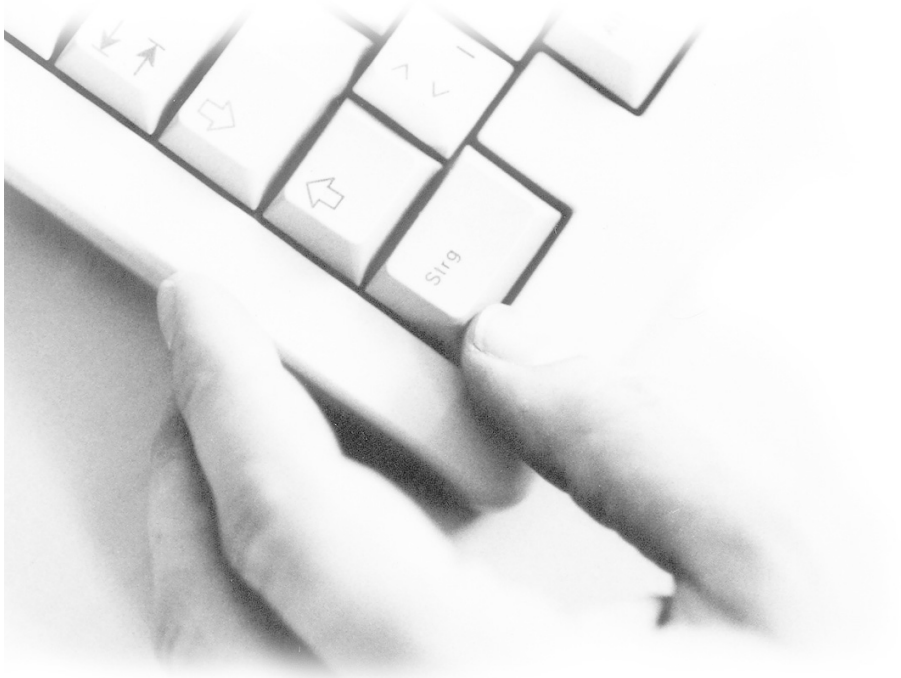
.riessapplications

Herzlich Willkommen.



.riessapplications

Java als Makro-Sprache für OpenOffice mit Eclipse

Quelle: http://specs.openoffice.org/scripting_framework/index.html

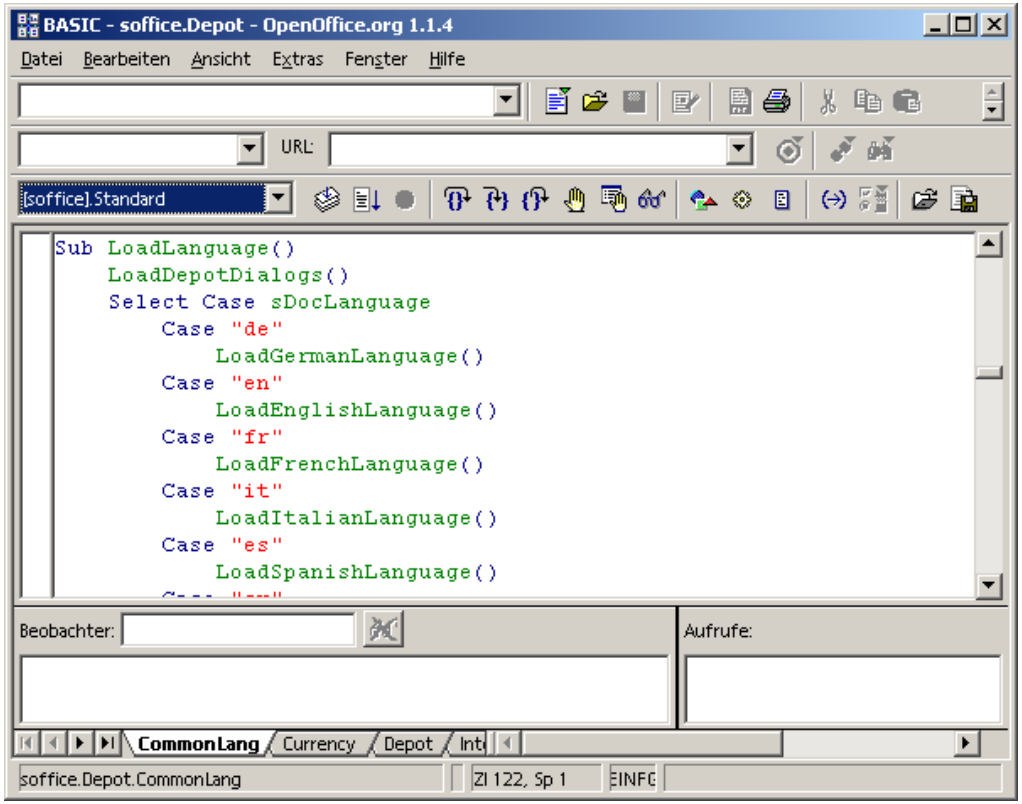
.riessapplications

Übersicht

- Makros
  - Historie
  - Szenarien/Einsatzzwecke
- Warum OpenOffice?
  - ScriptingFramework
- Java als Makrosprache
  - UNO
- Warum Eclipse?
  - PlugIn-Konzept
  - einfaches Makro-Deployment

Makros

- Historie
  - Basic!
  - Basic!
  - Basic!
  - ...



Makros

- Szenarien/Einsatzzwecke
  - wiederholbare Vorgänge
    - automatisieren
    - vereinfachen
    - standardisieren
  - Dokumente
    - um Zusatzfunktionalität erweitern
    - als Basis für Applikationen verwenden
    - als „Träger“ für Funktionalität verwenden

.riessapplications

Übersicht

- Makros
  - Historie
  - Szenarien/Einsatzzwecke
- Warum OpenOffice?
  - ScriptingFramework
- Java als Makrosprache
  - UNO
- Warum Eclipse?
  - PlugIn-Konzept
  - einfaches Makro-Deployment

.riessapplications

Warum OpenOffice?

- OpenSource
- frei
- Unterstützung vieler Plattformen
- Unterstützung vieler Programmierungsprachen
- Benutzung offener Standards (XML)
- viele Import-/Export-Filter (u.a. MS Office, PDF)

.riessapplications

Warum OpenOffice?

- ScriptingFramework
  - (z.Z.) Makros schreiben in
    - OpenOffice.org Basic
    - Java
    - JavaScript
    - BeanShell
  - Skalierbarkeit
    - Mechanismus zum Hinzufügen weiterer Sprachen

Warum OpenOffice?

- ScriptingFramework
  - Makro-Code kann assoziiert werden an
    - Menüs
    - Tastatureingaben
    - Anwendungs-/Dokumentereignisse
    - GUI-Elemente in Formularen
    - GUI-Elemente in Dialogen
    - Hyperlinks
    - Grafiken usw.

.riessapplications

Übersicht

- Makros
  - Historie
  - Szenarien/Einsatzzwecke
- Warum OpenOffice?
  - ScriptingFramework
- Java als Makrosprache
  - UNO
- Warum Eclipse?
  - PlugIn-Konzept
  - einfaches Makro-Deployment

Java als Makrosprache

- UNO
  - Standard-UNO-Programmierung
  - Makros erhalten XScriptContext
    - Zugriff auf Dokument, Desktop, ComponentContext
  - parcel-descriptor definiert enthaltene Makros
    - plus optionale Metadaten
    - plus optionale Abhängigkeiten zu JARs
  - Klasse entspricht „Module“
  - Library wird als Unterordner definiert (ist KEIN Package!

Java als Makrosprache

- Schritte zum eigenen Java-Makro
  - Java-Klassen erstellen (Default-Package!)
  - Methode mit vorgegebener Signatur erstellen
    - public static void NAME(XScriptContext c, ...)
  - Parcel erzeugen
    - Java-Klassen
    - parcel-descriptor.xml
  - Parcel an Dokument deployen
    - Tools aus SFrameworkInstall.jar
  - Debugging über Java-Standard-Mechanismen

.riessapplications

Übersicht

- Makros
  - Historie
  - Szenarien/Einsatzzwecke
- Warum OpenOffice?
  - ScriptingFramework
- Java als Makrosprache
  - UNO
- Warum Eclipse?
  - PlugIn-Konzept
  - einfaches Makro-Deployment

Warum Eclipse?

- außergewöhnliche Programmierhilfen
  - Refactoring
  - Code-Assist
  - Formatter
- gute Antwortzeiten
- Echtzeitübersetzung
- quasi-plattformunabhängig
  - SWT
- mächtiges PlugIn-Konzept

Warum Eclipse?

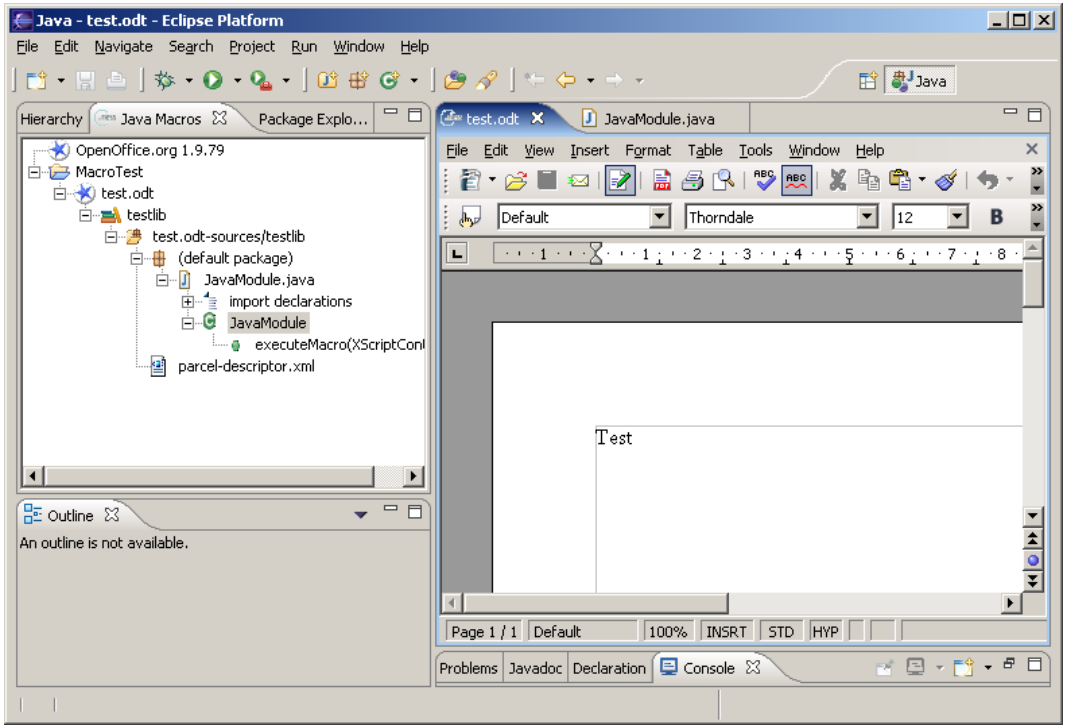
- PlugIn-Konzept
  - Implementierung neuer Views
    - View für Makros?
  - Implementierung neuer Editoren
    - OOoBean?
  - Integration in den Übersetzungsprozeß
    - Echtzeit-Deployment von Makros?

Warum Eclipse?

- einfaches Makro-Deployment
  - Implementierung
    - eigener View für Makros
    - eigener Editor für Dokumente
    - Erweiterung des Build-Prozesses
    - Erstellung passender Remote-Debug-Konfiguration

Warum Eclipse?

- einfaches Makro-Deployment
  - DEMO!



Vielen Dank für Ihre Aufmerksamkeit

- Kontakt
  - .riess applications gmbh
    Mathias Supp
    Draisstraße 10
    76307 Karlsbad-Langensteinbach
    mathias.supp@riess.de
    07202/707-0
- Fragen?