



# Hogyan készítsünk új Writer funkciókat?

- ▼ Vajna Miklós
- ▼ 2012. december 8.



All text and image content in this document is licensed under the [Creative Commons Attribution-Share Alike 3.0 License](#) (unless otherwise specified). "LibreOffice" and "The Document Foundation" are registered trademarks. Their respective logos and icons are subject to international copyright laws. The use of these therefore is subject to the [trademark policy](#).

# Bevezetés

Nem csak a funkciók számítanak

# Hogyan segíthet a közösség egy tagja

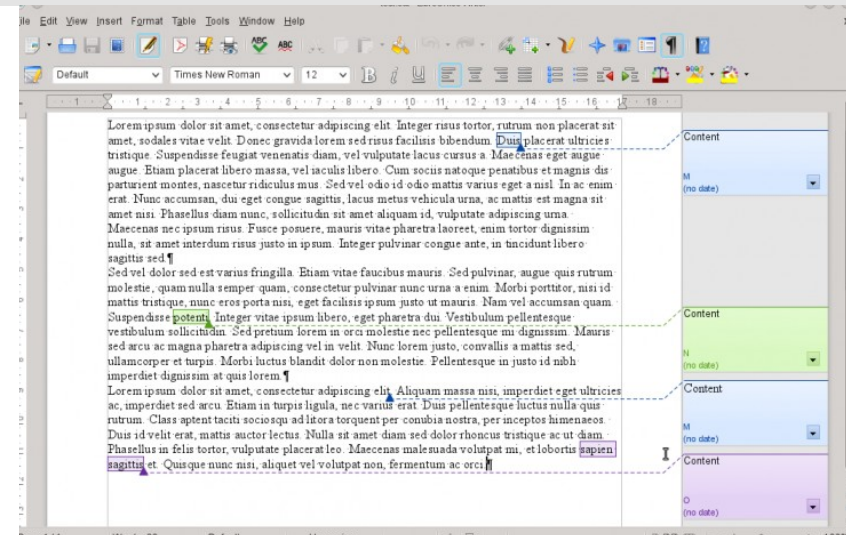
- ▼ Felhasználók segítése (levelezőlista, fórum, IRC)
- ▼ Minőségbiztosítás (hibajelentések, hibák megerősítése)
- ▼ Fejlesztés
- ▼ Útmutatók írása
- ▼ Design
- ▼ Infrastruktúra üzemeltetése

# Hogyan segíthet egy fejlesztő?

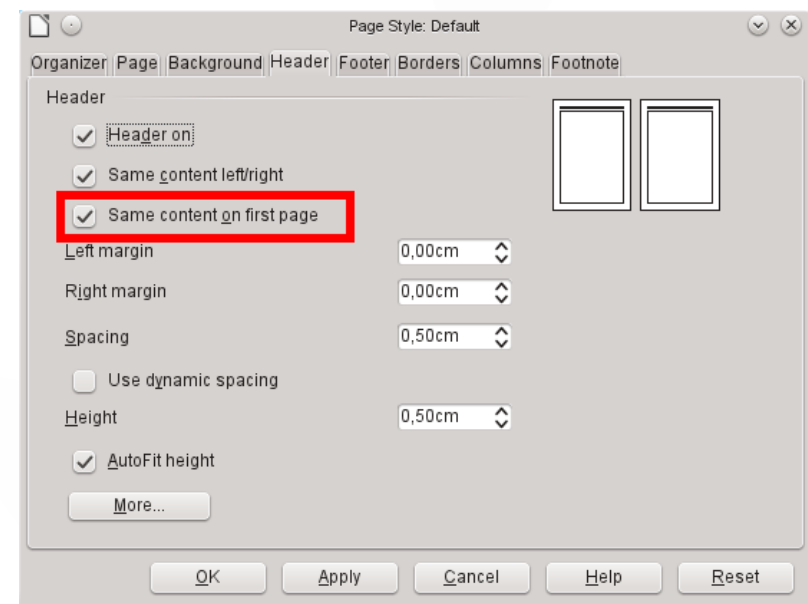
- ▼ Hibák javítása
- ▼ Tesztelés
- ▼ Új funkciók készítése
- ▼ Refaktorálás
- ▼ Mentorálás
- ▼ Dokumentálás

# Példák: új Writer funkciók a LibreOffice 4.0-ban

## ▼ Szövegtartományhoz hozzászólás



## ▼ Más fejléc / lábléc az első oldalon



# Egy új funkció lépései

- ▼ Dokumentum modell
- ▼ UNO API
- ▼ Megjelenítés (layout)
- ▼ Szűrők
- ▼ Felhasználói felület
- ▼ Tesztek
- ▼ Dokumentáció
- ▼ Specifikáció

# Dokumentum modell

Azt lehetne gondolni, hogy csak ennyi szükséges

# Mi a dokumentum modell?

- ▼ Klasszikus Modell-View-Controller
- ▼ Egy dokumentum ↔ **SwDoc**
  - ▼ Építőkocka: paragrafusok ↔ **SwNode**
  - ▼ Forrás az **sw/source/core/** alatt
  - ▼ Nincs külön szöveges csomópont a szövegtartományoknak, helyette **SwpHints**
    - ▼ Benne **SwTxtAttr** minden tulajdonsághoz: kezdet/vég/érték
  - ▼ Sok más kollekció, például:
    - ▼ **SwPageDesc** : oldalstílusok
- ▼ Hozzáadásra került az „első oldali fejléc” funkcióhoz:
  - ▼ **SwPageDesc::aFirst**

# Hogyan fedezzük fel a dokumentummodellt?

## ▼ XML kiírás:

```
$ SW_DEBUG=1 ./soffice.bin
```

## ▼ A Shift-F12 létrehoz egy **nodes.xml** file-t

## ▼ UNO-n keresztül:

```
enum = ThisComponent.Text.createEnumeration  
para = enum.NextElement  
para = enum.NextElement  
xray para
```

## ▼ Ugyanez szövegtartományokra

# UNO API

Hogy makrókból lehessen manipulálni

# Az UNO API célja

- ▼ A makrók tudják írni és olvasni az új funkciókat
  - ▼ Hasznos: még az előtt lehet tesztelni, hogy lenne UI
- ▼ Az UNO szűrők is ezt használják
- ▼ A tesztesetek megírását is könnyíti
- ▼ Forrás az [sw/source/core/unocore/](#) alatt

# Hogyan játszunk a makrókkal / UNO API-val

- ▼ Környezetérzékeny térközök bekapcsolása egy paragrafusra:

```
enum = ThisComponent.Text.createEnumeration  
para = enum.NextElement  
xray para.ParaContextMargin  
para.ParaContextMargin = True
```

- ▼ Más fejléc/lábléc bekapcsolása az első oldalakon:

```
oDefault = ThisComponent.StyleFamilies.PageStyles.Default  
oDefault.HeaderIsOn = True  
oDefault.HeaderIsShared = False  
oDefault.FirstIsShared = False
```

- ▼ Megjegyzés csatolása szövegtartományhoz:

```
oTextField =  
oDoc.createInstance("com.sun.star.text.TextField.Annotation")  
oTextField.TextRange.String = "Tartalom"  
oDoc.Text.insertTextContent(oCurs, oTextField, True)
```

# Layout

Jelenítsük meg azt, ami eddig rejtve volt

# Layout

- ▼ A “View” az MVC-ből
- ▼ A Writer esetében ennek is saját dokumentum modellje van:
  - ▼ A dokumentum modell építőkövei a paragrafusok
  - ▼ Itt mindent keretektől építünk:
    - ▼ Egy megnyitott dokumentum ↔ **SwRootFrm**
    - ▼ Egy oldal ↔ **SwPageFrm**
    - ▼ Egy paragrafus ↔ **SwTxtFrm**
- ▼ Egy modell elem ↔ több layout elem
  - ▼ Fejléc minden oldalon
  - ▼ A paragrafusok több oldalra kiterjedhetnek
- ▼ A SwClient / SwModify mechanizmus kapcsolja össze
- ▼ Kiírás: u. az, mint a dokumentum modell, csak Shift-F12

# Szűrők

Próbáljuk meg túlélni az újraindítást

# Szűrők

- ▼ Export szűrők: az **SwDoc** példányt mentik egy stream-re
- ▼ Import szűrők: egy stream alapján felépítik az **SwDoc** példányt újra
- ▼ ~Minden szűrő “alien”
  - ▼ Ismert, hogy információt vesztenek a konvertálás során
  - ▼ Kivétel: ODF szűrők (a.k.a. “own”)
- ▼ Típusok:
  - ▼ UNO: DOCX import, RTF import
  - ▼ Beépített – belső sw API-t használja: DOC import, Word export
  - ▼ Kever: ODF (főleg UNO, de pár rész belső)
- ▼ Kód: **sw/source/filter/** , **writerfilter/** , **xmloff/**

## Felhasználói felület

Hogy a felhasználók is felfedezhessék a munkánkat

# Régi / új felhasználói felület

- ▼ Régi: ne használjunk egeret az UI elkészítéséhez
  - ▼ Tipikusan egy **.src** , **.hrc** és egy **.cxx** file minden ablakhoz
  - ▼ Minden elem rögzített pozícióval / mérettel rendelkezik:

```
CheckBox CB_SHARED_FIRST
{
    HelpID = "svx:CheckBox:RID_SVXPAGE_HEADER:CB_SHARED_FIRST";
    Pos = MAP_APPFONT ( 12 , 46 ) ;
    Size = MAP_APPFONT ( 152 , 10 ) ;
    Text [ en-US ] = "Same content on first page" ;
};
```

- ▼ Új elem hozzáadása az ablak közepén
  - ▼ És az összes többi vezérlőt le kell mozgatni kézzel
- ▼ Új: Glade-alapú
  - ▼ Caolán külön **előadást tartott** csak erről (LOCon 2012)

# UI ↔ dokumentum modell interakció

- ▼ Az UI gyakran közös, nem lehet alkalmazás-specifikus
- ▼ Feltalált megoldás: **SfxItemSet**
- ▼ Jelölőnégyzet → **SfxItemSet** :

```
aSet.Put(SfxBoolItem(nWSharedFirst,aCntSharedFirstBox.IsChecked()));
```

- ▼ **SfxItemSet** → dokumentum modell (**SwPageDesc**):

```
rPageDesc.ChgFirstShare(((const SfxBoolItem&)
rHeaderSet.Get(SID_ATTR_PAGE_SHARED_FIRST)).GetValue()));
```

- ▼ Hasonló a másik irányban
- ▼ A közös kód az **svx/source/dialog/** , **cui/** alatt
- ▼ Writer-specifikus kód az **sw/source/ui/**

# Tesztesetek

Miért ismételnénk meg régi hibákat, ha választhatunk újak közül is?

# Tesztesetek

- ▼ Teszt típusok a LibreOffice-ban:
  - ▼ Unitcheck: **minden** részleges modul fordítás végén
    - ▼ Szuper-gyorsnak kell lennie (CVE tesztek, például)
  - ▼ Slowcheck: minden teljes fordítás végén
    - ▼ A szűrők tesztelése tipikusan ide tartozik
  - ▼ Subsequentcheck: régi Java tesztek
    - ▼ Lassan migrálásra kerülnek natív kódra
- ▼ Mit tesztelünk:
  - ▼ Dokumentum modell (UNO API)
  - ▼ Layout (XML kiírás + XPath kifejezés kiértékelése):

```
CPPUNIT_ASSERT_EQUAL(OUString("Első fejléc"),  
    parseDump("/root/page[1]/header/txt/text()));
```

# Import / export tesztek

- ▼ Minimális reprodukáló dokumentum készítése
- ▼ Import vagy import → export → import
  - ▼ Attól függően, hogy mit akarunk tesztelni
- ▼ Az elkészülő dokumentum modell vizsgálata UNO API-val:

```
CPPUNIT_ASSERT_EQUAL(6, getLength());
```

- ▼ Futtassuk le
- ▼ Vonjuk vissza a javítást / funkciót
- ▼ Bizonyosodjunk meg róla, hogy nem fut le
- ▼ Dobjuk el a visszavonást, vonjuk össze a két commitot
- ▼ Kód: [sw/qa/](#)

# Dokumentáció

Amikor a nyilvánvaló nem annyira nyilvánvaló

# Felhasználói súgó

- ▼ Tipikusan nem érdekli a fejlesztőket
- ▼ Mindig ez jön fel, ha a felhasználó F1-et nyom
- ▼ Hova illesszük be az új tartalmat?
  - ▼ Használjuk a felhasználói felületet
  - ▼ Keressük meg azt a helyet ami elé/mögé/közbe írnánk
  - ▼ Futtassuk a **git grep** parancsot a **helpcontent2/** alatt
  - ▼ Adjunk új tartalmat a megtalált XML file-okhoz
- ▼ Problémák
  - ▼ Nincs **linkoo** , így **make dev-install** szükséges próba előtt
  - ▼ Egyedi ID minden paragrafushoz
    - ▼ Nem probléma, csak file-on belül kell, hogy egyedi legyen
    - ▼ Másoljunk le egyet, és pl. kvadratikusan próbáljuk :-)

# Fejlesztői dokumentáció

- ▼ Eszköz: **doxygen**
- ▼ Dokumentáljunk minden osztályt – legalább egysoros leírás, hogy mégis mit csinál
  - ▼ Preferáltan minden publikus metódust
  - ▼ Eszköz: **bin/find-undocumented-classes**
- ▼ A régi kód tele van német megjegyzésekkel
  - ▼ Ha már amúgy is rászántad az időt, hogy megértsd...
  - ▼ ... akkor legyen még egy perced, hogy lefordítsd
  - ▼ Eszköz: **bin/find-german-comments**

# Specifikáció

Ezzel kellett volna indítani... vagy mégse?

# Specifikáció

- ▼ Mindent ODF-be mentünk
  - ▼ Az ODF egy nyílt szabvány, csak több évente frissítik
  - ▼ 22-es csapdája
- ▼ Megoldás: először implementáljuk, legyen a neve “ODF Extended”
  - ▼ Majd javaslat küldése az OASIS felé
  - ▼ Ideális esetben: elfogadásig kiterjesztési névterek használata

# ODF JIRA hibajegy példa

- ▼ Rationale: mi a motiváció
- ▼ Requested changes to the standard
  - ▼ Változások a szabvány szövegében
  - ▼ A séma változásai
- ▼ Impacts
  - ▼ Visszafele kompatibilitás kérdései
  - ▼ Egyszerű, ha az új elem opcionális



Köszönöm a figyelmet ...

- ▼ ... kérdések?
- ▼ Diák: <http://vmiklos.hu/odp/>



All text and image content in this document is licensed under the [Creative Commons Attribution-Share Alike 3.0 License](#) (unless otherwise specified). "LibreOffice" and "The Document Foundation" are registered trademarks. Their respective logos and icons are subject to international copyright laws. The use of these therefore is subject to the [trademark policy](#).