



# Erweiterung der Skriptfähigkeit von OpenOffice.org durch BSF und JSR-223 Teil 2 – Ausgewählte Beispiele

Walter Augustin

Skiseminar

WS 2004/05

Univ.Prof. Dr. Rony G. Flatscher



# Struktur



- **Grundlagen**
  - Open Office, Java UNO, BSF, Rhino, Rexx, VBScript, PHP
  - Architektur-Gesamtbild
- **Techniken**
  - Arbeiten mit der Objekthierarchie
  - Ansprechen von Klassen, Reflection
- **Beispiele**
  - Scalc, Swriter, Sdraw, Simpress
- **OLE**
- **Konklusio, Ausblick**

# Grundlagen: Open Office



- **Calc, Writer, Impress**
- **Ursprünglich kommerziell als Star Office**
  - Abspaltung von Star Office: Open-Source-Variante
- **Programmiert in C++**
- **UNO-Interface, Java UNO**
- **OpenOffice.org Basic oder Java als Makrosprache**

# Grundlagen: Java UNO



- **Universal Network Objects**
  - Einheitliche Klassenarchitektur
  - Interaktion von Komponenten
    - verschiedener Sprachen
    - auf verschiedenen Betriebssystemen
    - in verschiedenen Netzen (lokal oder remote)
  - Betriebssysteme: Linux, Solaris, Windows
  - Sprachen:
    - Nativ: Java, C++, OpenOffice.org Basic
    - Über Microsofts COM: VB, Delphi, PHP, ...
- **Hohe Erweiterbarkeit**
- **Hohe Komplexität**



- **Bean Scripting Framework**
  - Entwickelt von IBM
  - Der Apache Software Foundation überlassen
- **Brücke zwischen Script Engines und Java-Klassen**
  - Entwicklung von BSF-Engines für verschiedene Programmiersprachen möglich
  - Rhino (Javascript), Jython (Python), Tcl, NetRexx, Jruby, Groovy, ...
  - BSF4Rexx

# Grundlagen: Rhino



- **Skriptsprache (ECMAScript)**
- **Entwickelt von Netscape als „Javascript“**
- **Nunmehr Open Source, Mozilla.org**
  - „Javascript für Java“
  - Gedacht für Scripting von größeren Applikationen
- **Engine für BSF verfügbar**
  - Sehr mächtige Kombination

# Grundlagen: Object Rexx, BSF4Rexx



- **Skriptsprache**
- **Rexx: Entwickelt von IBM**
- **Grundausstattung verschiedenster IBM-Plattformen**
  - Mainframe, Midrange, PC
- **Weiterentwicklung: Object Rexx**
  - Ermöglicht Windows-Scripting
- **Weiterentwicklung: BSF4Rexx**
  - Ebenfalls mächtige Kombination

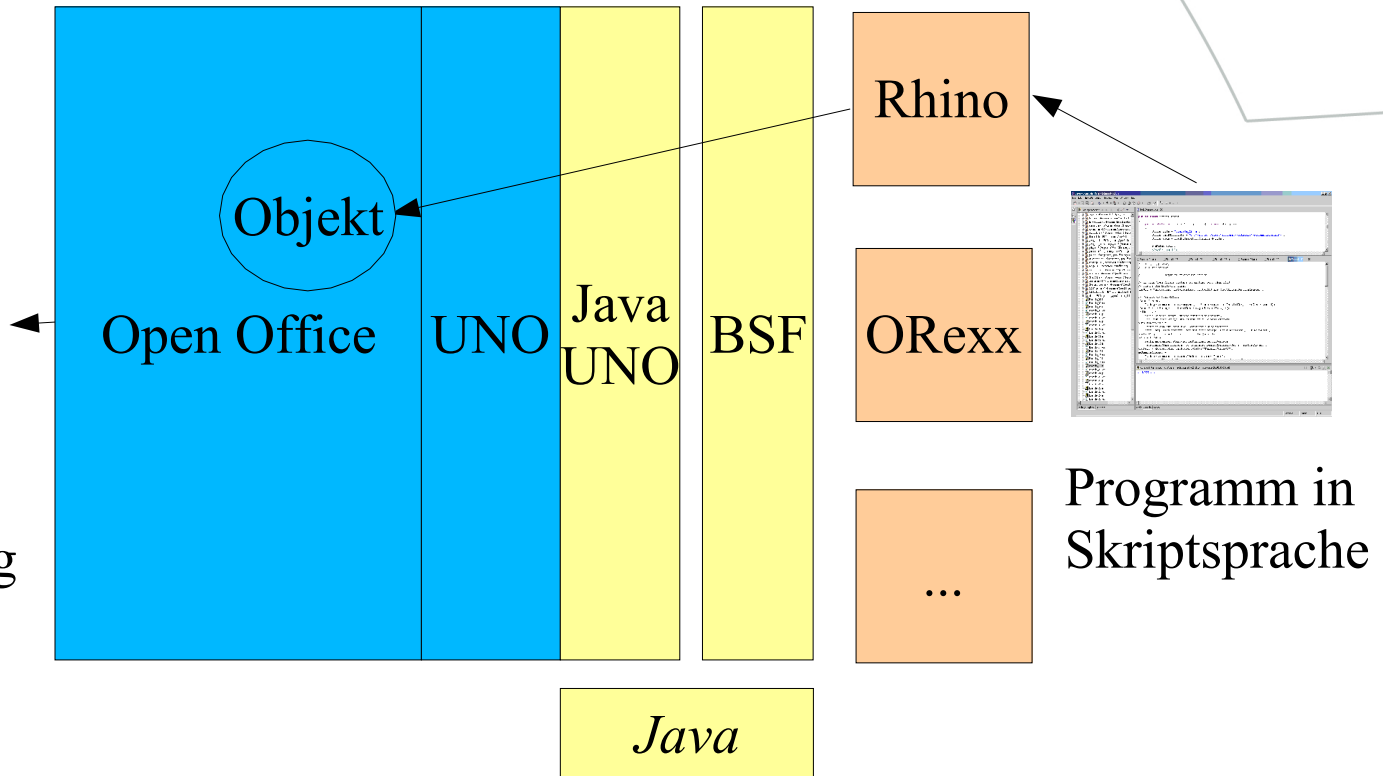
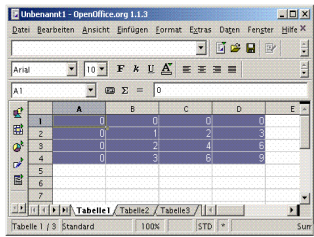
# Grundlagen: Weitere Skriptsprachen



- **Visual Basic Script**
  - Basic-Syntax (60er Jahre)
  - COM-fähig (Batchsprache für Windows)
  - Verwandtschaft zu Visual Basic: verschiedene Varianten, verschiedene Schwerpunkte
- **PHP**
  - C/Java-Syntax (1970)
  - Objektorientierung an Bord
  - Vereint OO und Strukturierte Programmierung
  - Schwerpunkt Web-Entwicklung
  - Auch Kommandozeileninterpreter verfügbar
  - COM-fähig

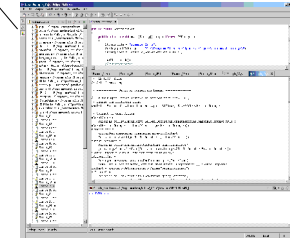
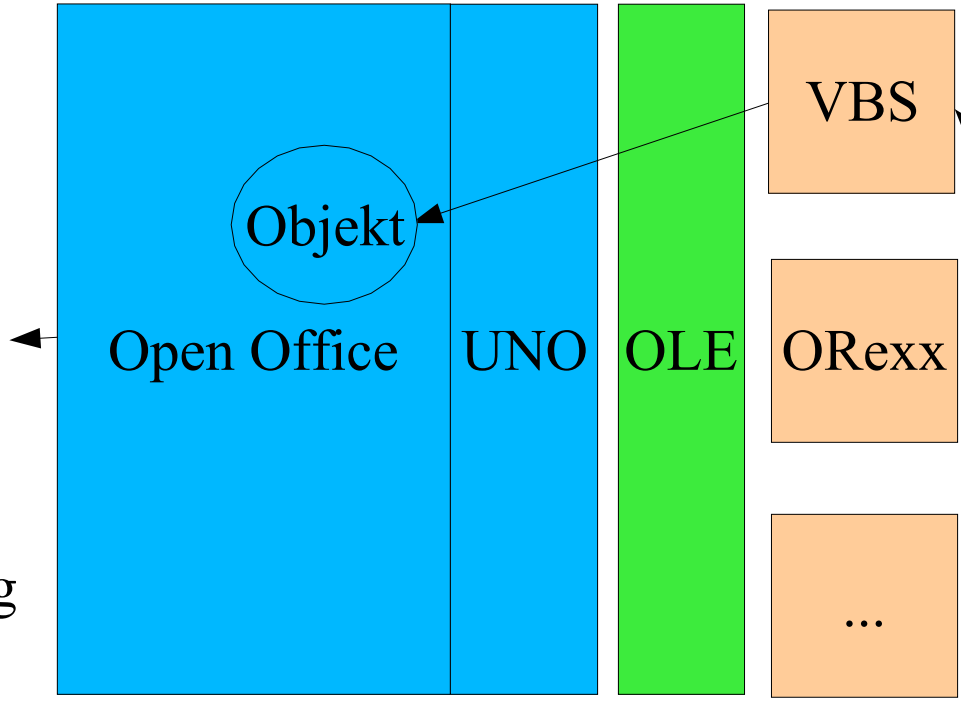
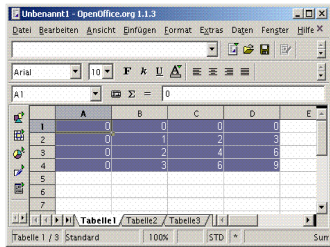


# Grundlagen: Das Gesamtbild – BSF



Resultat:  
Fernsteuerung  
von OO

# Grundlagen: Das Gesamtbild - OLE



Resultat:  
Fernsteuerung  
von OO

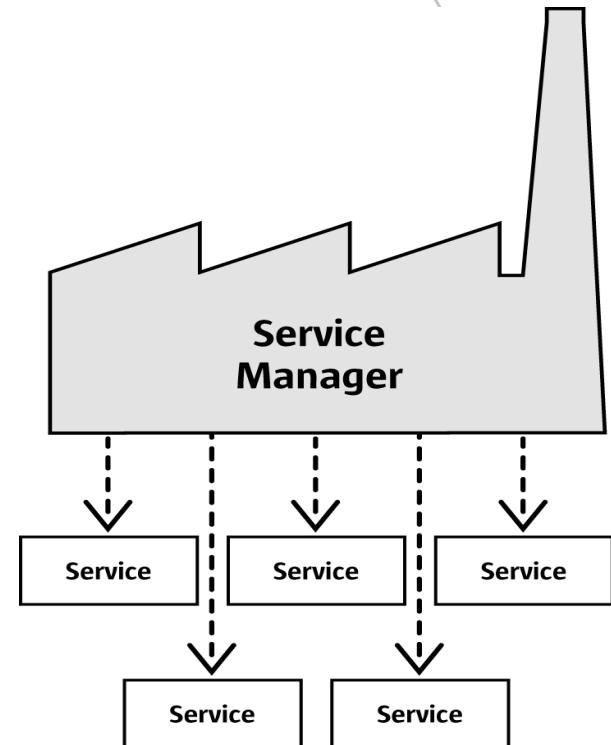
Programm in  
Skriptsprache

**OLE**  
**Windows-only!**

# Techniken: Arbeiten mit der Objekthierarchie (1)

- **Factory Pattern**

- Factory = „Service Manager“
- Wichtigster Service Manager: Desktop
- **Zweck: „Services“ (Objekte, die das System repräsentieren) erzeugen**



*Grafik: Open Office SDK Developer's Guide*

# Techniken: Arbeiten mit der Objekthierarchie (2)



- **Jeder Service existiert in einem Component Context**
  - Services = die Objekte, mit denen wir arbeiten
  - Component = eine Applikation von Open Office (zB das Tabellenkalkulationsprogramm)
- **Zusammenfassung – „UNO-Lingo“:**
  - „Service Manager“ = Factory (Objekterzeuger)
  - „Service“ = Objektinstanz (Interfaces + Properties)
  - „Component“ = eine der Office-Applikationen

*„Ich will aus der Textverarbeitung heraus drucken“ =  
„Ich brauche das XPrintable-Interface im Kontext der Komponente Swriter“ =  
`unoRuntime.queryInterface(com.sun.star.view.XPrintable, xWriterComponent);`*

# Techniken: Arbeiten mit der Objekthierarchie (3)



- **Objekte erzeugen**

- In den meisten Fällen nicht einfach mit *new* möglich, sondern:

- 1. Via Service Manager:**

- `desktop.loadComponentFromURL(Applikationsname, ...)`
- `sm.createInstanceWithContext(Klassenname, Kontext)`

- 2. Via Uno Runtime: Interface-Objekte aus Komponenten**

- `unoRuntime.queryInterface(Klassenname, Komponente)`

- 3. Über andere Objekte:**

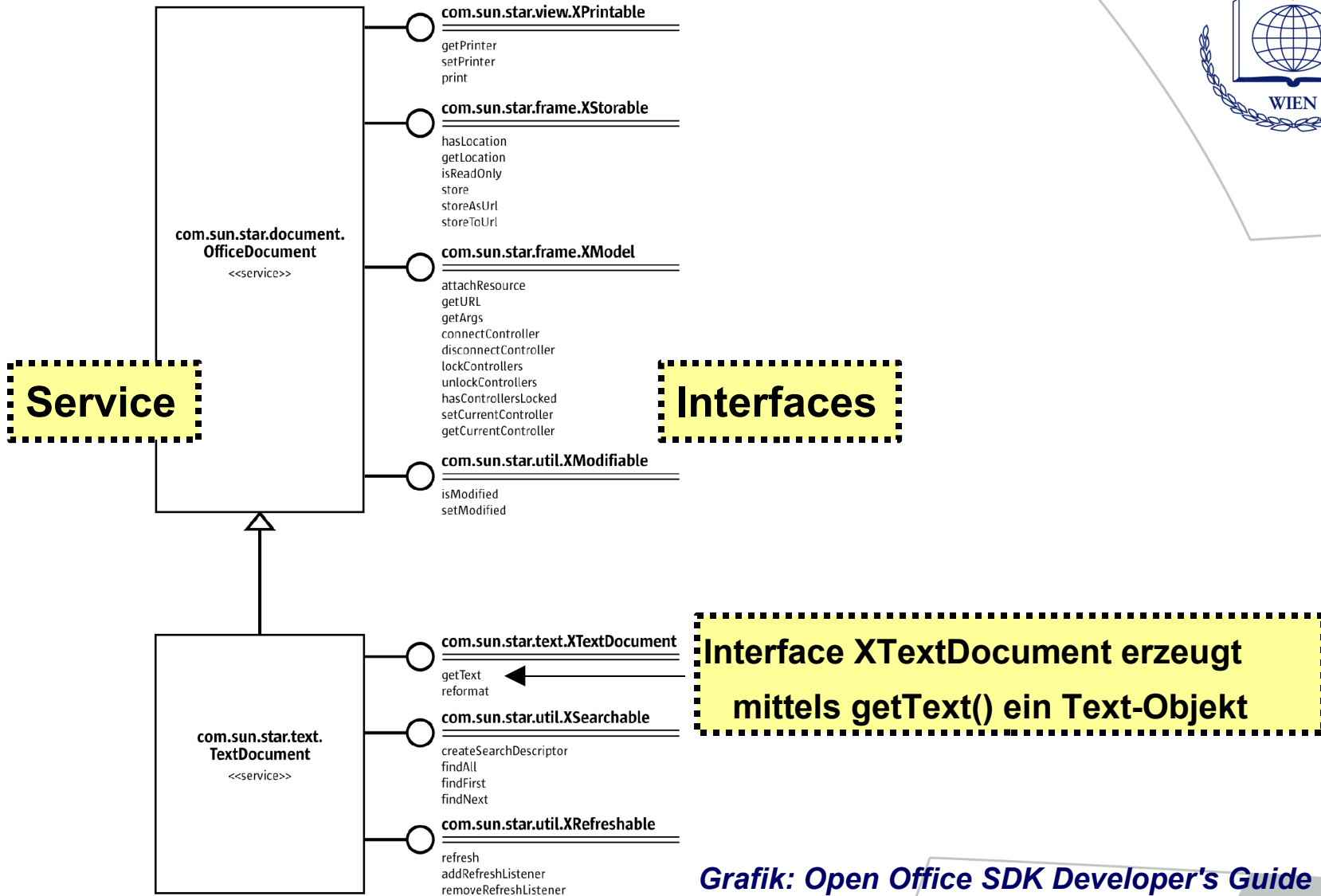
- `calcComponent.getSheets()`

# Techniken: Arbeiten mit der Objekthierarchie (4)



- **Interfaces (im Sinn von UNO)**
  - Ein Set von Methoden, die zusammen einen Aspekt eines Service ausmachen
  - Beispiel: XPrintable-Interface: print(), getPrinter(), setPrinter()
  - Vorteil: Wiederverwendbarkeit, Einheitlichkeit
- **Unterschied zu herkömmlichem Begriff**
  - Nicht nur Gestaltungsvorschrift
  - Ich kann mir ein „lebendiges“ Interface mit queryInterface() erzeugen lassen und damit arbeiten
  - Arbeit wird übersichtlicher
    - *seminar.biertrinken()* oder
    - *xAbendunterhaltung = unoRuntime.queryInterface („com.wu.XAbendunterhaltung“, seminar);  
xAbendunterhaltung.biertrinken()*

# Techniken: Arbeiten mit der Objekthierarchie (5)



Grafik: Open Office SDK Developer's Guide

# Techniken: Ansprechen von Klassen in Rhino



- **Java-eigene Klassen:**
  - java.\*
- **Klassen aus externen Packages:**
  - Packages.\*
- **Beispiele:**
  - `java.lang.System.out.println(„Hallo“);`
  - `unoRuntime = Packages.com.sun.star.uno.UnoRuntime;`
  - `pVal = new Packages.com.sun.star.beans.PropertyValue();`

**Pointer**

**Objekt**



# Techniken: Ansprechen von Klassen in Rexx (1)



- **Objekt instanzieren:**
  - `pVal = .bsf~new(„com.sun.star.beans.PropertyValue“)`
- **Methoden aufrufen:**
  - `blah = myObj~getSomething(„info“)`
  - `things = myObj~getThings`
- **Felder ansprechen:**
  - `theName = bsf(„getFieldValue“, myJavaClass, „name“)`
  - `theConstant = bsf(„getStaticValue“, „java.lang.Math“, „PI“)`

**Die Tilde ~ ist der Nachrichtenoperator  
und heißt bei Object Rexx „Twiddle“**

# Techniken: Ansprechen von Klassen in Rexx (2)



- „Type hinting“ kann manchmal notwendig sein
  - `cellProperties~bsf.invokeStrict(„setPropertyValue“, „ST“, „CellStyle“, „ST“, „My Style“)`
- **Entsprechung in Java:**
  - `cellProperties.setPropertyValue(„CellStyle“, „MyStyle“)`
- **Preisfrage: Wozu?**

**Um die richtige Signatur zu finden**

# Techniken: Reflection (1)



- **Objektorientierte Programmiersprachen**
- **Klassen, die andere Klassen „spiegeln“**
  - Auskünfte über Beschaffenheit von Klassen
- **Bietet rein objektorientierte Art, Klassen zu instanzieren**
  - ohne spezifische Syntax-Vereinfachungen
- **Beispiel:**
  - `PropertyValue myArray = new PropertyValue[10];`
- **entspricht:**
  - `Object myArray = java.lang.reflect.Array.newInstance(  
Packages.com.sun.star.beans.PropertyValue, 10);`

# Techniken: Reflection (2)



- **Nicht nur Programmiersprachen bieten Reflection-Dienste**
  - Open Office: UNO-Reflection
  - Notwendig, wenn Java nicht direkt zur Verfügung steht (so bei OLE-Programmierung)
- **Zusammenfassend: Reflection ...**
  1. liefert Infos über Klassen
  2. hilft, Objekte zu instanzieren und anzusprechen,
    - wenn Java-Kurzschreibweise nicht verfügbar
    - oder wenn Java überhaupt nicht verfügbar

# Beispiele: Verbindungskopf in Rhino



## URL-Definition

```
unoUrl = "uno:socket,host=localhost,port=8100;urp;StarOffice.ServiceManager";
```

## Wo läuft der Server?

```
xLocalContext =
```

```
    Packages.com.sun.star.comp.helper.Bootstrap.createInitialComponentContext(null);  
xLocalServiceManager = xLocalContext.getServiceManager();  
urlResolver = xLocalServiceManager.createInstanceWithContext(  
    "com.sun.star.bridge.UnoUrlResolver", xLocalContext);
```

```
xUnoUrlResolver = xLocalContext.getServiceManager().createInstanceWithContext(  
    "com.sun.star.bridge.UnoUrlResolver", xLocalContext);
```

**Aus lokalem Kontext heraus Verbindung zum „entfernten“ Server herstellen**

```
initialObject = xUnoUrlResolver.resolve(unoUrl);
```

**„Initial Object“ enthält Informationen über die Verbindung**

```
xPropertySet = java.lang.Class.forName("com.sun.star.beans.XPropertySet", initialObject);  
cont = xPropertySet.getPropertyValue("Context");
```

**Damit kann ich mir einen „Remote Service Manager“ erzeugen lassen**

```
mxRemoteContext = java.lang.Class.forName("com.sun.star.uno.XRemoteContext", cont).newInstance(cont);  
mxRemoteServiceManager = mxRemoteContext.getServiceManager();
```

**Der „Remote Service Manager“ erzeugt mir meinen „Desktop“**

```
desktop = mxRemoteServiceManager.createInstanceWithContext("com.sun.star.frame.Desktop", mxRemoteContext);
```

**Der Desktop hat das Interface „Component Loader“**

```
mxComponentLoader = desktop.getComponentLoader();  
xComponentLoader = java.lang.Class.forName("com.sun.star.frame.XComponentLoader", desktop);
```

**Der „Component Loader“ erzeugt in der Folge die Applikationsfenster**

# Beispiele: Umgang mit Dokumenten (1)



- **Dokument öffnen:**

scalc	Tabellenkalkulation
swriter	Textverarbeitung
simpres	Präsentationsprogramm
sdraw	Zeichentool

```
-- We need no properties  
propertyValueName = .bsf4rexx~Class.class~forName  
  ("com.sun.star.beans.PropertyValue")  
loadProps = .bsf~createArray(propertyValueName, 0)  
  
xWriterComponent = xComponentLoader~loadComponentFromURL( ,  
  "private:factory/swriter", "_blank", 0, loadProps)
```

```
::requires "BSF.cls"
```

file:///c:/documents/myfile.sxw
http://www.blah.com/myfile.sxw
ftp://111.111.111.111/myfile.sxw

# Beispiele: Umgang mit Dokumenten (2)



- **Drucken:**

```
// Get a "Printable" object
xPrintable = Printable-Interface aus der Writer-Komponente holen
    Packages.com.sun.star.uno.UnoRuntime.queryInterface(
        java.lang.Class.forName("com.sun.star.view.XPrintable"),
        xWriterComponent);
```

```
// We need to define a printer name
printerDesc = new Array();
printerDesc[0] = new Packages.com.sun.star.beans.PropertyValue();
printerDesc[0].Name = "Name";
printerDesc[0].Value = "HP PSC 950"; // This is to your printer name!
xPrintable.setPrinter(printerDesc);
```

```
// Print options are properties, too: which pages to print
printOpts = new Array();
printOpts[0] = new Packages.com.sun.star.beans.PropertyValue();
printOpts[0].Name = "Pages";
printOpts[0].Value = "1";
```

```
// This prints the (empty) text
xPrintable.print(printOpts); Druck starten
```

**Einstellungen vornehmen (bzgl. Drucker und Druckvorgang)**

# Beispiele: Scalc – die Insert-Funktion



```
// Test for the insert function:  
// (1) insert a value (number)  
insert(1, 1, 123.456, xSheet, true);  
// (2) insert a text string (formula)  
insert(1, 2, "Hello", xSheet, false);
```

**Test mit Zahl und Text**

```
// A function for inserting values or formulas into cells  
// Parameters:  
// x, y      cell coordinates  
// content   value or formula to insert  
// container the spreadsheet or cell range where the cell resides  
// isValue   (true or false) indicates whether the content should be  
//           regarded as a number or a character string
```

```
function insert(x, y, content, container, isValue)  
{  
  oCell = container.getCellByPosition(x, y);  
  if(isValue)  
    oCell.setValue(new java.lang.Float(content).floatValue());  
  else  
    oCell.setFormula(content);  
}
```

**Rhino-Funktion definieren**



# Beispiele: Scalc - Zellformatierung



- (get) **Neuen (leeren) Style im „Style Family Name Container“ der Komponente erzeugen**

```
oStyleFamilyNameContainer~insertByName("My Style", style)
```

- **Dem leeren Style mittels „Property Set“ ein paar Farbeigenschaften zuweisen**

```
oCellProperties1 = unoRuntime~queryInterface(oCellProperties1Name, style)
oCellProperties1~setProperty("IsCellBackgroundTransparent", .false)
oCellProperties1~setProperty("CellBackColor", .bsf~new("java.lang.Integer", 6710932))
-- = hex 66 66 94
oCellProperties1~setProperty("CharColor", .bsf~new("java.lang.Integer", 16777215))
-- = hex FF FF FF
```

- **Zieldefinition: „Cell Range“ festlegen**

```
cellRange = xSheet~getCellRangeByPosition(0, 0, 3, 3)
```

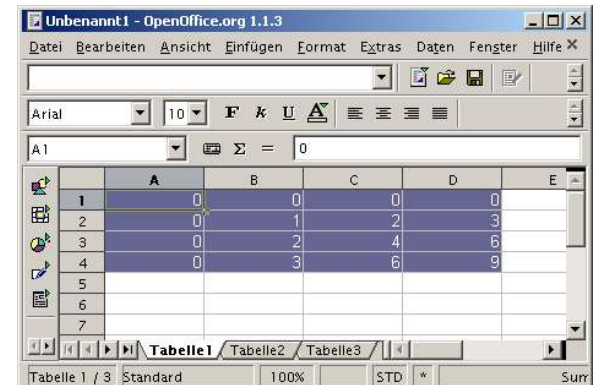
```
-- get existing properties of cells in range
```

```
cellProperties = unoRuntime~queryInterface(oCellPropertiesName, cellRange)
("con")
```

```
cellProperties = unoRuntime~queryInterface(oCellPropertiesName, cellRange)
```

- **Dem „Cell Property“ namens „CellStyle“ meinen Style zuweisen**

```
cellProperties~bsf.invokeStrict("setProperty", "ST", "CellStyle", "ST", "My Style")
```



# Beispiele: Swrite – Textmanipulation



- **(gekürzt)**

```
// Open a blank text document
```

```
loadProps = new Array(); // no properties
```

```
x
```

**Neues Textdokument öffnen und das Objekt „Text“ herausholen**

```
..., loadProps);
```

```
xTextDocument =
```

```
    ...queryInterface("com.sun.star.frame.XTextDocument", xWriterComponent);
```

```
x
```

**Über das Model-Interface der Writer-Komponente bekomme ich**

```
xWriterComponent);
```

→ einen Controller

→ einen ViewCursorSupplier

→ einen ViewCursor

```
xModel = ...queryInterface("com.sun.star.frame.XModel", xWriterComponent);
```

```
xController = xModel.getCurrentController();
```

```
xViewCursorSupplier =
```

```
    ...queryInterface("com.sun.star.text.XTextViewCursorSupplier", xController);
```

```
xViewCursor = xViewCursorSupplier.getViewCursor();
```

```
x
```

**In ein neues Set von View-Cursor-Properties schreibe ich einen Formatvorlagennamen**

```
xCursorPropertySet = ...queryInterface("com.sun.star.beans.XPropertySet", xViewCursor);
```

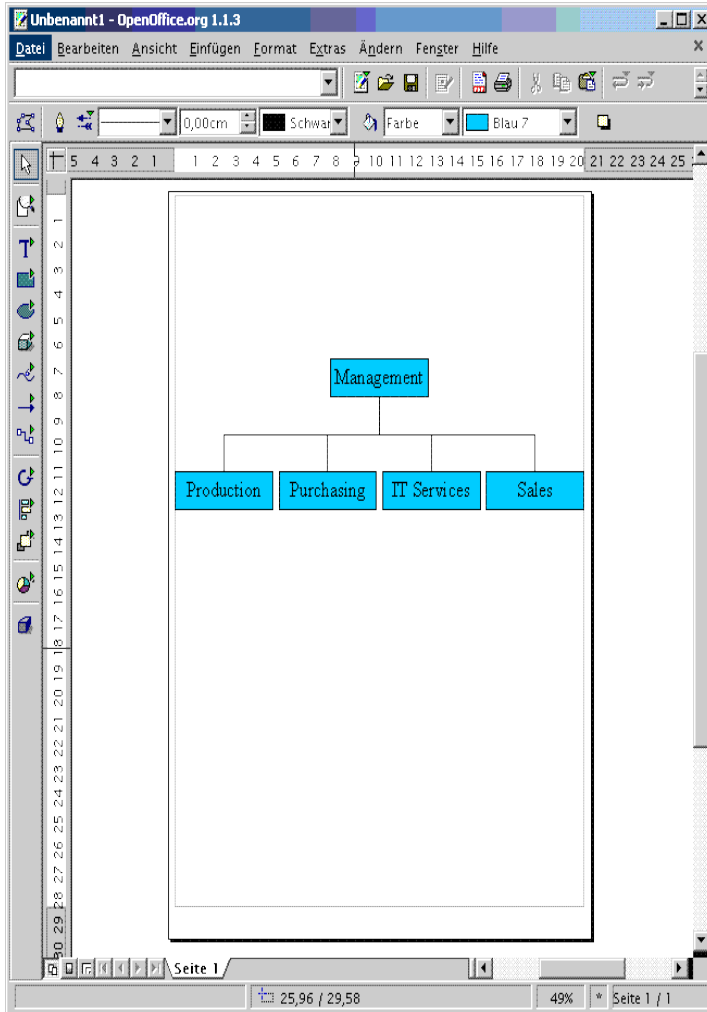
```
x
```

**Der View Cursor gibt mir den Page Cursor → Abfrage der Seitennummer**

```
xPageCursor = ...queryInterface("com.sun.star.text.XPageCursor", xViewCursor);
```

```
xText.setString("The current page number is " + xPageCursor.getPage());
```

# Beispiele: Sdraw - Organigramm



1. Sdraw öffnen
2. Draw Component->xDrawPagesSupplier
3. xDrawPagesSupplier->getDrawPages
4. Draw Page = xDrawPages.getByIndex(0)
5. Draw Page -> Page Properties
6. Page Properties->Page width etc ermitteln
7. Positionen der Grafikelemente errechnen
8. Für alle Abteilungsnamen  
Document Factory -> create Instance(shape)
9. SetPosition(errechnete Position)
10. Draw Page.add(shape)
11. Analog mit Verbindungslinien

# OLE: Allgemeines



- **COM (Component Object Model)**
  - RPC ("remote procedure call")
  - Weiterentwicklungen: DCOM, COM+
- **OLE (Object Linking and Embedding)**
  - COM-basiert
  - Verknüpfen von Dokumenten
    - (Dynamischer Datenaustausch)
  - Einbetten von fremden Dokumenten

*Quelle: Prof Dr. Rony Flatscher, Folien zu POOLV*

# OLE: Swrite mit VBS steuern



- **VBS = Visual Basic Script, Basic-Syntax**

```
'The service manager is always the starting point
'If there is no office running then an office is started up
Set objServiceManager = WScript.CreateObject("com.sun.star.ServiceManager")

'Create the Desktop
Set objDesktop = objServiceManager.CreateInstance("com.sun.star.frame.Desktop")

'Open a new empty writer document
Dim args()
Set objDocument= objDesktop.loadComponentFromURL( _
    "private:factory/swriter", "_blank", 0, args)

'Create a text object
Set objText= objDocument.getText

'Create a cursor object
Set objCursor= objText.createTextCursor

'Inserting some Text
objText.insertString objCursor, _
    "The first line in the newly created text document." & vbLf, false
```

**Vereinfachter Weg zum Service Manager**

**Weiter wie bei BSF**

# OLE: Swrite mit PHP steuern



- **PHP = PHP Hypertext Preprocessor, C-Syntax**

```
// The service manager is always the starting point
// If there is no office running then an office is started up
$objServiceManager = new COM("com.sun.star.ServiceManager");

// Create the Desktop
$objDesktop= $objServiceManager->createInstance("com.sun.star.frame.Desktop");

// Open a new empty writer document
$args = array();
$objDocument= $objDesktop->loadComponentFromURL(
    "private:factory/swriter", "_blank", 0, $args);

// Create a text object
$objText= $objDocument->getText();

// Create a cursor object
$objCursor= $objText->createTextCursor();

// Inserting some Text
$objText->insertString($objCursor,
    "The first line in the newly created text document.\n", false);
```

# Konklusio



- **Open Office-Automatisierung (abseits von OpenOffice.org Basic) ist in zweierlei Hinsicht kompliziert:**
  - **Sprache:** Java (starke Typisierung, starke Objektorientierung)
  - **Programmierschnittstelle (Java UNO):** Sehr komplex
- **Zwei Ansätze:**
  1. **BSF:** Vermindert Komplexität der *Sprache* – aber nicht die der Objekthierarchie
  2. **OLE:** Etwas einfacher, aber
    - (a) Abhängigkeit von der Implementation in der Skriptsprache
    - (b) nur für MS Windows verwendbar



- **Nicht nur Sprache vereinfachen, sondern auch Zugriff auf Objekte**
- **Entwicklung einer „Meta-API“ analog zum DOM-Modell für Webseiten:**

```
spreadsheet.insertIntoCell(0, 0, 123.45);  
spreadsheet.colorizeCell(0, 0, "blue");
```

- Möglichkeit des umfassenden Zugriffs auf Java UNO sollte jedoch erhalten bleiben
- **Alternative: Funktionsbibliotheken für jede Skriptsprache**
- **In Entwicklung:**
  - Vereinfachung der UNO-Schnittstelle in der neuen Open Office-Version (April 2005) geplant



# Danke für Ihre Aufmerksamkeit



- **Wir stehen für Fragen zur Verfügung**
- **Download der Arbeit, dieser Folien sowie der Beispielskripte:**
  - [http://www.matto.at/oo\\_examples/](http://www.matto.at/oo_examples/)
  - OOBabylon
  - Live JavaDoc
  - 31 Beispielskripte
    - 14 in Rhino
    - 14 in Object Rexx
    - OLE-Beispiel in VB, Object Rexx, PHP
  - Die Arbeit enthält eine Sammlung weiterer Download-Links