

BSF4ooRexx

Scripting Apache OpenOffice (AOO) and LibreOffice (LO)
using Universal Network Objects (UNO)

Business Programming 2



BSF4ooRexx



NetRexx

Windows
GUIs
(AWT)

Sockets
SSL/TLS

XML
SAX/DOM
JSON

Scripting
AOO/LO
(UNO)

Rexx
Script
Engine

Portable
GUIs
(JavaFX)

Java Web
Server
(Tomcat)

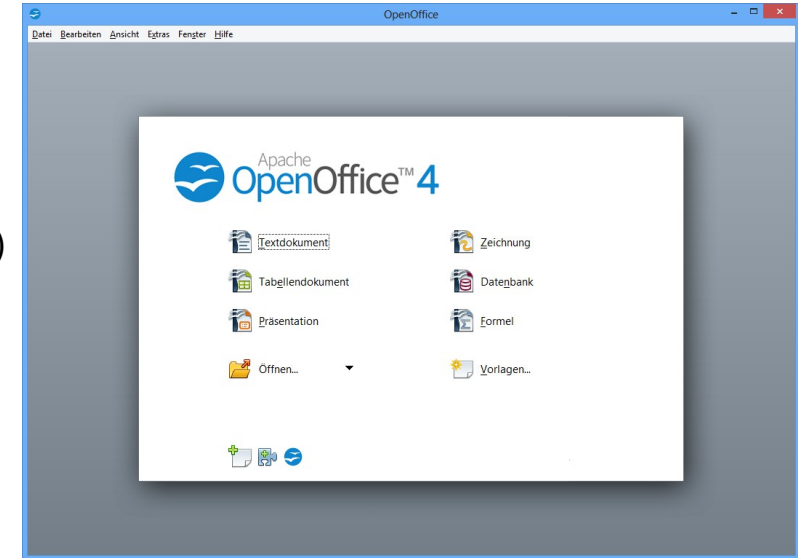
Java Classes
written in Rexx
style

- StarOffice
 - Originates in Germany
 - StarDivision, est. 1985
 - Portable C++ class library ("Star")
 - Allow creation of a portable integrated office suite – Goal:
 - Compatibility with MS Office, 1995
 - Read and write MS Office files
 - 1990s
 - OS/2, Windows
 - Slow hardware, small bandwidths!



StarOffice 9.1.0 (Windows 7)

- StarOffice → OpenOffice
 - 1998 bought by Sun
 - StarOffice 5.1 → **OpenOffice.org** 1.0 (2002)
 - 2010 bought by Oracle
 - Oracle OpenOffice
 - 2011 donated to ASF (Apache Software Foundation)
 - **Apache OpenOffice** (AOO), incubating
 - First release of AOO 3.4 (May 2012)
 - 3.4.1 AOO graduates at ASF! (October 2012)
- As of 2022-12-11
 - OpenOffice 4.1.13
 - <<https://www.openoffice.org/>>

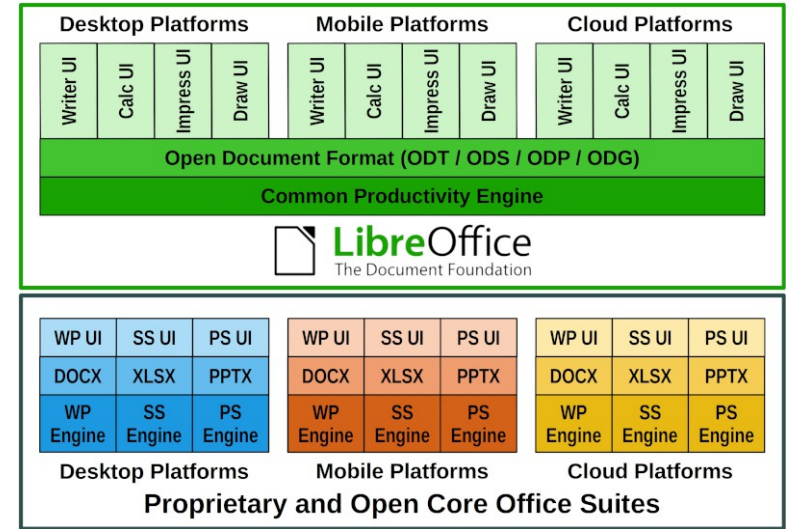


OpenOffice 4.1.13

History, 3



- OpenOffice.org → **LibreOffice**(LO)
 - 2010 forked from OpenOffice.org
 - Currently: LibreOffice 7.4
 - <<https://www.libreoffice.org/>> (2022-12-11)
- OpenOffice.org → NeoOffice (Mac only)
 - 2003 commercial fork of OpenOffice.org
 - Currently: NeoOffice 4.4
 - <<https://www.neooffice.org/>> (2022-12-11)



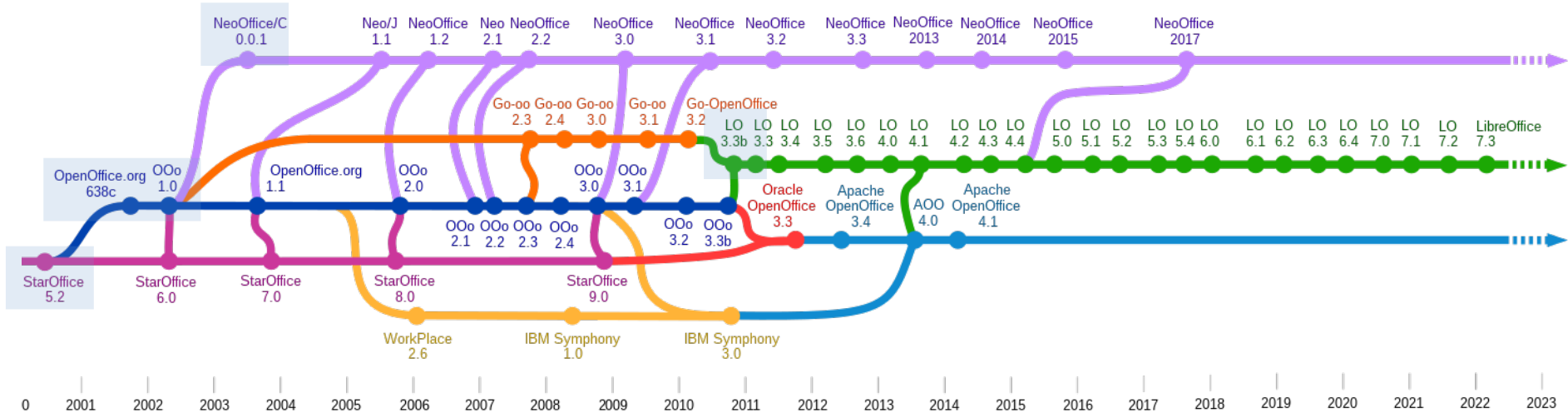
LibreOffice

Overview

History, 4



- Forks
 - taking a copy of source code and start independent development project



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

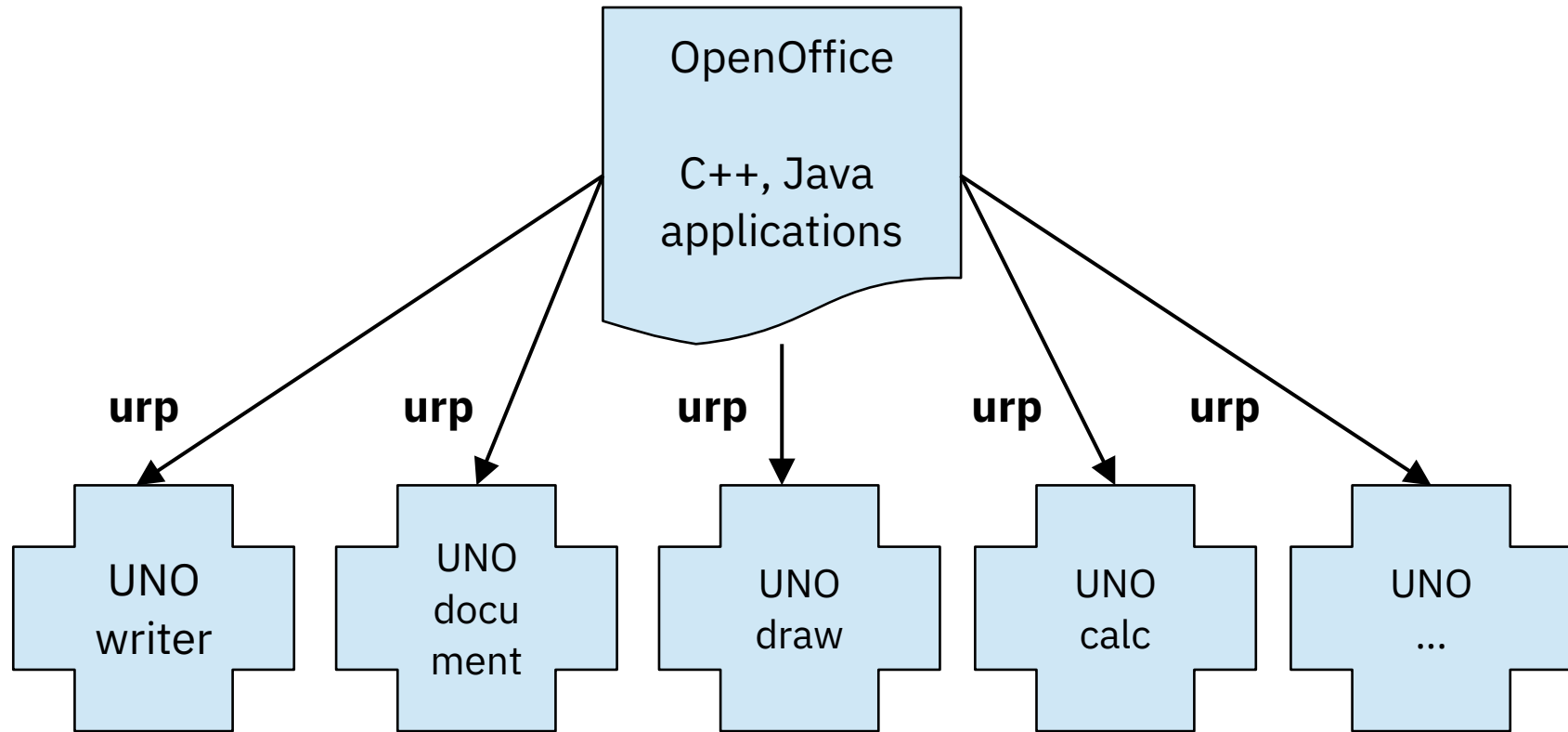


- Client-/Server-Architecture
 - Communication via TCP/IP
 - Employing distributable components (“UNO”)
 - Server can run on any computer in the world!
 - Operating systems of the server and the client are irrelevant for the purpose of communication!
 - Client may run on the same machine as the server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

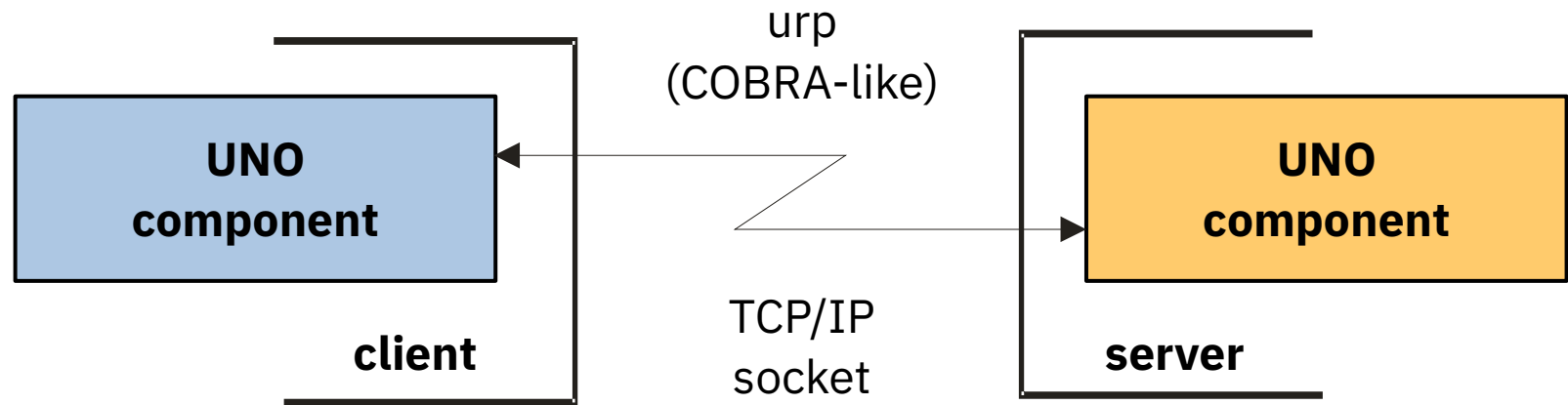
Bird Eye's View, 3



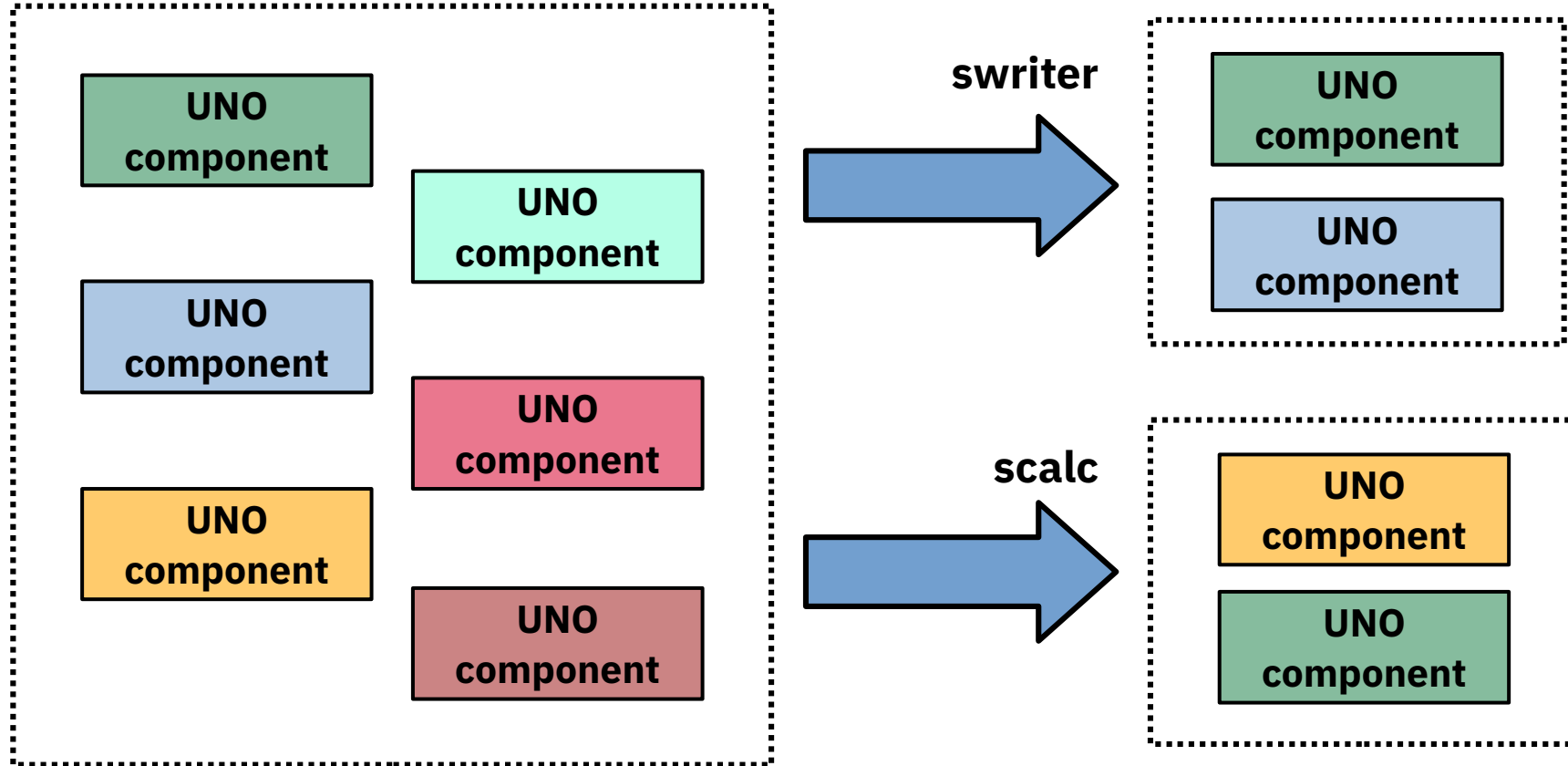
- “UNO”
 - **U**niversal **N**etwork **O**bjects
 - Distributable, interconnected infrastructure
 - All functionality is organized in the form of classes (“UNO classes”)
 - UNO classes (types) get defined in an IDL (**I**nterface **D**escription **L**anguage)
- “urp”
 - **U**NO remote **p**rotocol
 - CORBA-like
 - **C**ommon **O**bject **R**equest **B**roker **A**rchitecture



Bird Eye's View, 5



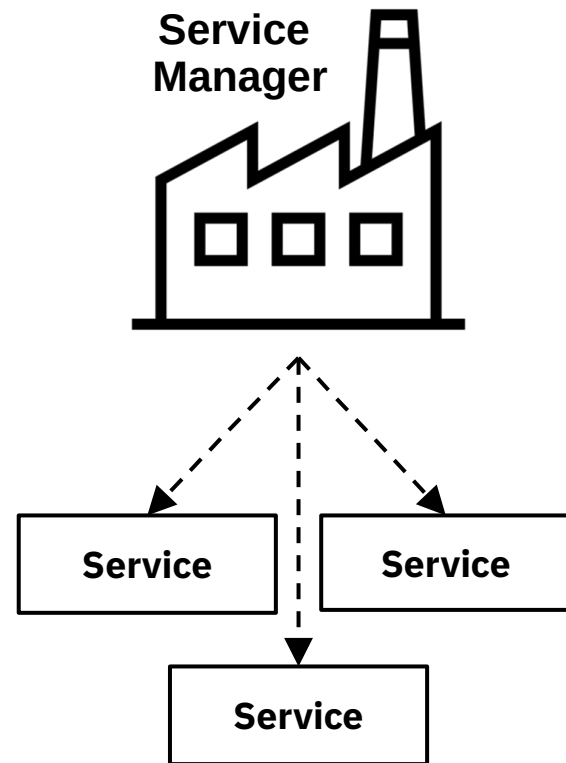
Bird Eye's View, 6



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`



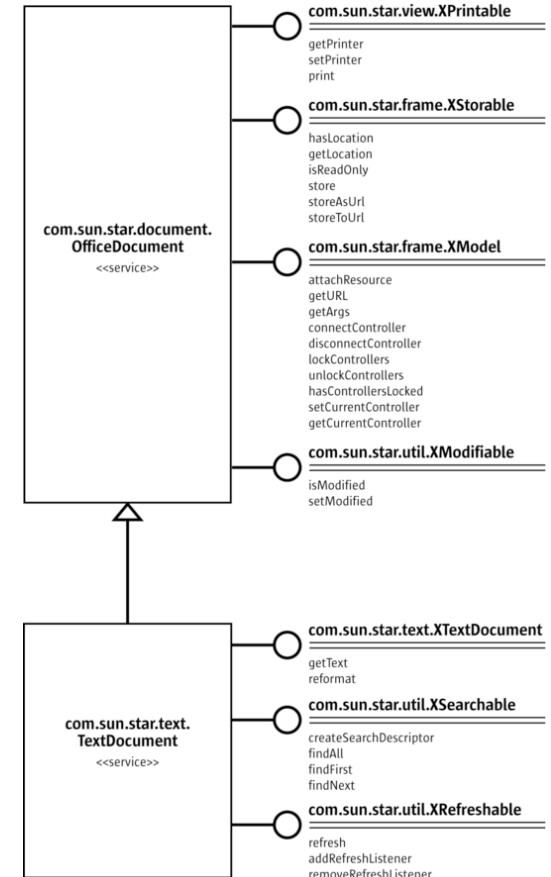
Bird Eye's View, 8



- “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, 9 – An Example

- Two services with seven interfaces
 - "OfficeDocument"
 - Four interfaces
 - `com.sun.star.view.XPrintable`
 - `com.sun.star.frame.XStorable`
 - `com.sun.star.frame.XModel`
 - `com.sun.star.util.XModifiable`
 - "TextDocument"
 - Three interfaces
 - `com.sun.star.text.XTextDocument`
 - `com.sun.star.util.XSearchable`
 - `com.sun.star.util.XRefreshable`



Programming Languages

- Programming languages
 - C++ (queryInterface)
 - Java (queryInterface)
 - Basic (implicit queryInterface)
 - Python (implicit queryInterface)
- Java-based scripting framework
 - BeanShell (queryInterface)
 - JavaScript (queryInterface)
 - ooRexx (queryInterface)
 - ...

Basic UNO Datatypes

Basic UNO Datatype	Java Datatype
UNO_ANY	com.sun.star.uno.Any or java.lang.Object
UNO_VOID	void
UNO_BOOLEAN	boolean
UNO_BYTE (8-bit)	byte
UNO_CHAR (16-bit)	char
UNO_SHORT (16-bit)	short
UNO_UNSIGNED_SHORT (16-bit)	short
UNO_LONG (32-bit)	int
UNO_UNSIGNED_LONG (32-bit)	int
UNO_HYPER (64-bit)	long
UNO_UNSIGNED_HYPER (64-bit)	long
UNO_FLOAT	float
UNO_DOUBLE	double

UNO Types/Classes, 1

- IDL
 - **I**nterface **d**escription **l**anguage
 - Text based definition of UNO types
 - Can be reflected at runtime
- UNO Types/Classes (in alphabetical order)
 - *UNO Constants*, members:
 - Fields, usually of the same UNO datatype
 - *UNO Enum*, members:
 - Fields are always of type UNO_LONG (32-Bit integers)

UNO Types/Classes, 2

- UNO Types/Classes (continued)
 - *UNO Exception*, members:
 - Fields of any datatype
 - *UNO Interface*, members:
 - UNO Methods
 - UNO Attributes
 - *UNO Module*, members:
 - Any UNO Type/Class
 - Name of the module(s) are denoted in the fully qualified name of an UNO type, e.g.
 - **com.sun.star.beans**.PropertyValue



- UNO Types/Classes (continued)
 - *UNO Service*, members:
 - *UNO Interfaces*
 - *UNO Services*
 - *UNO Properties* ([com.sun.star.beans.PropertyValue](#))
 - Regarded as a set ([com.sun.star.beans.XPropertySet](#))
 - *UNO Singleton*
 - *UNO Struct*, members:
 - Fields only!
 - *UNO Typedef*



- Extremely important
 - Wealth of services and interfaces
 - Created in pure German ;) engineering style
 - To miss the forest for the trees!
- AOO API documentation
 - <http://www.openoffice.org/api/> (2022-12-11)
 - Developer's guide, API wiki, UNO wiki, extensions, examples, tutorials
 - <http://www.openoffice.org/api/docs/common/ref/com/sun/star/module-ix.html> (2022-12-11)
 - Extensive, HTML-linked API reference
 - Use its Index to locate services, interfaces, etc.

Scripting AOO

Documentation, 2



Apache OpenOffice® The Free and Open Productivity Suite

Released: Apache OpenOffice 4.1.12

home » api » docs » common » ref » com » sun » star » Product » Download » Support » Blog » Extensions & Templates » Get Involved » Focus Areas » Native Languages

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

Overview Module Use Devguide Index

Nested Modules

accessibility
animations
auth
awt
beans
bridge
chart
chart2
configuration
connection
container
corba

Overview Module Use Devguide Index

Nested Modules

accessibility
animations
auth
awt
beans
bridge
chart
chart2
configuration
connection
container
corba

Apache OpenOffice® The Free and Open Productivity Suite

Released: Apache OpenOffice 4.1.12

home » api » docs » common » ref » index-files » Product » Download » Support » Blog » Extensions & Templates » Get Involved » Focus Areas » Native Languages

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

Overview Module Use Devguide Index

Global Index A

ABCDEFGHIJKLMNOPQRSTUVWXYZ_

A - constant in constants group :com::sun::star::awt::Key
aArgs - field in struct :com::sun::star::frame::DispatchStatement
abbreviateString() - function in interface :com::sun::star::util::XStringAbbreviation
ABBREVIATION - constant in constants group :com::sun::star::linguistic2::ConversionPropertyType
AbbrevName - field in struct :com::sun::star::i18n::CalendarItem
aBitmapMode - field in struct :com::sun::star::chart2::FillBitmap
ABORT - value in enum :com::sun::star::ucb::IOErrorCode
abort() - function in interface :com::sun::star::ucb::XCommandProcessor
Aborted - property in service :com::sun::star::document::MediaDescriptor
aborted() - function in interface :com::sun::star::sheet::XRangeSelectionListener
abortRangeSelection() - function in interface :com::sun::star::sheet::XRangeSelection
ABOVE - constant in constants group :com::sun::star::awt::FontEmphasisMark
AboveCenter - constant in constants group :com::sun::star::awt::ImagePosition
AboveLeft - constant in constants group :com::sun::star::awt::ImagePosition
AboveRight - constant in constants group :com::sun::star::awt::ImagePosition
ABOVE_WORD - constant in constants group :com::sun::star::i18n::reservedWords
ABSOLUTE - constant in constants group :com::sun::star::chart::ErrorBarStyle
ABSOLUTE - value in enum :com::sun::star::util::SearchAlgorithms

Apache OpenOffice® The Free and Open Productivity Suite

Released: Apache OpenOffice 4.1.12

home » api » docs » common » ref » index-files » Product » Download » Support » Blog » Extensions & Templates » Get Involved » Focus Areas » Native Languages

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

Overview Module Use Devguide Index

Global Index X

ABCDEFGHIJKLMNOPQRSTUVWXYZ_

X - field in struct :com::sun::star::geometry::RealPoint2D
X - constant in constants group :com::sun::star::awt::PosSize
X - field in struct :com::sun::star::awt::Point
X - field in struct :com::sun::star::awt::MouseEvent
X - constant in constants group :com::sun::star::awt::Key
X - constant in constants group :com::sun::star::awt::FontStrikeout
X - field in struct :com::sun::star::geometry::IntegerPoint2D
X - field in struct :com::sun::star::awt::Rectangle
X - field in struct :com::sun::star::awt::WindowEvent
XAbortChannel - interface :com::sun::star::task::XAbortChannel
XAbstractView - interface :com::sun::star::xml::dom::views::XAbstractView
XAcceleratorConfiguration - interface :com::sun::star::util::XAcceleratorConfiguration
XAcceptor - interface :com::sun::star::connection::XAcceptor
XAccessControlContext - interface :com::sun::star::security::XAccessControlContext
XAccessController - interface :com::sun::star::security::XAccessController
XAccessible - interface :com::sun::star::accessibility::XAccessible
XAccessibleAction - interface :com::sun::star::accessibility::XAccessibleAction
XAccessibleComponent - interface :com::sun::star::accessibility::XAccessibleComponent



- Codesnippets
 - <https://web.archive.org/web/20130530183917/http://codesnippets.services.openoffice.org/index.xml> (2022-12-11)
 - Scripts in Basic, Java, ooRexx, Python
- ooRexx' "[UNO_API_info.rxo](#)"
 - Installed with BSF4ooRexx
 - Uses reflection and generates writer/pdf documents containing the documentation , linked to the official AOO API reference documentation!
 - Can be invoked via the dispatch interface from any programming language
 - Cf.: https://wi.wu.ac.at/rgf/rexx/misc/OOoCon/2010_Budapest/ (2022-12-11)



- WU Vienna
 - <http://wi.wu.ac.at/rgf/diplomarbeiten/> (2022-12-11)
 - Select [AOO](#), [OOo](#), [LibreOffice](#) and/or [UNO](#) in the keyword dropdown list
 - BSF4ooRexx samples
 - Mostly based on student's work
 - Thesis describe the frameworks and document the samples
 - Some samples installed with BSF4ooRexx in the subdirectory
bsf4oorexx/samples/OOo (BSF4ooRexx 6.41: Java 6+, ooRexx 4.1+) or
bsf4oorexx850/samples/OOo (BSF4ooRexx 8.50: Java 8+, ooRexx 5.0+)



- MRI extension
 - <http://extensions.services.openoffice.org/project/MRI> (2022-12-11)
 - Great AOO inspector written in Python
 - Code (snippet) support for Basic, Java, C++, C# CLI, Python
- AOO mailing lists
 - Consult: http://www.openoffice.org/mail_list.html (2022-12-11)
 - ooo-dev@openoffice.apache.org
 - ooo-api@openoffice.apache.org



- Results of analyzing the AOO Java archives
 - Types and Interfaces (AOO 3.4.1, summer 2012)

jar	Total Types	Interfaces	Share %
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%)
Sum	3 308	1 651	(49.9%)

Querying an Interface

- `queryInterface()` examples
 - `sDispatchHelper`, a service of type `com.sun.star.frame.DispatchHelper`

- `queryInterface()` in Java

```
import com.sun.star.frame.XDispatchHelper;
// ...
XDispatchHelper xDispatchHelper=(XDispatchHelper)
    UnoRuntime.queryInterface(XDispatchHelper.class, sDispatchHelper);
```

- `queryInterface()` in JavaScript

```
importClass(Packages.com.sun.star.frame.XDispatchHelper);
// ...
xDispatchHelper = UnoRuntime.queryInterface(XDispatchHelper, sDispatchHelper);
```

- `queryInterface()` in ooRexx

```
xDispatchHelper=sDispatchHelper~com.sun.star.frame.XDispatchHelper
-- or simpler:
xDispatchHelper=sDispatchHelper~XDispatchHelper
```

- Two kinds of scripting (programming)
 - **Stand-alone**
 - Need to bootstrap OpenOffice in order to initialize the AOO environment to interact with
 - Full control about addressing different AOO servers, if needed
 - Dispatched by AOO (“macro”)
 - AOO supplies a script context that allows access to the initialized AOO environment ([getDesktop](#), [getComponentContext](#)) and to the document ([getDocument](#)) for which the dispatch occurred

Java

```
// import ...
XComponentContext xLocalContext =
com.sun.star.comp.helper.Bootstrap.createInitialComponentContext(null);
// initial serviceManager
XMultiComponentFactory xLocalServiceManager = xLocalContext.getServiceManager();
// create a URL resolver
Object urlResolver = xLocalServiceManager.createInstanceWithContext(
"com.sun.star.bridge.UnoUrlResolver", xLocalContext);
// query for the XUnoUrlResolver interface
XUnoUrlResolver xUrlResolver = (XUnoUrlResolver)
UnoRuntime.queryInterface(XUnoUrlResolver.class, urlResolver);
// Import the object
Object rInitialObject = xUrlResolver.resolve(
"uno:socket,host=localhost,port=8100;urp;StarOffice.ServiceManager");
// test whether we got a reference to the remote ServiceManager
if (null != rInitialObject) {
    System.out.println("initial object successfully retrieved");
} else {
    System.out.println("given initial-object name unknown at server side");
}

... cut ...
```

ooRexx

```
url="uno:socket,host=localhost,port=8100;urp;StarOffice.ServiceManager"
rInitialObject=uno.connect(url)

if rInitialObject<>.nil then
    say "initial object successfully retrieved"
else
    say "given initial-object name unknown at server side"
-- ... cut ...

::requires UNO.CLS    -- get UNO support
```

Creating/Loading Documents



"file:///c:/docs/aFile.odt"
 "http://www.RexxLA.org/aFile.ods"

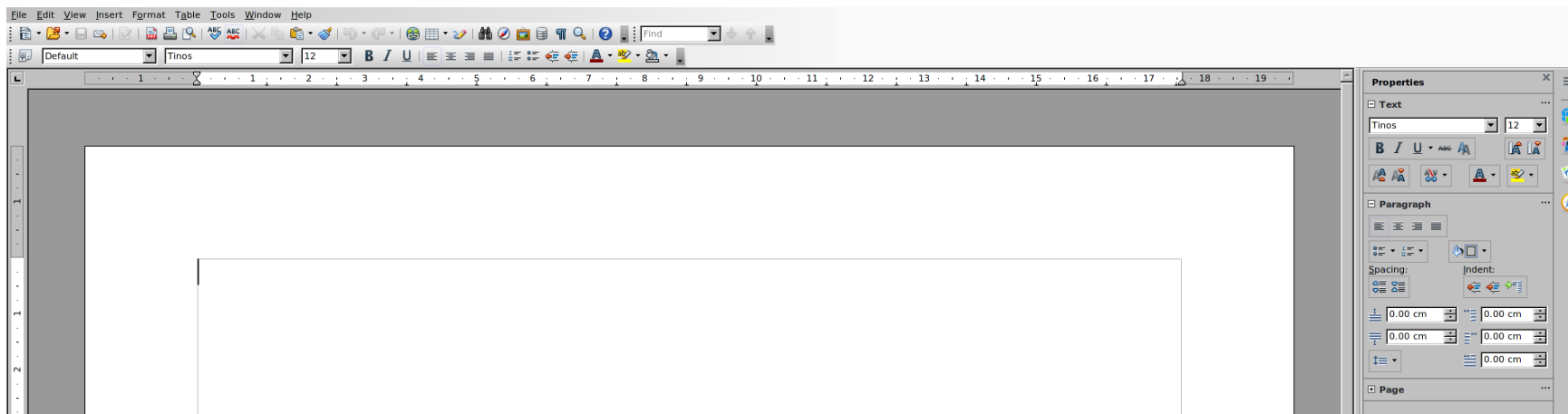
```
xDesktop=uno.createDesktop()  -- bootstrap & get access to
                              XDesktop
xcl=xDesktop~XComponentLoader -- get XComponentLoader interface

uri="private:factory/swriter" -- new swriter document
doc=xcl~loadComponentFromURL(uri,"_blank",0,.uno~noProps)

-- ... now do whatever you want or need to do ...

::requires UNO.CLS  -- get UNO support
```

scal
 swriter
 simpress
 sdraw



Word Processor (“swriter”), 1



- 3 Services

GenericTextDocument (com.sun.star.text.GenericTextDocument),
OfficeDocument (com.sun.star.document.OfficeDocument),
TextDocument (com.sun.star.text.TextDocument)

- 35 Interfaces (unqualified)

XBookmarksSupplier, XChapterNumberingSupplier, XDocumentEventBroadcaster,
XDocumentIndexesSupplier, XDocumentInfoSupplier, XDocumentPropertiesSupplier,
XEmbeddedScripts, XEndnotesSupplier, XEventBroadcaster, XEventsSupplier,
XFootnotesSupplier, XLineNumberingSupplier, XModel, XModifiable, XMultiServiceFactory,
XNumberFormatsSupplier, XPagePrintable, XPrintJobBroadcaster, XPrintable,
XPropertySet, XReferenceMarksSupplier, XRefreshable, XReplaceable, XSearchable,
XStorable, XStyleFamiliesSupplier, **XTextDocument**, XTextEmbeddedObjectsSupplier,
XTextFieldsSupplier, XTextFramesSupplier, XTextGraphicObjectsSupplier,
XTextSectionsSupplier, XTextTablesSupplier, XUndoManagerSupplier, XViewDataSupplier

Word Processor (“swriter”), 2



- 37 Properties
 - ApplyFormDesignMode, ApplyWorkaroundForB6375613, AutomaticControlFocus, BasicLibraries, BuildId, CharFontCharSet, CharFontCharSetAsian, CharFontCharSetComplex, CharFontFamily, CharFontFamilyAsian, CharFontFamilyComplex, CharFontName, CharFontNameAsian, CharFontNameComplex, CharFontPitch, CharFontPitchAsian, CharFontPitchComplex, CharFontStyleName, CharFontStyleNameAsian, CharFontStyleNameComplex, CharLocale, **CharacterCount**, DialogLibraries, ForbiddenCharacters, HasValidSignatures, HideFieldTips, IndexAutoMarkFileURL, LockUpdates, ParagraphCount, RecordChanges, RedlineDisplayType, RedlineProtectionKey, RuntimeUID, ShowChanges, TwoDigitYear, WordCount, WordSeparator

Word Processor (“swriter”), 3



- Interface `com.sun.star.text.XTextDocument`
 - Get access to the text object representing the text of the entire document using `getText()`
 - Returns `XText`, which is derived from `XSimpleText`, which is derived from `XRangeText`, hence the methods of all three interfaces are available!
- Concept of “cursors”, e.g.
 - Pages, Paragraphs, Sentences, Words, Characters
- Possible to also insert tables, fields, pictures, drawings, ...

Create Word Processor Document (“swriter”), 1

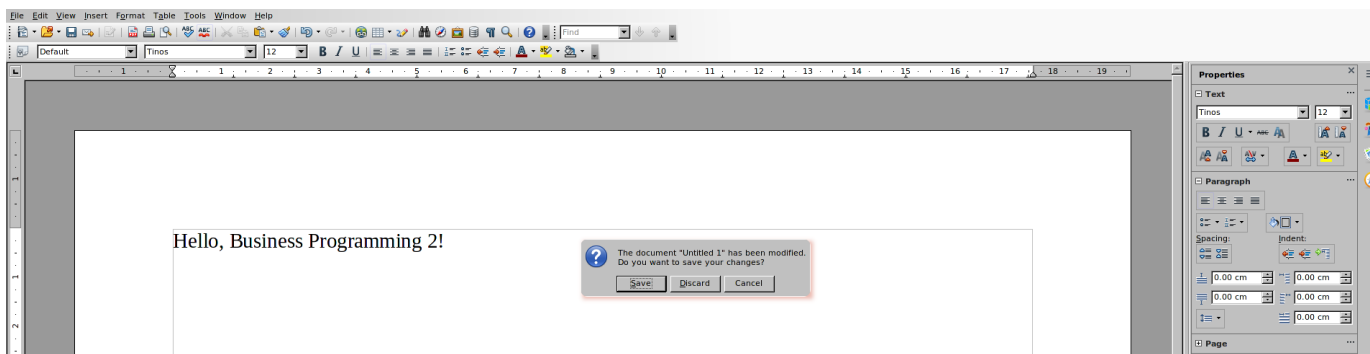
- Add text “[Hello, Business Programming 2!](#)”
- Closing the word processor document manually will cause the “Save”-dialog to appear

```
xDesktop=uno.createDesktop()      -- bootstrap & get access to XDesktop
xcl=xDesktop~XComponentLoader    -- get XComponentLoader interface

uri="private:factory/swriter"     -- new swriter document
doc=xcl~loadComponentFromURL(uri,"_blank",0,.uno~noProps)

xText=doc~XTextDocument~getText  -- get text object
xText~setString("Hello, Business Programming 2!")

::requires UNO.CLS                -- get UNO support
```



Create Word Processor Document (“swriter”), 2

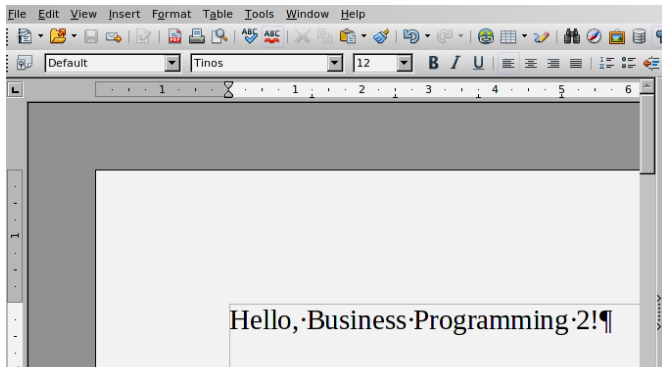
```
xDesktop=uno.createDesktop()      -- bootstrap & get access to
XDesktop
xcl=xDesktop~XComponentLoader     -- get XComponentLoader interface

uri="private:factory/swriter"     -- new swriter document
doc=xcl~loadComponentFromURL(uri,"_blank",0,.uno~noProps)

xText=doc~XTextDocument~getText  -- get text object
xText~setString("Hello, Business Programming 2!")

doc~XModifiable~setModified(.false) -- set document to unmodified
call SysSleep 5                  -- sleep 5 seconds
doc~XCloseable~close(.false)     -- close document (window)

::requires UNO.CLS                -- get UNO support
```



- Change state of document to “unmodified”
 - Leftover document can be closed without a save dialog
 - Using interface [com.sun.star.util.XModifiable](#)
- Sleep five seconds, then close document
 - Using interface [com.sun.star.util.XCloseable](#)

Create Word Processor Document (“swriter”), 3

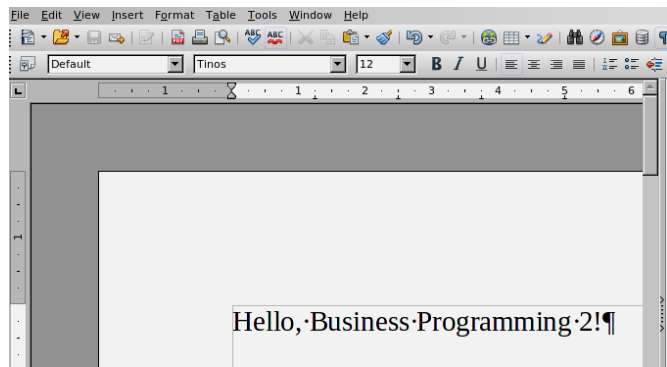
```
xDesktop=uno.createDesktop()      -- bootstrap & get access to XDesktop
xcl=xDesktop~XComponentLoader     -- get XComponentLoader interface
uri="private:factory/swriter"     -- new swriter document
doc=xcl~loadComponentFromURL(uri,"_blank",0,.uno~noProps)
```

```
xText=doc~XTextDocument~getText  -- get text object
xText~setString("Hello, Business Programming 2!")
```

```
xprops=doc~XPropertySet          -- get access to the properties
say "character count:" xprops~getPropertyValue("CharacterCount")
```

```
doc~XModifiable~setModified(.false) -- set document to unmodified
call SysSleep 5                     -- sleep 5 seconds
doc~XCloseable~close(.false)       -- close document (window)
```

```
::requires UNO.CLS -- get UNO support
```



- Access and show property [CharacterCount](#)
- Change state of document to “unmodified”
 - Leftover document can be closed without a save dialog
 - Using interface [com.sun.star.util.XModifiable](#)
- Sleep five seconds, then close document
 - Using interface [com.sun.star.util.XCloseable](#)

Output:

```
character count: 30
```

Create Word Processor Document (“swriter”), 4, 1

```
xDesktop=uno.createDesktop()      -- bootstrap & get access to XDesktop
xcl=xDesktop~XComponentLoader     -- get XComponentLoader interface

uri="private:factory/swriter"      -- new swriter document
doc=xcl~loadComponentFromURL(uri,"_blank",0,.uno~noProps)

xText=doc~XTextDocument~getText   -- get text object
xText~setString("Hello, Business Programming 2!")

-- change second word
xTextCursor=xText~createTextCursor -- character based cursor
xTextCursor~gotoStart(.false)      -- make sure we are at start

xWordCursor=xTextCursor~XWordCursor -- get the XWordCursor interface
xWordCursor~gotoNextWord(.false)    -- XTextRange represents first word
xWordCursor~gotoNextWord(.true)     -- select second word, includes blank!
xWordCursor~setString("ooRexx with BSF4ooRexx ") -- note trailing blank

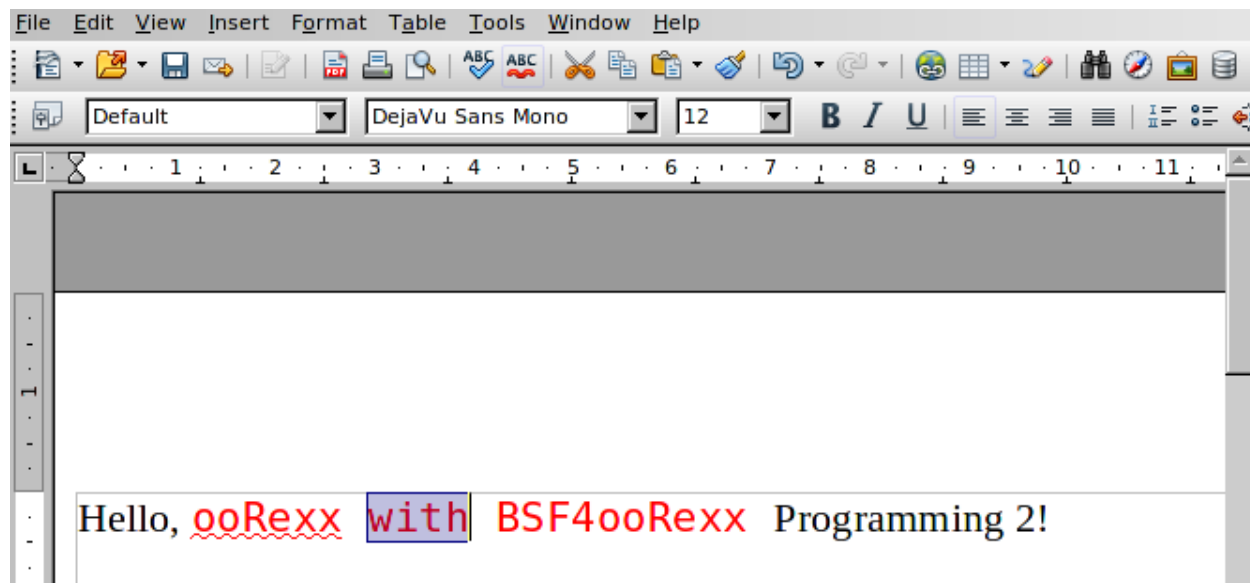
-- change color
red=box("int", "FF 00 00"x ~c2d)    -- color red (RGB color) as integer
xWordCursor~XPropertySet~setProperty("CharColor", red)

-- change font
fontName="DejaVu Sans Mono"
xWordCursor~XPropertySet~setProperty("CharFontName", fontName)
say ppd(xWordCursor~uno.getDefinition)

::requires UNO.CLS -- get UNO support
```

- Replace “Business” with “ooRexx with BSF4ooRexx”
- Change the color to red
- Change the font name to “DejaVu Sans Mono”
- Show textual definition of UNO object referred to with variable “xWordCursor”

Create Word Processor Document (“swriter”), 4, 2



Output:

```
UNO_SERVICE|
com.sun.star.text.TextCursor+com.sun.star.style.CharacterProperties+com.sun.star.style.CharacterPropertiesAsian+com.sun.star.st
yle.CharacterPropertiesComplex+com.sun.star.style.ParagraphProperties+com.sun.star.style.ParagraphPropertiesAsian+com.sun.star.
style.ParagraphPropertiesComplex+com.sun.star.text.TextSortTable|SwXTextCursor
... cut ...
```

Create Word Processor Document (“swriter”), 5, 1

```
xDesktop=uno.createDesktop()      -- bootstrap & get access to XDesktop
xcl=xDesktop~XComponentLoader     -- get XComponentLoader interface

uri="private:factory/swriter"      -- new swriter document
doc=xcl~loadComponentFromURL(uri,"_blank",0,.uno~noProps)

xText=doc~XTextDocument~getText   -- get text object
xText~setString("Hello, Business Programming 2!")

xTextCursor=xText~createTextCursor -- create the character based cursor
-- make paragraph's properties accessible:
xParaProps=xTextCursor~XParagraphCursor~XPropertySet

ctlChars=.uno_constants~new("com.sun.star.text.ControlCharacter") -- UNO_CONSTANT
paraBreak=ctlChars~paragraph_break -- get paragraph break constant

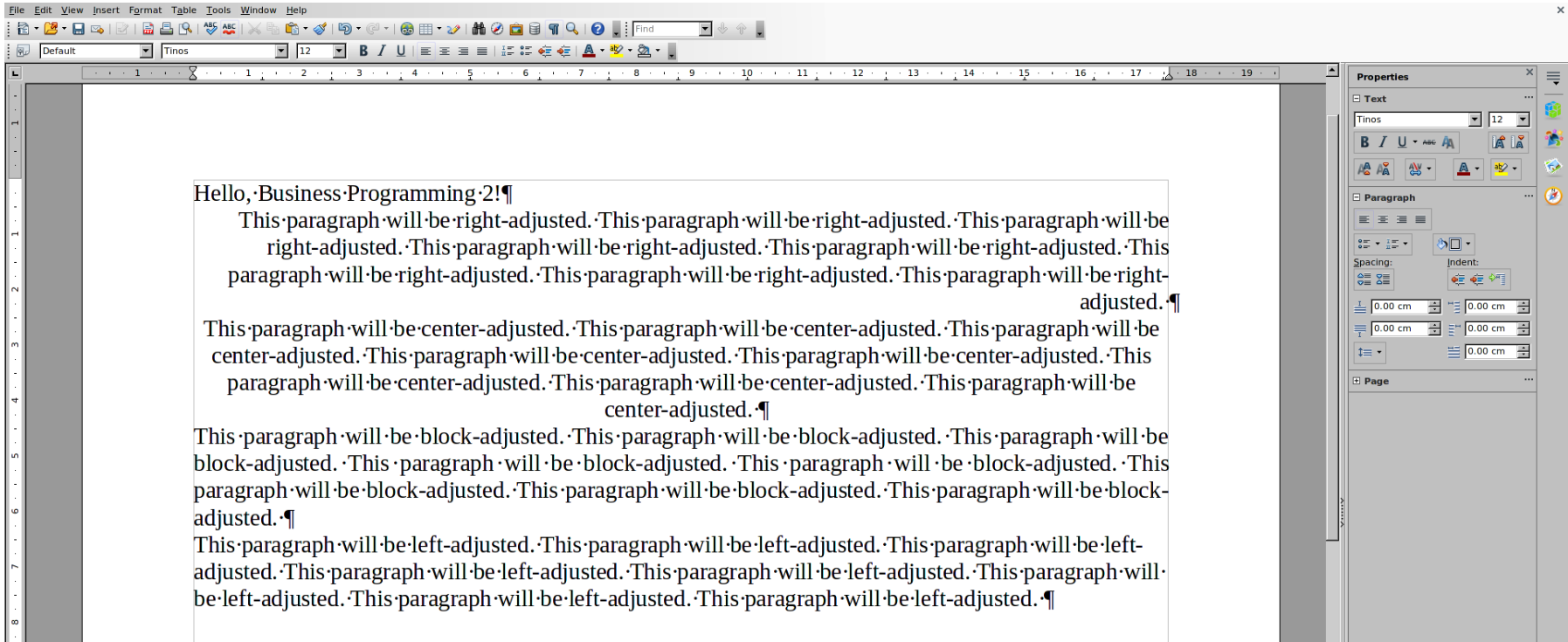
paraAdj =.uno_enum~new("com.sun.star.style.ParagraphAdjust") -- UNO_ENUM

arr=.array~of("right", "center", "block", "left") -- adjustments
do adj over arr -- iterate over adjustments, create string, adjust
  xTextCursor~gotoEnd(.false) -- position at end
  xText~insertControlCharacter(xTextCursor, paraBreak, .false)
  string=("This paragraph will be" adj"-adjusted. ")~copies(8)
  xText~insertString(xTextCursor, string, .true)
  xParaProps~setProperty("ParaAdjust", paraAdj~send(adj))
end

::requires UNO.CLS -- get UNO support
```

- Demonstrate creating and styling paragraphs
 - Get access to the paragraph properties
 - Access [com.sun.star.text.ControlCharacter](#) constants
 - Access to [com.sun.star.style.ParagraphAdjust](#) enums
 - Demonstrate adjusting paragraphs to “right”, “center”, “block”, “left” using a string that contains the adjustment verb

Create Word Processor Document (“swriter”), 5, 2



Spreadsheet (“scal”), 1



- 3 Services
 - OfficeDocument (com.sun.star.document.OfficeDocument),
SpreadsheetDocument (com.sun.star.sheet.SpreadsheetDocument),
SpreadsheetDocumentSettings (com.sun.star.sheet.SpreadsheetDocumentSettings)
- 26 Interfaces (unqualified)
 - XActionLockable, XCalculatable, XConsolidatable, XDocumentAuditing,
XDocumentEventBroadcaster, XDocumentInfoSupplier, XDocumentPropertiesSupplier,
XDrawPagesSupplier, XEmbeddedScripts, XEventBroadcaster, XEventsSupplier,
XGoalSeek, XLinkTargetSupplier, XModel, XModifiable, XMultiServiceFactory,
XNumberFormatsSupplier, XPrintJobBroadcaster, XPrintable, XPropertySet, XProtectable,
[XSpreadsheetDocument](#), XStorable, XStyleFamiliesSupplier, XUndoManagerSupplier,
XViewDataSupplier

Spreadsheet (“scal”), 2



- 40 Properties
 - ApplyFormDesignMode, AreaLinks, AutomaticControlFocus, BasicLibraries, BuildId, CalcAsShown, CharLocale, CharLocaleAsian, CharLocaleComplex, CodeName, ColumnLabelRanges, DDELinks, DatabaseRanges, DefaultTabStop, DialogLibraries, ExternalDocLinks, ForbiddenCharacters, HasDrawPages, HasValidSignatures, IgnoreCase, IsAdjustHeightEnabled, IsChangeReadOnlyEnabled, IsExecuteLinkEnabled, IsIterationEnabled, IsLoaded, IsUndoEnabled, IterationCount, IterationEpsilon, LookUpLabels, MatchWholeCell, NamedRanges, NullDate, ReferenceDevice, RegularExpressions, RowLabelRanges, RuntimeUID, SheetLinks, SpellOnline, StandardDecimals, VBAGlobalConstantNamer

Spreadsheet (“scal”), 3



- Interface `com.sun.star.sheet.XSpreadsheetDocument`
 - Get name access to the collection of `XSpreadsheets`
 - Numeric (0-based) access with `XIndexAccess`
- Concept of “table” consisting of a collection of rows, which each have columns
 - `XCellRange` (a tabular area of a spreadsheet)
 - Origin “0,0” represents upper left-hand corner
 - Offsets relative to upper left-hand corner

Spreadsheet (“scal”), 4



- Addressing a cell
 - Numerically (0-based) representing offsets from origin
 - e.g. “0,1” (first column, second row)
 - `getCellByPosition(columnOffset,rowOffset)` returns a `XCell`
 - By name
 - a named range, or
 - column: a name, row: a 1-based number), e.g. “A2”
 - `getCellRangeByName(Name)` returns a `XCellRange`, then
 - `getCellByPosition(0,0)` returns a `XCell`
 - Possible to also insert charts, drawings, ...

Create Spreadsheet Document (“scal”), 1

- Add text “Hello, Business Programming 2!”
- Demonstrate how to store a document

```
xDesktop=uno.createDesktop()           -- bootstrap & get access to XDesktop
xcl=xDesktop~XComponentLoader          -- get XComponentLoader interface

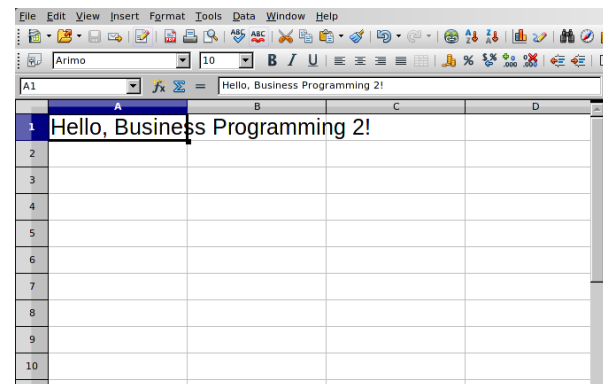
uri="private:factory/scalc"            -- new scalc document
doc=xcl~loadComponentFromURL(uri,"_blank",0,.uno~noProps)

xSheets=doc~XSpreadSheetDocument~getSheets~XIndexAccess
xSheet =xSheets~getByIndex(0)~XSpreadSheet -- get first spreadsheet
-- add entry to "A1"
xSheet~getCellByPosition(0,0)~setFormula("Hello, Business Programming 2!")

storeURL=directory()"/scalcl.ods"      -- save document in local directory
storeURL=uno.convertToUrl(storeURL)    -- change path to URL-style
doc~XStorable~storeAsURL(storeURL,.uno~noProps) -- save document

doc~XClosable~close(.false)           -- close document (window)

::requires UNO.CLS                     -- get UNO support
```



Create Spreadsheet Document (“scal”), 2

- Demonstrate how to change the height of table rows

```
xDesktop=uno.createDesktop()      -- bootstrap & get access to XDesktop
xcl=xDesktop~XComponentLoader     -- get XComponentLoader interface

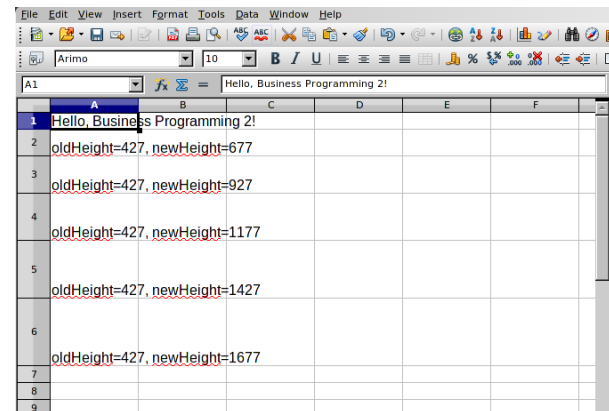
uri="private:factory/scalc"       -- new scalc document
doc=xcl~loadComponentFromURL(uri,"_blank",0,.uno~noProps)

xSheets=doc~XSpreadSheetDocument~getSheets~XIndexAccess
xSheet =xSheets~getByIndex(0)~XSpreadSheet -- get first spreadsheet
-- add entry to "A1"
xSheet~getCellByPosition(0,0)~setFormula("Hello, Business Programming 2!")

xRows=xSheet~XColumnRowRange~getRows-- get XTableRow

do i=1 to 5                      -- 0-based, hence lines # 2 through # 6
  xRow=xRows~getByIndex(i)       -- fetch XRow
  props=xRow~XPropertySet        -- get access to its properties
  oldHeight=props~getPropertyValue("Height") -- get current value
  newHeight=oldHeight+i*250      -- increase by i*0.250 cm
  props~setProperty("Height", box("int",newHeight)) -- set new Height
  text="oldHeight="+oldHeight+", newHeight="+newHeight -- create info text
  xSheet~getCellByPosition(0,i)~setFormula(text) -- set cell to info text
end

::requires UNO.CLS               -- get UNO support
```



	A	B	C	D	E	F
1	Hello, Business Programming 2!					
2	oldHeight=427, newHeight=677					
3	oldHeight=427, newHeight=927					
4	oldHeight=427, newHeight=1177					
5	oldHeight=427, newHeight=1427					
6	oldHeight=427, newHeight=1677					
7						
8						
9						

Create Spreadsheet Document (“scal”), 3

- Demonstrate how to change the width of table columns

```
xDesktop=uno.createDesktop()      -- bootstrap & get access to XDesktop
xcl=xDesktop~XComponentLoader     -- get XComponentLoader interface

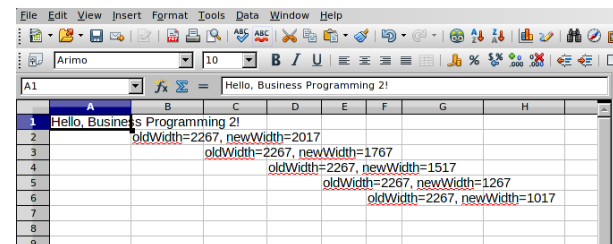
uri="private:factory/scalc"       -- new scalc document
doc=xcl~loadComponentFromURL(uri,"_blank",0,.uno~noProps)

xSheets=doc~XSpreadSheetDocument~getSheets~XIndexAccess
xSheet =xSheets~getByIndex(0)~XSpreadSheet -- get first spreadsheet
-- add entry to "A1"
xSheet~getCellByPosition(0,0)~setFormula("Hello, Business Programming 2!")

xCols=xSheet~XColumnRowRange~getColumns-- get XTableColumns

do i=1 to 5                      -- 0-based, hence columns # 2 (B) through # 6 (F)
  xCol=xCols~getByIndex(i)      -- fetch xCol
  props=xCol~XPropertySet       -- get access to its properties
  oldWidth=props~getPropertyValue("Width") -- get current value
  newWidth=oldWidth-i*250       -- decrease by i*0.250 cm
  props~setPropertyValue("Width", box("int",newWidth)) -- set new Width
  text="oldWidth="oldWidth", newWidth="newWidth -- create info text
  xSheet~getCellByPosition(i,i)~setFormula(text) -- set cell to info text
end

::requires UNO.CLS              -- get UNO support
```



Create Spreadsheet Document (“scalc”), 4

```
xDesktop=uno.createDesktop()      -- bootstrap & get access to XDesktop
xcl=xDesktop~XComponentLoader     -- get XComponentLoader interface

uri="private:factory/scalc"       -- new scalc document
doc=xcl~loadComponentFromURL(uri,"_blank",0,.uno~noProps)

xSheets=doc~XSpreadSheetDocument~getSheets~XIndexAccess
xSheet =xSheets~getByIndex(0)~XSpreadSheet -- get first spreadsheet
```

```
call uno.setCell xSheet, 0, 0, "Name:"      -- cell "A1"
call uno.setCell xSheet, "B1", "John Doe"   -- cell "B1"
call uno.setCell xSheet, "A2", "Date:"      -- cell "A2"
call uno.setCell xSheet, 1, 1, "=&TODAY()"   -- cell "B2"
-- format individual cells
```

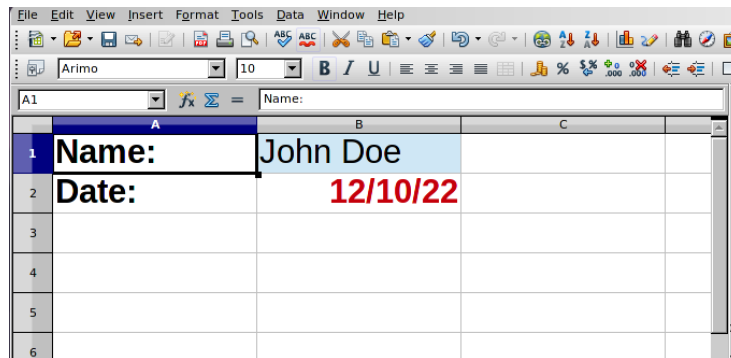
```
xCellB1=xSheet~getCellByPosition(1, 0)     -- get access to cell "B1"
cbc=box("int", "CF E7 F5"x ~c2d)            -- define a RGB color
xCellB1~XPropertySet~setProperty("CellBackColor", cbc) -- set color
```

```
xCellB2=xSheet~getCellByPosition(1, 1)     -- get access to cell "B2"
cc=box("int", "c5 00 0b"x ~c2d)            -- define a RGB color
props=xCellB2~XPropertySet
props~setProperty("CharColor", cc)         -- set color
fontWeight=.uno_constants~new("com.sun.star.awt.FontWeight")
props~setProperty("CharWeight", fontWeight~semiBold)
```

```
-- format using the properties of a XCellRange for "A1:A2"
props=xSheet~XCellRange~getCellRangeByName("A1:A2")~XPropertySet
props~setProperty("CharWeight", fontWeight~bold)
```

```
::requires UNO.CLS -- get UNO support
```

- Add text and a date
- Demonstrate how to format individual cells and a cell range



Create Spreadsheet Document (“scal”), 5, 1

```

xDesktop=uno.createDesktop()      -- bootstrap & get access to XDesktop
xcl=xDesktop-XComponentLoader     -- get XComponentLoader interface

uri="private:factory/scalc"       -- new scalc document
doc=xcl-loadComponentFromURL(uri,"_blank",0,.uno-noProps)

xSheets=doc-XSpreadSheetDocument-getSheets-XIndexAccess
xSheet =xSheets-getByIndex(0)-XSpreadSheet -- get first spreadsheet

call uno.setCell xSheet, "A1", "Quarter"
call uno.setCell xSheet, "B1", "2011"
call uno.setCell xSheet, "C1", "2012"
do i=1 to 4
  call uno.setCell xSheet, 0, i, "Q"i
  call uno.setCell xSheet, 1, i, random(0,5000)
  call uno.setCell xSheet, 2, i, random(0,5000)
end

props=xSheet-XCellRange-getCellRangeByName("A1:C1")-XPropertySet -- column headings
fontWeight=.uno_constants-new("com.sun.star.awt.FontWeight")
props-setPropertyValue("CharWeight", fontWeight-bold)

props=xSheet-XCellRange-getCellRangeByName("B2:C5")-XPropertySet -- format numbers
props-setPropertyValue("NumberFormat", 4) -- predefined style, format: "#,##0.00"

structRect = .bsf-new("com.sun.star.awt.Rectangle") -- position & size of chart
structRect-X      = 300      -- x-offset: 0.300 cm
structRect-Y      = 2250     -- y-offset: 2.250 cm
structRect-Width  = 16000    -- width: 16.000 cm
structRect-Height = 8000     -- height: 8.000 cm

xRange=xSheet-XCellRange-getCellRangeByName("A1:C5") -- data to be used for chart
rangeAddr = xRange-XCellRangeAddressable-getRangeAddress
arrAddr=bsf.createArrayOf(rangeAddr-getClass, rangeAddr) -- create array

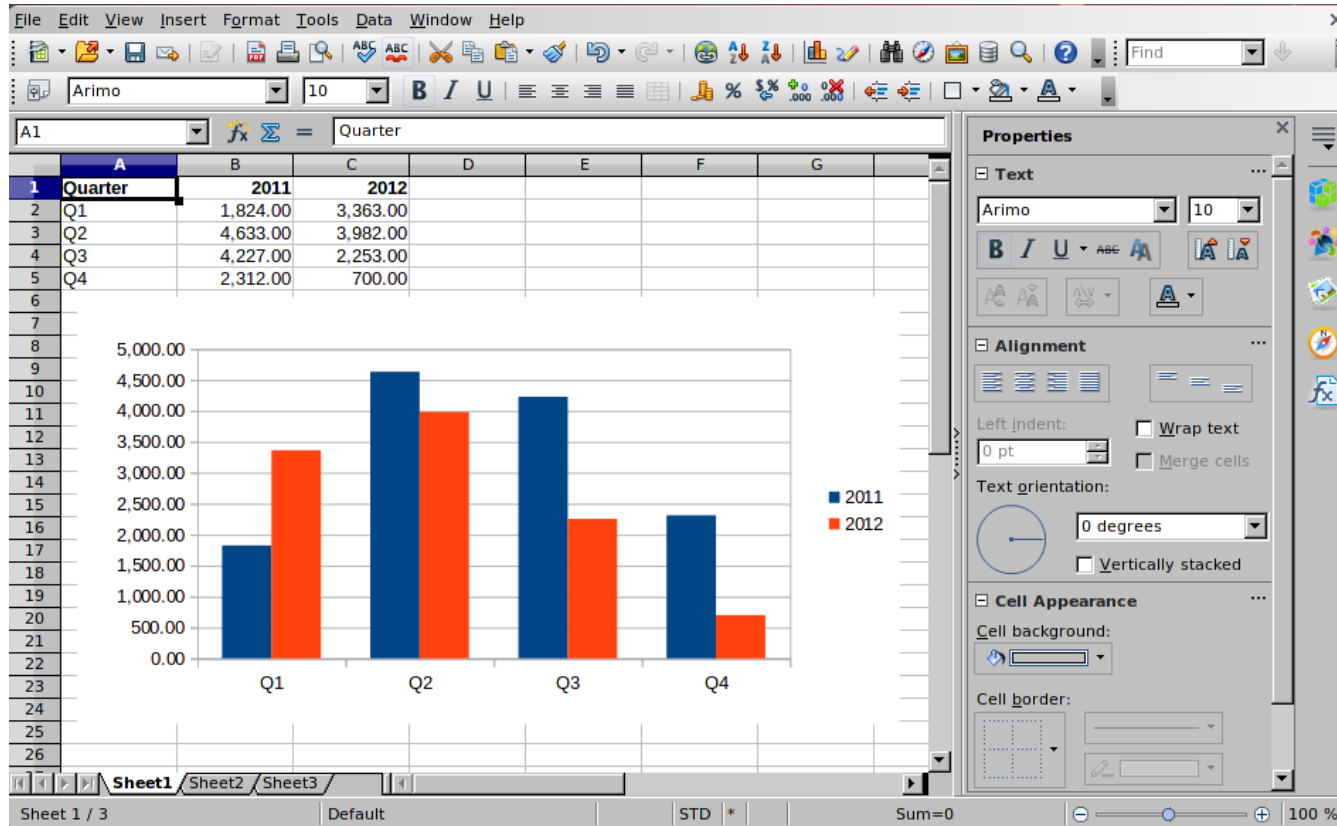
xTableCharts = xSheet-XTableChartsSupplier-getCharts -- get Chart collection & insert
xTableCharts-addNewByName("FirstChart", structRect, arrAddr, .true, .true)

::requires UNO.CLS -- get UNO support

```

- Generate data for four quarters for 2011 and 2012
- Format column headings
- Format numbers
- Create a chart from the generated data

Create Spreadsheet Document (“scal”), 5, 2



Drawing (“sdraw”), 1



- 4 Services
 - DrawingDocument (com.sun.star.drawing.DrawingDocument),
DrawingDocumentFactory (com.sun.star.drawing.DrawingDocumentFactory),
GenericDrawingDocument (com.sun.star.drawing.GenericDrawingDocument),
OfficeDocument (com.sun.star.document.OfficeDocument)
- 20 Interfaces (unqualified)
 - XDocumentEventBroadcaster, XDocumentInfoSupplier, XDocumentPropertiesSupplier,
XDrawPageDuplicator, XDrawPagesSupplier, XEmbeddedScripts, XEventBroadcaster,
XEventsSupplier, XLayerSupplier, XMasterPagesSupplier, XModel, XModifiable,
XMultiServiceFactory, XPrintJobBroadcaster, XPrintable, XPropertySet, XStorable,
XStyleFamiliesSupplier, XUndoManagerSupplier, XViewDataSupplier

Drawing (“sdraw”), 2



- 12 Properties
 - ApplyFormDesignMode, AutomaticControlFocus, BasicLibraries, BuildId, CharLocale, DialogLibraries, ForbiddenCharacters, HasValidSignatures, MapUnit, RuntimeUID, TabStop, VisibleArea

Drawing (“sdraw”), 3



- A collection of draw pages
- Each draw page
 - Allows any kind of drawing
 - Allows animation effects to be applied
- The draw concepts are fully reused for presentation documents!

Create Drawing Document (“scal”), 1

```
xDesktop=uno.createDesktop()      -- bootstrap & get access to XDesktop
xcl=xDesktop~XComponentLoader     -- get XComponentLoader interface

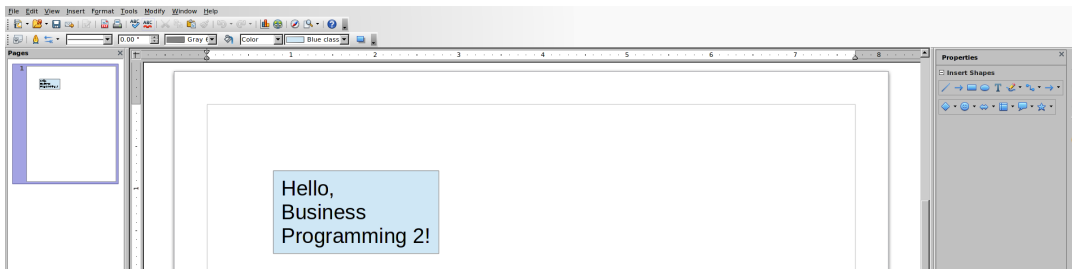
uri="private:factory/sdraw"       -- new sdraw document
doc=xcl~loadComponentFromURL(uri, "_blank", 0, .uno~noProps)

xsf=doc~XMultiServiceFactory     -- get the service manager (factory)
-- get access to the first draw page
xDrawPage = doc~XDrawPagesSupplier~getDrawPages~getByIndex(0)~XDrawPage

-- create a Rectangle shape and determine its position and size
xShape=xsf~createInstance("com.sun.star.drawing.RectangleShape") ~XShape
xShape~setPosition(.bsf~new("com.sun.star.awt.Point", 3000, 3000))
xShape~setSize(.bsf~new("com.sun.star.awt.Size", 5000, 2500))

xDrawPage~add(xShape)            -- add new shape to first draw page
cr="0d"x                         -- ASCII carriage return char
xShape~XText~setString("Hello,"cr"Business"cr"Programming 2!") -- now set string

::requires UNO.CLS               -- get UNO support
```



- Fetch the drawing component's service manager
 - Used to create shapes that can be stored with the document
- Create and draw a rectangular shape, add it to the document
 - Set the shape's text to “Hello, Business Programming 2!”
 - Break up the text such that it fits into the rectangle

Presentation (“simplpress”), 1



- 4 Services
 - DrawingDocumentFactory (com.sun.star.drawing.DrawingDocumentFactory),
GenericDrawingDocument (com.sun.star.drawing.GenericDrawingDocument),
OfficeDocument (com.sun.star.document.OfficeDocument),
PresentationDocument (com.sun.star.presentation.PresentationDocument)
- 23 Interfaces (unqualified)
 - XCustomPresentationSupplier, XDocumentEventBroadcaster, XDocumentInfoSupplier,
XDocumentPropertiesSupplier, XDrawPageDuplicator, XDrawPagesSupplier,
XEmbeddedScripts, XEventBroadcaster, XEventsSupplier, XLayerSupplier,
XLinkTargetSupplier, XMasterPagesSupplier, XModel, XModifiable, XMultiServiceFactory,
XPresentationSupplier, XPrintJobBroadcaster, XPrintable, XPropertySet, XStorable,
XStyleFamiliesSupplier, XUndoManagerSupplier, XViewDataSupplier

Presentation (“simpres”), 2



- 12 Properties
 - ApplyFormDesignMode, AutomaticControlFocus, BasicLibraries, BuildId, CharLocale, DialogLibraries, ForbiddenCharacters, HasValidSignatures, MapUnit, RuntimeUID, TabStop, VisibleArea

Presentation (“simpres”), 3



- A collection of draw pages
- Each draw page
 - Allows any kind of drawing
 - Allows animation effects to be applied
- Concept of “Master Pages”
 - Allows definition of specific layouts
- Layouts for title, listings, charts, etc.
- Presentation mode

Create Presentation Document (“simpres”), 1

```
xDesktop=uno.createDesktop()           -- bootstrap & get access to XDesktop
xcl=xDesktop~XComponentLoader           -- get XComponentLoader interface

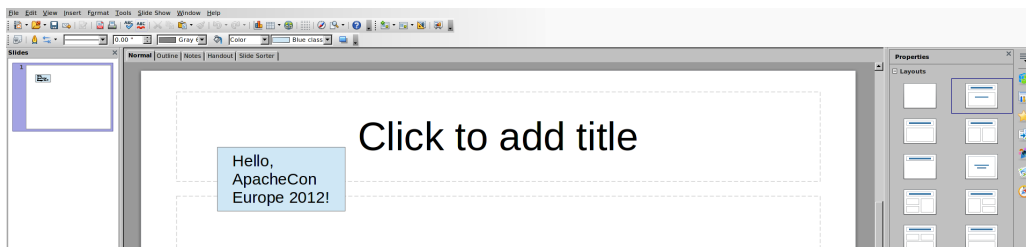
uri="private:factory/simpres"           -- new simpres document
doc=xcl~loadComponentFromURL(uri,"_blank",0,.uno~noProps)

xsf=doc~XMultiServiceFactory           -- get the service manager (factory)
-- get access to the first draw page
xDrawPage = doc~XDrawPagesSupplier~getDrawPages~getByIndex(0)~XDrawPage

-- create a Rectangle shape and determine its position and size
xShape=xsf~createInstance("com.sun.star.drawing.RectangleShape") ~XShape
xShape~setPosition(.bsf~new("com.sun.star.awt.Point", 3000, 3000))
xShape~setSize(.bsf~new("com.sun.star.awt.Size", 5000, 2500))

xDrawPage~add(xShape)                  -- add new shape to first draw page
cr="0d"x                               -- ASCII carriage return char
xShape~XText~setString("Hello,"cr"ApacheCon"cr"Europe 2012!") -- now set string

::requires UNO.CLS                      -- get UNO support
```



- Fetch its component's service manager
 - Used to create shapes that can be stored with the document
- Create and draw a rectangular shape, add it to the document
 - Set the shape's text to **“ApacheCon Europe 2012!”**
 - Break up the text such that it fits into the rectangle
- Except for the URL, the same code as for “sdraw”!

Create Presentation Document (“simplpress”), 2, 1

```
xDesktop=uno.createDesktop()      -- bootstrap & get access to XDesktop
xcl=xDesktop~XComponentLoader      -- get XComponentLoader interface

uri="private:factory/simplpress"    -- new simplpress document
doc=xcl~loadComponentFromURL(uri, "_blank", 0, .uno~noProps)

xDrawPages = doc~XDrawPagesSupplier~getDrawPages    -- get DrawPages

xDrawPage=xDrawPages~getByIndex(0)  -- get first (empty) page
xDrawPage~XPropertySet~setProperty("Layout", box("short",0)) -- "Title Slide"
xShapes=xDrawPage~XShapes           -- get access to its shapes
xShapes~getByIndex(0)~XText~setString("Business Programming 2!")
xShapes~getByIndex(1)~XText~setString("Scripting Apache OpenOffice")

xDrawPage=xDrawPages~insertNewByIndex(1)~getByIndex(1) -- insert at end, get access
xDrawPage~XPropertySet~setProperty("Layout", box("short",1)) -- "Title Content"
xShapes=xDrawPage~XShapes           -- get access to its shapes
xShapes~getByIndex(0)~XText~setString("Scripting Apache OpenOffice")

lf="0a"x      -- define line-feed character
tab="09"x     -- define tabulator character
str="First" lf"Second" lf tab "Second, 1" lf tab "Second, 2" lf"Third"
xShapes~getByIndex(1)~XText~setString(str)

doc~XPresentationSupplier~getPresentation~bsf.dispatch("start") -- start presentation

::requires UNO.CLS      -- get UNO support
```

- Create two pages with different layouts
 - One “Title Slide” page, layout number: 0
 - One “Title, Content” page, layout number: 1
- Start the presentation at the end

Create Presentation Document (“simpres”), 2, 2



The image displays two screenshots of the OpenOffice Impress application interface. The top screenshot shows a slide titled "Business Programming 2!". The bottom screenshot shows a slide titled "Scripting Apache OpenOffice" with a list of items: "First", "Second", "Second, 1", "Second, 2", and "Third". A blue callout bubble with a dashed border points to the second slide and contains the text "Problem: no bullets!". The application windows show standard menus (File, Edit, View, Insert, Format, Tools, Slide Show, Window, Help) and a Properties panel on the right.

Create Presentation Document (“simpres”), 3, 1

```

...cut...
xText=xShapes~getByIndex(1)~XText  -- content's XText
call addItem xText, "First",      0 -- add string, determine level
call addItem xText, "Explored by many", 0
call addItem xText, "Kudos! go to",  1
call addItem xText, "Christoph Jopp!", 1
call addItem xText, "On 2012-11-07", 0, .false
...cut...

::routine addItem  -- adds string at the given (0-based outline) level
  use arg xText, string, level, bNewParagraph=.true

  xTR=xText~XTextRange~getEnd  -- get end, a XTextRange
  xTR~XPropertySet~setProperty("NumberingLevel",level) -- set XTextRange level

  xTR~setString(string)        -- set string

  if bNewParagraph=.true then  -- add new paragraph
    xTR~getEnd~setString("0a"x) -- add linefeed character -> new paragraph
    -- xText~insertControlCharacter(xTextCursor,0,.false) -- o.k.

::routine dumpItems  -- show level and string from XText
  use arg xText

  enum=xText~XEnumerationAccess~createEnumeration -- enumerate paragraphs
  do i=1 while enum.hasMoreElements
    xtr=enum.nextElement~XTextRange -- we need XTextRange's string & properties
    nl=xtr~XPropertySet~getPropertyValue("NumberingLevel")
    say "    item #" i: NumberingLevel="pp(nl) .pp(xtr~getString)
  end

```

Routine originally created for debugging,
helpful for analyzing any enumeration in UNO,
hence leaving it on the slide

- Create two pages with different layouts
 - One “Title Slide” page, layout number: 0
 - One “Title, Content” page, layout number: 1
 - Use AOO's impress outline levels!
 - Kudos to Christoph Jopp, who found the property to use!

Create Presentation Document (“simpres”), 3, 2



Business Programming 2!

Scripting Apache OpenOffice

Scripting Apache OpenOffice

- First
- Explored by many
 - Kudos! go to
 - Christoph Jopp!
- On 2012-11-07

Solved (by Christoph Jopp): proper bullets !

URE (UNO Runtime Environment)



- There are UNO types that can be used independently of the AOO GUI! E.g.
 - `"com.sun.star.lang.Locale"`
 - `"com.sun.star.linguistic2.LinguServiceManager"`
- Can be used by/incorporated into any other application!
- Need to bootstrap and connect to the **UNO runtime environment** (URE)
 - Fetch its service manager
 - Instantiate services
 - Use services, request their interfaces



```

xContext = UNO.connect()           -- bootstrap and connect to URE
xSM = xContext~getServiceManager   -- get the service manager

serviceName="com.sun.star.linguistic2.LinguServiceManager"
lsm=xsm~createInstanceWithContext(serviceName, xContext) -- create the service
xSpellChecker = lsm~XLinguServiceManager~getSpellChecker -- get the spell checker
locales=xSpellChecker~XSupportedLocales~getLocales      -- get all supported locales

word="thru"                        -- word to spellcheck
do locale over locales             -- iterate over all available Locales
  str=locale~language/"locale~country"/"locale~variant "-> word:" pp(word):"
  ok=xSpellChecker~isValid(word, locale, .UNO~noProps) -- check word
  if ok then str=str "correct"
  else str=str "NOT correct! Available alternatives:"
  say str

  if \ok then                      -- not correct, get & show alternatives
  do
    alternatives=xSpellChecker~spell(word, locale, .UNO~noProps)
    if alternatives <> .nil then
    do
      do a over alternatives~getAlternatives
        say "0909"x pp(a)
      end
    end
  end
end
end

::requires UNO.CLS                 -- get UNO support

```

- Create a connection to URE
- Get its service manager
 - Used to create the spellchecker service using the service "com.sun.star.linguistic2.LinguServiceManager"
- Use all locales available to the spellchecker
 - In this example: some English locales
- Spellcheck the word “thru” with the different English locales
 - If not correct, list the alternatives of the locale

URE, Spellchecker, 1, 2



Output:

```
en/US/ -> word: [thru]: correct
en/GB/ -> word: [thru]: NOT correct! Available alternatives:
    [thrum]
    [thou]
    [thrush]
    [thrust]
    [truth]
    [Ruth]
en/AU/ -> word: [thru]: correct
en/CA/ -> word: [thru]: correct
en/ZA/ -> word: [thru]: NOT correct! Available alternatives:
    [thrum]
    [thou]
    [thrush]
    [thrust]
    [Thur]
    [truth]
    [through]
    [three]
```