
CIB GROUP

LibreOffice extension development

Tools, Tips & Tricks of the Trade

Thorsten.Behrens@cib.de



LibreOffice Eclipse Plugin

- > Work done during several GSoC periods by Cedric Bosdonnat (with contributions by Dan Corneanu, Oliver Specht, and Ludovic Smadja)
 - > Fixed & improved to work with LibreOffice by Samuel Mehrbrodt:
<https://smehrbrodt.wordpress.com/2016/01/08/libreoffice-and-eclipse-loeclipse-2-0-released/>
 - > Grab it here:
<http://eclipse-plugins.libreoffice.org/>
-

LibreOffice Eclipse Plugin: Selling points

- > Wizard-based extension development
 - > Streamlined deployment & debugging experience:
 - Deploy on press of a button, optionally into separate userdir
 - Debugging deployed extensions from inside Eclipse IDE
-

LibreOffice Eclipse Plugin: Debugging

The screenshot displays the Eclipse IDE in a debugging state. The main window title is "Debug - Test/source/org/libreoffice/example/comp/TestImpl.java - Eclipse Platform".

Debug Console: Shows the execution stack. The current frame is "TestImpl.<init>(XComponentContext) line: 22". Other frames include "NativeConstructorAccessorImpl.newInstance0(Constructor, Object)", "DelegatingConstructorAccessorImpl.newInstance(Object[])", and "Factory.newInstance(XComponentContext) line: 216".

Variables View: Lists the state of variables at the current line:

Name	Value
this	TestImpl (id=27)
m_adapter	null
m_xContext	null
context	\$Proxy6 (id=34)
h	JNI_proxy (id=45)
com.sun.star.bridges.jni_uno.JNI_proxy@4808e0e1	

Code Editor: Shows the source code of TestImpl.java. The constructor is expanded, and the line `m_xContext = context;` is highlighted, corresponding to the current execution point.

```

private final XComponentContext m_xContext;
private static final String m_implementationName = TestImpl.class.getName();
private static final String[] m_serviceNames = {
    "org.libreoffice.example.Test" };

public TestImpl( XComponentContext context )
{
    m_xContext = context;
};

public static XSingleComponentFactory __getComponentFactory( String sImplementationName,
XSingleComponentFactory xFactory = null;

    if ( sImplementationName.equals( m_implementationName ) )
        xFactory = Factory.createComponentFactory( TestImpl.class, m_serviceNames );
    return xFactory;
}
    
```

Outline View: Shows the class structure. The current method being debugged, `TestImpl(XComponentContext)`, is highlighted.

Status Bar: Shows "Writable", "Smart Insert", and the time "22:30".

NOA-Libre: Nice Access to Office API

- > high level functions for common stuff
- > Done by IOn AG, originally:
<http://ubion.ion.ag/solutions/004niceofficeaccess>
- > LibreOffice version here:
<https://github.com/LibreOffice/noa-libre>



NOA-Libre: Nice Access to Office API

- > lightweight UNO wrapper API for many UNO interfaces
 - > access to underlying UNO interfaces from wrapper objects
 - > Now able to use in-tree Office jars, and build against maven
 - > Used in production for a large OSGI-Plugin based Rich-Client Java application
-

JEXO: Jave Extension Helper

- > Developed by Benjamin Sponring, now at BRZ
- > Available here:
<http://sourceforge.net/projects/jexo/>
- > Same thrust as for NOA – simplify Office API usage



Further resources

- > Maven LibreOffice repo:
<http://repo1.maven.org/maven2/org/libreoffice/>
 - > Cedric's LOEclipse Tips & Tricks:
<http://bosdonnat.fr/tag/ooeclipse.html>
 - > Dmitri Popov tutorial: <http://www.linuxjournal.com/article/9412>
 - > LibreOffice extensions:
https://wiki.documentfoundation.org/Development/Extension_Development
 - > OOO ecosystem links:
 - https://wiki.openoffice.org/wiki/Extensions_development
 - <https://wiki.openoffice.org/wiki/JavaEclipseTuto>
 - And of course the Developer Guide as the ultimate compendium:
https://wiki.openoffice.org/wiki/Documentation/DevGuide/OpenOffice.org_Developers_Guide
 - > Need for extension/api discussion list or forum?
-

Thanks for watching!

Q & A
