

**Common Experiences using the
OpenOffice.org API
by
Berend Cornelius
bc@openoffice.org**

About the speaker

- Name: Berend Cornelius
- Title: Product Specialist
- Programming AutoPilots, macros and internal tools
- Localization of Templates and Documents in OpenOffice.org/StarOffice

- About me
- The “old” StarOffice API and the new one
- Access from OpenOffice.org Basic
- Access from Java
- Creating user interfaces
- Common methodology
- Conclusion/questions



The “Old” API and the “New” One

- The “Old” API
 - Designed to be a VBA Clone
 - Strongly oriented at the User Interface
 - Tightly bound to StarOffice Basic
 - Suitable for Macros only
- The “New” API
 - Concept of Components, Services, Interfaces
 - Language independent
 - Java
 - OpenOffice.org Basic
 - C++
 - Ole Automation
 - Python



Access from OpenOffice.org Basic

- Developed to be Visual Basic Clone
- Not typesafe, not Object oriented, no Thread support
- Easy to handle IDE
 - Lacks unfortunately an Object browser
 - Provides dialog editor
- Defines several Properties/Runtime Functions to enter the “world” of OpenOffice.org API like
 - StarDesktop, ThisComponent, CreateUnoService(),
dbg_supportedInterfaces, dbg_Properties, dbg_Methods
- Methods of Basic UNO objects can be called directly
- Conclusion: Suitable tool for little projects

- Requires good planning due to Object orientation
 - Possibility to create components
- No deployment within documents (Not yet)
- Set up your own work environment
 - Develop remotely
 - Debug within a java component
 - Register component with pkgchk.exe
- Create different proxy objects for each instantiated UNO Object
- General better performance compared to Basic
- Conclusion: Suitable for larger projects

Creating User Interfaces

- Swing
 - Java Look and Feel
 - Supports dynamic layout creation
 - Low performance
 - Modal dialogs cannot be attached to Office window
- Java AWT
 - Native dialogs with restricted variety of available controls
- Office API Dialogs and Controls (Toolkit Module)
 - “StarOffice Look and Feel”
 - XY layout that accounts for fontsize
 - Follows the model-view-controller paradigm
 - Dialog Editor for OpenOffice.org Basic
 - Creation during runtime in Java
- Forms in Documents
 - Database frontend
 - Work differently from Dialog controls

- SDK
 - Developer's Guide
 - IDL Files
 - Example Files
- Basic Migration Guide
- qadevOOo package
- Macro Recorder – Dispatch Framework
- OpenOffice.org mailing lists

- What will await the Developer
 - An API designed for high level components
 - Detailed documentation/support via newsgroups
 - Steady improvement of various language bindings and integration with other products
 - A stable, extremely powerful product
 - Future-proof technology due to the strong support of both the OpenOffice community and Sun Microsystems

- OpenOffice.org <http://www.openoffice.org>
- qadevOOo package
http://qa.openoffice.org/qadevOOo_doc/index.html

Questions?