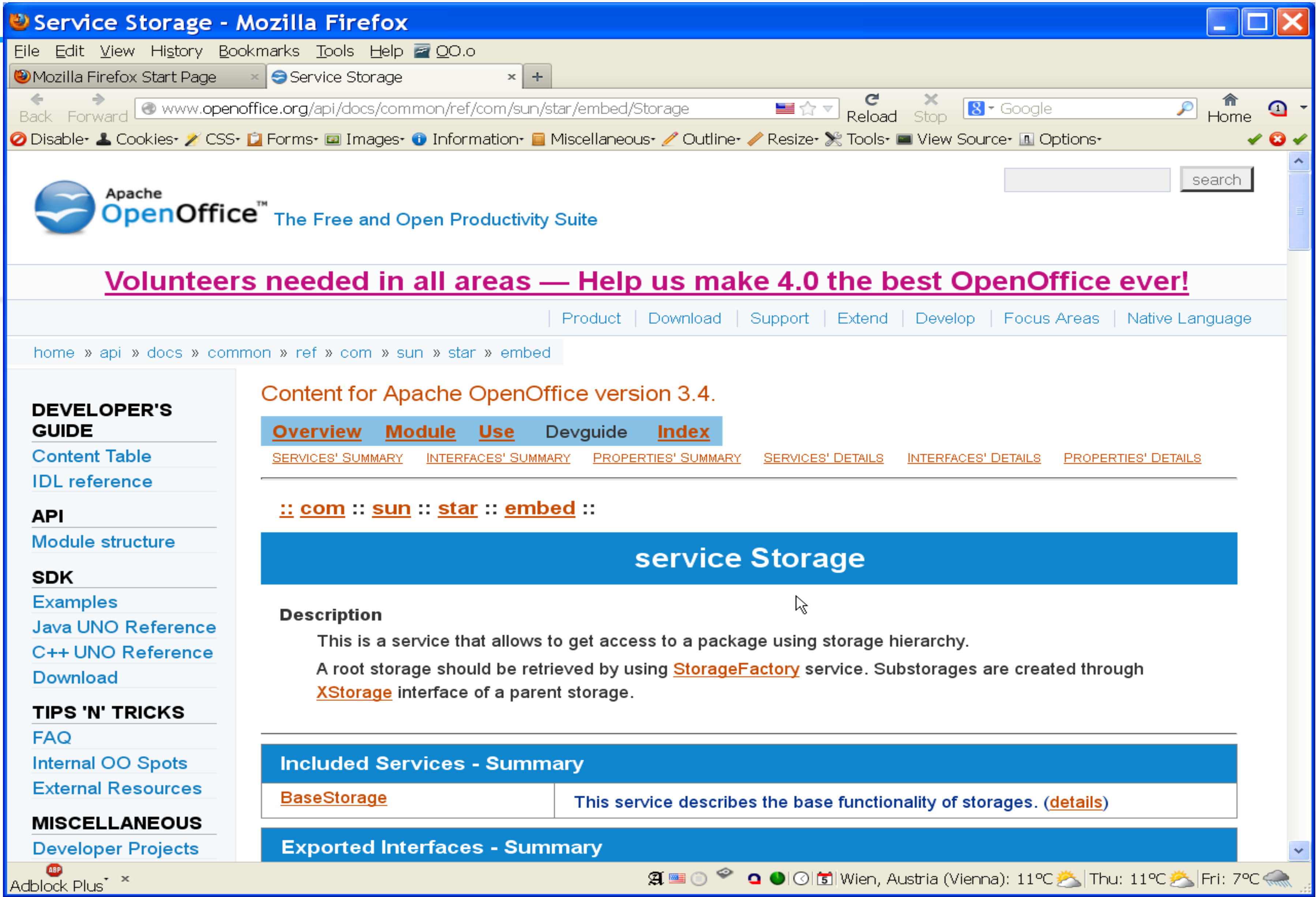
UNO_PROPERTY

HasEncryptedEntries
HasNonEncryptedEntries
IsRoot
MediaType
MediaTypeFallbackIsUsed
RepairPackage
Version

UNO_SERVICE

BaseStorage (com.sun.star.embed)

UNO API Info – Official Docs, 1


Service Storage - Mozilla Firefox
File Edit View History Bookmarks Tools Help OO.o
Mozilla Firefox Start Page
Service Storage
Back Forward
www.openoffice.org/api/docs/common/ref/com/sun/star/embed/Storage
Reload Stop
Google
Home
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options
search
Adblock Plus
Wien, Austria (Vienna): 11°C Thu: 11°C Fri: 7°C

Apache OpenOffice™ The Free and Open Productivity Suite

Volunteers needed in all areas — Help us make 4.0 the best OpenOffice ever!

| Product | Download | Support | Extend | Develop | Focus Areas | Native Language

home » api » docs » common » ref » com » sun » star » embed

DEVELOPER'S GUIDE
- Content Table
- IDL reference

API
- Module structure

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

TIPS 'N' TRICKS
- FAQ
- Internal OO Spots
- External Resources

MISCELLANEOUS
- Developer Projects

Content for Apache OpenOffice version 3.4.

Overview Module Use Devguide Index

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

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

service Storage

Description

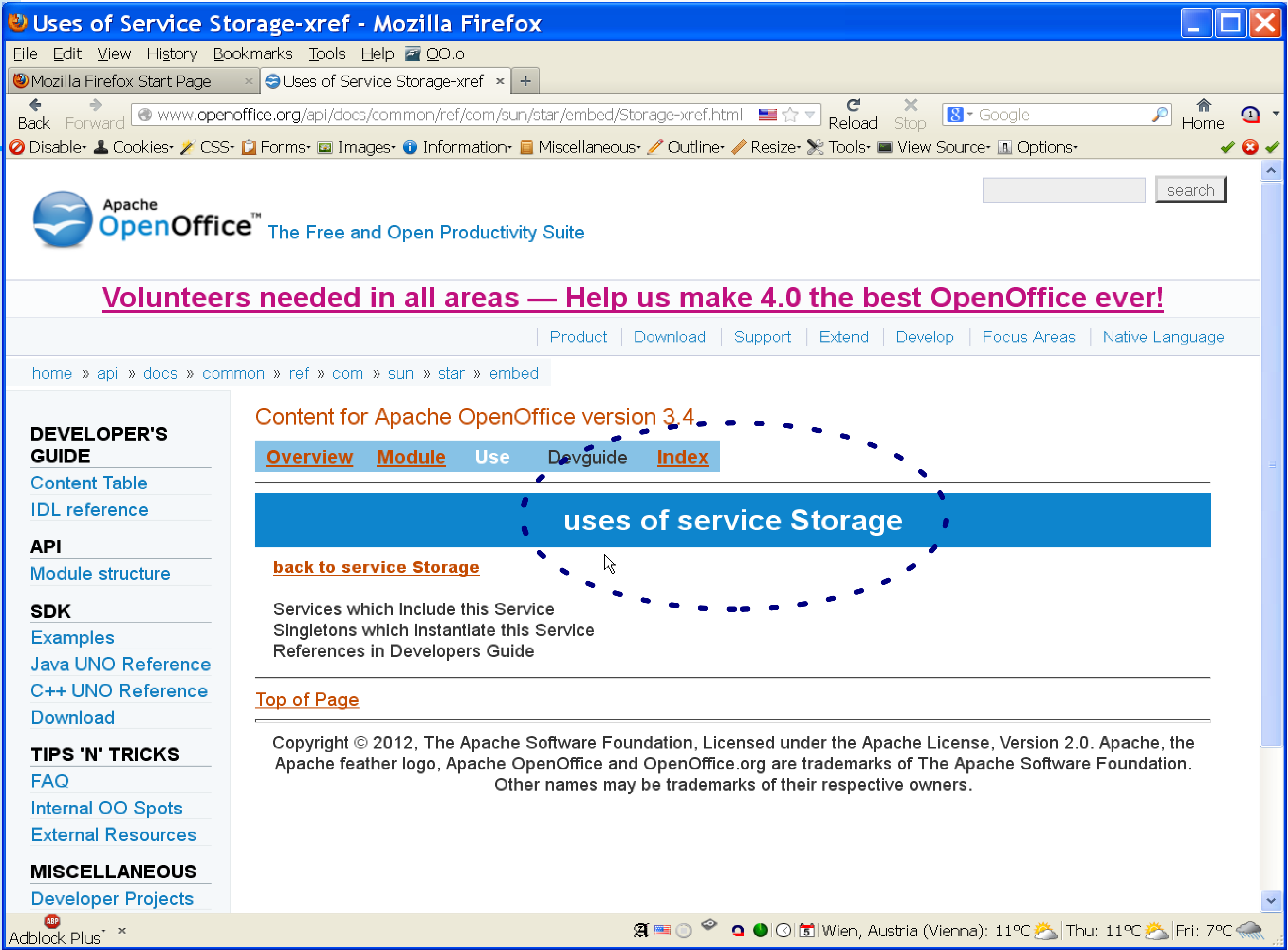
This is a service that allows to get access to a package using storage hierarchy.

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

UNO API Info – Official Docs, 2



Uses of Service Storage-xref - Mozilla Firefox
File Edit View History Bookmarks Tools Help OO.o
Mozilla Firefox Start Page
Uses of Service Storage-xref
Back Forward
www.openoffice.org/api/docs/common/ref/com/sun/star/embed/Storage-xref.html
Reload Stop Google Home
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options
search
Adblock Plus
Wien, Austria (Vienna): 11°C Thu: 11°C Fri: 7°C

Apache OpenOffice™ The Free and Open Productivity Suite

Volunteers needed in all areas — Help us make 4.0 the best OpenOffice ever!

Product | Download | Support | Extend | Develop | Focus Areas | Native Language

home » api » docs » common » ref » com » sun » star » embed

DEVELOPER'S GUIDE
- Content Table
- IDL reference

API
- Module structure

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

TIPS 'N' TRICKS
- FAQ
- Internal OO Spots
- External Resources

MISCELLANEOUS
- Developer Projects

Content for Apache OpenOffice version 3.4

Overview Module Use Devguide Index

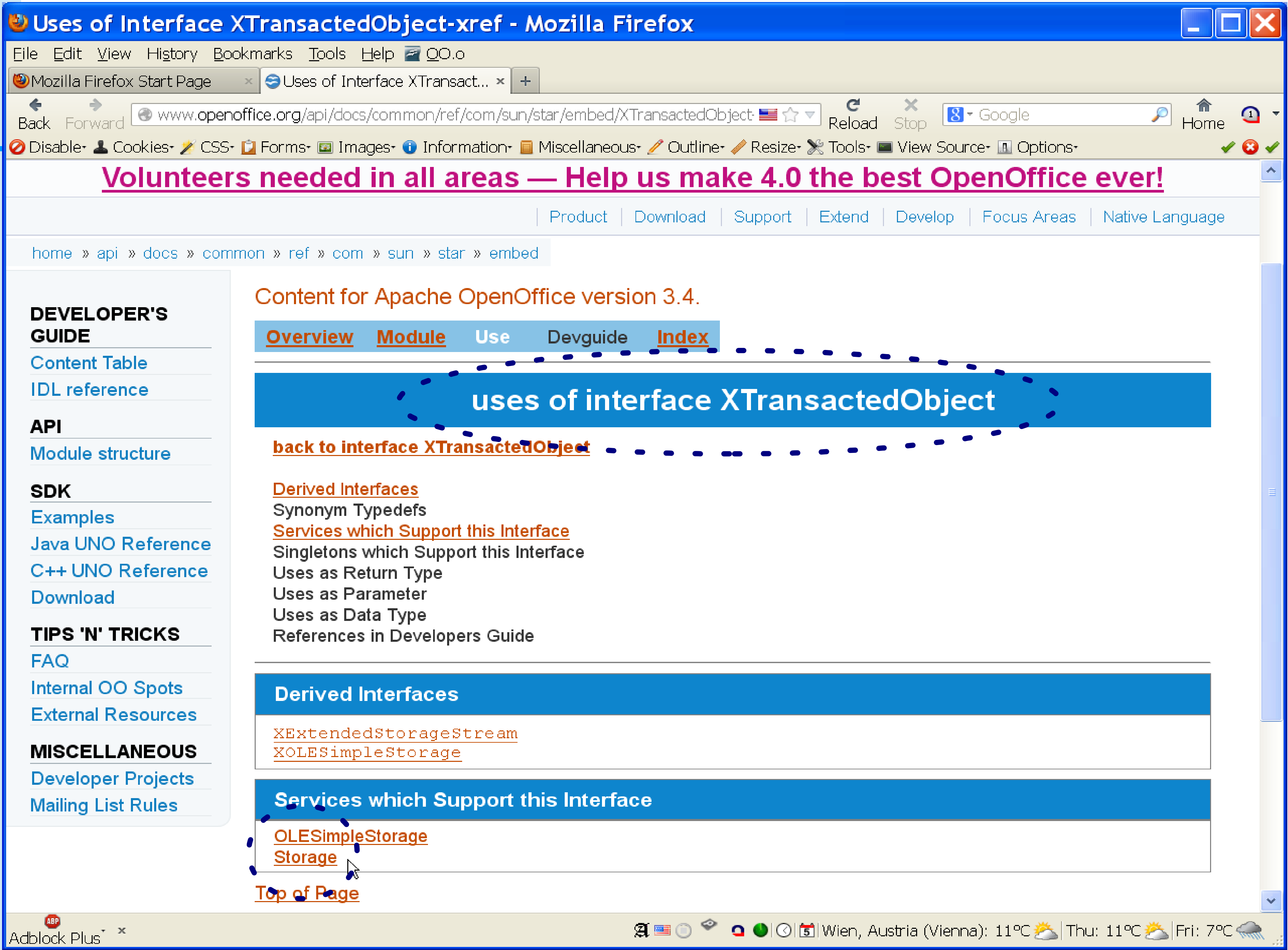
uses of service Storage

back to service Storage

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

Top of Page

UNO API Info – Official Docs, 3



Uses of Interface XTransactedObject-xref - Mozilla Firefox

File Edit View History Bookmarks Tools Help OO.o
Mozilla Firefox Start Page
Uses of Interface XTransact...
Back Forward Reload Stop Home
www.openoffice.org/api/docs/common/ref/com/sun/star/embed/XTransactedObject
Google
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options
Adblock Plus
Wien, Austria (Vienna): 11°C Thu: 11°C Fri: 7°C

Volunteers needed in all areas — Help us make 4.0 the best OpenOffice ever!

Product | Download | Support | Extend | Develop | Focus Areas | Native Language

home » api » docs » common » ref » com » sun » star » embed

DEVELOPER'S GUIDE
- Content Table
- IDL reference

API
- Module structure

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

TIPS 'N' TRICKS
- FAQ
- Internal OO Spots
- External Resources

MISCELLANEOUS
- Developer Projects
- Mailing List Rules

Content for Apache OpenOffice version 3.4.

Overview Module Use Devguide Index

uses of interface XTransactedObject

back to interface XTransactedObject

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

Derived Interfaces

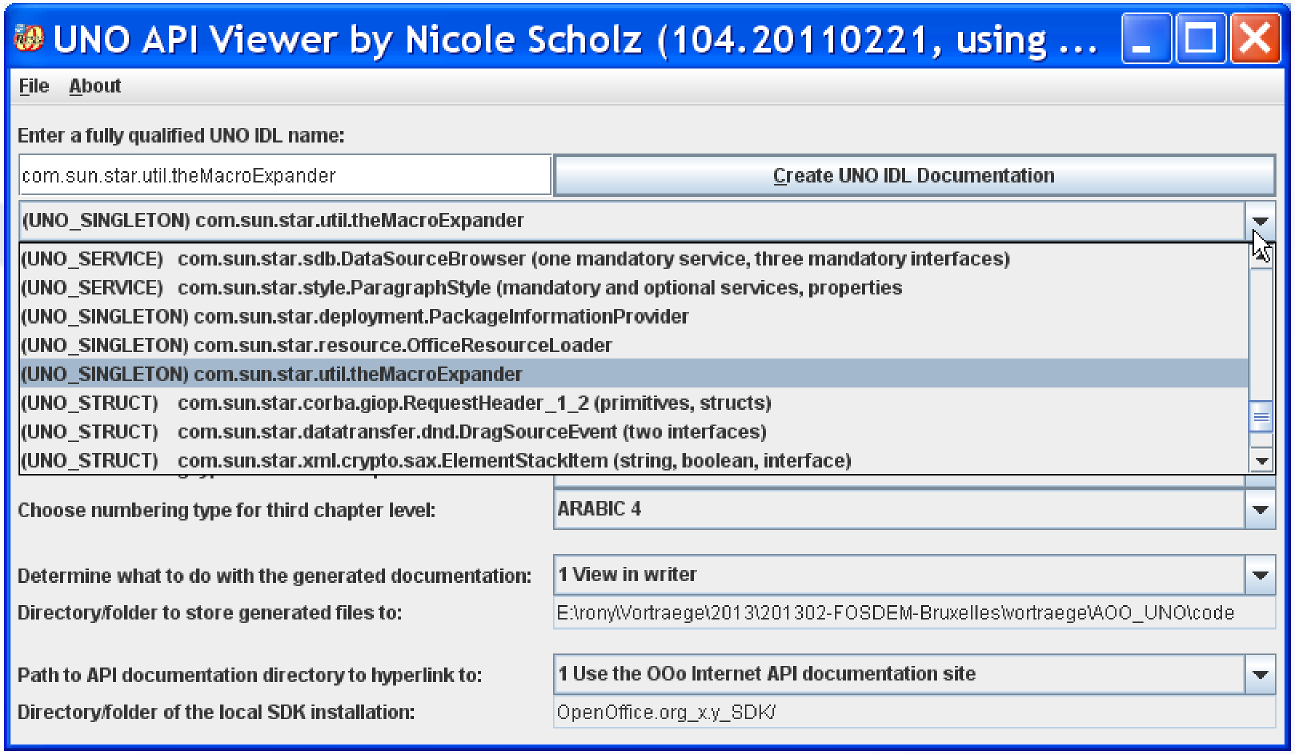
XExtendedStorageStream
XOLESimpleStorage

Services which Support this Interface

OLESimpleStorage
Storage

Top of Page

UNO API Info – The GUI, 2


UNO API Viewer by Nicole Scholz (104.20110221, using ...
File
About
Create UNO IDL Documentation

Enter a fully qualified UNO IDL name: com.sun.star.util.theMacroExpander

(UNO_SINGLETON) com.sun.star.util.theMacroExpander

(UNO_SERVICE) com.sun.star.sdb.DataSourceBrowser (one mandatory service, three mandatory interfaces)
(UNO_SERVICE) com.sun.star.style.ParagraphStyle (mandatory and optional services, properties
(UNO_SINGLETON) com.sun.star.deployment.PackageInformationProvider
(UNO_SINGLETON) com.sun.star.resource.OfficeResourceLoader
(UNO_SINGLETON) com.sun.star.util.theMacroExpander
(UNO_STRUCT) com.sun.star.corba.giop.RequestHeader_1_2 (primitives, structs)
(UNO_STRUCT) com.sun.star.datatransfer.dnd.DragSourceEvent (two interfaces)
(UNO_STRUCT) com.sun.star.xml.crypto.sax.ElementStackItem (string, boolean, interface)

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\Vortraege\2013\201302-FOSDEM-Bruxelles\vortraege\AOO_UNO\code

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/

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://sourceforge.net/projects/bsf4oorexx/files/GA/
  - Installation all in all: 60 seconds
- *Any* AOO programming language can use it via the dispatch interface
  - "com.sun.star.frame.XDispatchProvider"

Using "UNO API Info" From AOO Basic, 1/2

```
' demonstrates how to use "UNO_API_info.rxo" from AOO 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, AOO 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 AOO Basic, 2/2

```
' 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, 1/4

```java
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 AOO 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, 2/4

```
// 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, AOO 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, 3/4

```
// ============================== 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, 4/4

```
... 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, 1/3

```
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, 2/3

```
// 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, 3/3

```
// 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, 1/2

```
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, AOO 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, 2/2

```
   -- 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 öptions 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, 1/2

```
from com.sun.star.beans import PropertyValue
def createApiInfo( ):
   """Uses the ooRexx macro UNO_API_info.rxo to document an AOO service object and a UNO IDL type string"""
# get the AOO 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, 2/2

```
# --- 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 AOO-UI
g_exportedScripts = createApiInfo,
```

Roundup

- "UNO_API_Info.rxo"
  - A powerful and great 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 AOO* programming languages can take advantage of it!
  - Allows you to create Writer and PDF documents that are fully linked to the official AOO documentation on the Internet!

Links to ooRexx/BSF4ooRexx

- ooRexx (as of 2013-01-31, version: 4.1.2)
  - An easy to learn and easy to use scripting language
    - Compatible to ("classic") Rexx
    - Developped originally by IBM ("Object REXX")
  - Source code was received by the non-for-profit SIG "Rexx Language Association (http://www.RexxLA.org)"
    - Opensourced as "Open Object Rexx (ooRexx)"
  - Home: http://www.ooRexx.org
  - Downloads: http://sourceforge.net/projects/oorexx/files/oorexx/
  - Brief overview (since opensourcing a lot got added): http://wi.wu.ac.at/rgf/rexx/misc/ecoop06/ECOOP2006_RDL_Workshop_Flatscher_Paper.pdf
  - Authoring a new book that introduces ooRexx

Links to ooRexx/BSF4ooRexx

- BSF4ooRexx (with built-in AOO/LO support)
  - Allows to use all of Java from ooRexx as if it was an interpreted, typeless and caseless language!
    - “Camouflaging Java as ooRexx” (package “BSF.CLS”)
      - All Java classes and Java objects look like ooRexx' ones!
    - Includes specific AOO support (package “UNO.CLS”)
  - Developed since 2000 to allow the creation of platform independent Rexx and ooRexx scripts
    - Using Apache's “Bean Scripting Framework (BSF)”, cf. http://commons.apache.org/bsf/
  - Home: http://sourceforge.net/projects/bsf4oorexx/
  - Downloads: http://sourceforge.net/projects/bsf4oorexx/files/GA/