



OpenOffice.org im Backoffice

OOo im Serverbetrieb

webbasierter PDF-Generator

Martin Winkler
mw@agami.at



Voraussetzungen für Serverbetrieb

- Idealerweise *nix System
- Ausreichend RAM
- Jede CPU mit eigenem OOo-Server (Skalierbarkeit)



Auswahl der Sprachumgebung

- C++
- Java
- Python
- (Perl)



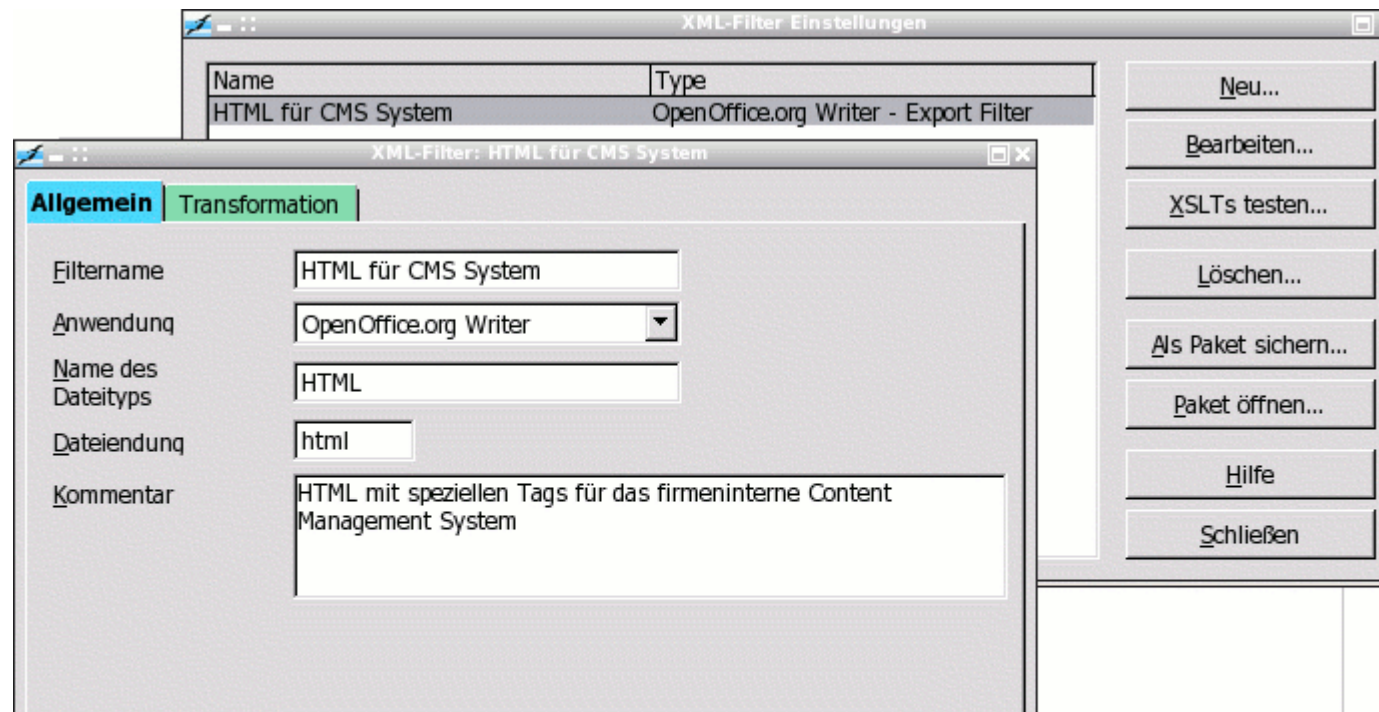
Tipps & Tricks

- XML-Format *nicht* auf Größe optimieren
- > unzip meinedatei.sxw

```
Archive: beispieldokument.sxw
  Length      Date      Time      Name
  ----
      30      03-29-05  21:19    mimetype
    12413     03-29-05  21:19    ObjBFFFE991/content.xml
      786     03-29-05  21:19    ObjBFFFE991/styles.xml
     9473     03-29-05  21:19
Pictures/10000000000001070000010755C7BE12.jpg
    64542     03-29-05  21:19
Pictures/10000000000003B100000277BB4D872B.jpg
    11547     03-29-05  21:19    content.xml
     6227     03-29-05  21:19    styles.xml
     1215     03-29-05  21:19    meta.xml
     7634     03-29-05  21:19    settings.xml
     1045     03-29-05  21:19    META-INF/manifest.xml
-----
    40897                                8 files
```



Nutzung der XML/XSLT Komponenten





Kommunikation mit dem Server

- OOo als Hintergrundprozeß
- UNO – Univeral Network Objects



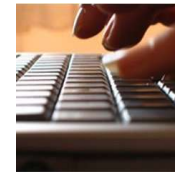
Performance

- Testergebnisse
 - 30 Dokumente mit jeweils 20 Seiten: 60sek
 - Durchsatz: 10 Seiten pro Sekunde
 - 2GHz Pentium4, 512MB RAM



Anwendungsbeispiel

- LIVE



Ideale Einsatzgebiete

- Dokumente mit Platzhaltern, Ausfüllen im Browser, PDF Download
- OOo-Java Applet im Browser