

Erweiterungen für LibreOffice erstellen

Andreas Mantke

31. August 2018

Erweiterungen für LibreOffice erstellen

© Andreas Mantke

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License. Dieses Werk ist lizenziert unter der Creative Commons Attribution-NonCommercial 4.0 International License.

<https://creativecommons.org/licenses/by-nc/4.0/>

<https://creativecommons.org/licenses/by-nc/4.0/legalcode>

Inhaltsverzeichnis

1	Arten von Erweiterungen	9
2	Der Container OXT und sein Inhalt	11
2.1	Die Datei description.xml	11
2.2	Meta-Informationen	16
2.3	Erstellen der Extension-Datei	17
2.4	Erste Testinstallation der Extension	18
2.5	Weitere Einträge in der Datei description.xml	19
3	Auslieferung von Inhalten mit Extensions	25
3.1	Gallery-Extension	25
3.1.1	Neues Gallery-Thema erstellen	25
3.1.2	Die Dateien des neuen Gallery-Themas	27
3.1.3	Bauen der Gallery-Extension	28
3.2	Template-Extension	29
3.3	AutoText-Extension	33
3.4	Autokorrektur-Extension	36
3.5	Farbpaletten-Extension	41
4	Ein neues Icon-Set mit einer Extension ausliefern	45
4.1	Bauen der Icon-Set-Extension	45

Abbildungsverzeichnis

2.1	Fehlermeldung wegen fehlerhafter Namespace-Deklaration	12
2.2	Fehlermeldung wegen fehlender Datei manifest.xml beim Entfernen der Extension	16
2.3	Der LibreOffice Extension Manager nach Installation der neuen Extension	18
3.1	Die LibreOffice Gallery	26
3.2	Das neue Gallery-Thema mit einigen Grafiken	27
3.3	Der Dialog Vorlagen verwalten mit neuer Kategorie education	31
3.4	Anzeige der verfügbaren Textbausteine	34
3.5	Anzeige Autokorrektur-Extension im Extension-Manager nach Installa- tion	39
3.6	Der Dialog Fläche	42

Vorwort

Für das Entwickeln von LibreOffice finden sich Informationen auf verschiedenen Ressourcen. Zum Teil muss ersatzweise auf - nicht immer noch aktuelle - Informationen zu OpenOffice.org zurückgegriffen werden. Das Ziel dieses Dokumentes ist es, möglichst detaillierte Informationen zur eigenständigen Entwicklung von neuen LibreOffice Extensions zu geben. Die Darstellung soll dabei anhand von Beispielen erfolgen. Diese werden die unterschiedlichen Bereiche beleuchten, mit denen sich weitere oder besondere Inhalte oder neue bzw. erweiterte Funktionen zu LibreOffice hinzugefügt werden können. Es geht also sowohl um beispielsweise das Hinzufügen von eigenen Themen zur sogenannten Gallery wie auch um weitere Funktionalität unter Verwenden von Programmiersprachen, wie z.B. Python. Für die letztere Programmiersprache bringt LibreOffice bereits eine Umgebung von Python 3 mit.

Das Dokument ist eine Arbeit im Fluss und wird nach und nach um weitere Kapitel ergänzt werden. Es ist aktuell nur in deutscher Sprache verfügbar. Eine Lokalisation in anderen Sprachen ist aber für die Zukunft eine Option.

1 Arten von Erweiterungen

Für LibreOffice können Sie Extensions (Erweiterungen) erstellen, die mit verschiedenen Programmiersprachen arbeiten. Es sind die folgenden Sprachen verfügbar:

- C++
- Python
- Java
- JavaScript
- LibreOffice Basic

Außerdem können Sie mit Extensions auch Konfigurationsdateien für LibreOffice für Ihre Benutzer beispielsweise im Rahmen einer automatischen Installation ausrollen. Mit Extensions können Sie auch eine neue Gallery (Zusammenstellung von Grafiken), weitere Paletten, Autotexte oder Assistenten für LibreOffice bereit stellen.

Alle Erweiterungen von LibreOffice werden in einer komprimierten Container-Datei mit der Dateiendung `.oxt` gepackt. Sie können, sofern Sie unter einer freien Software-Lizenz stehen, über die Seite <https://extensions.libreoffice.org> veröffentlicht und geteilt bzw. zum Download angeboten werden.

2 Der Container OXT und sein Inhalt

Innerhalb des komprimierten Datei-Containers OXT befinden sich mehrere Dateien und es ist regelmäßig auch eine spezielle Struktur angelegt. Diese Struktur und der Inhalt der einzelnen Dateien des Containers sollen nun anhand eines konkreten Beispiels erläutert werden. Diese Beispielextension soll den Namen `myfirstextension.oxt` führen. Wir erstellen dazu zunächst ein Verzeichnis mit dem Namen `myfirstextension`. In diesem erzeugen wir dann die Textdatei `description.xml`, die wir im folgenden Abschnitt bearbeiten.

2.1 Die Datei `description.xml`

Wir öffnen nun die zuvor erstellte Datei `description.xml` mit einem Texteditor. Diese Datei wird die wesentlichen Verwaltungsinformationen der Extension enthalten. Sie startet mit einer XML-Deklaration einschließlich Encoding:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Danach folgt das `Description`-Tag, das mit einer Definition versehen ist und als Root alle beschreibenden Elemente einschließt:

```
<description
  xmlns="http://openoffice.org/extensions/description/2006"
  xmlns:xlink="http://www.w3.org/1999/xlink">
</description>
```

Bereits mit diesem rudimentären Inhalt könnte die Extension als komprimierter OXT-Container (Zip-Container) vom LibreOffice Extension-Manager geladen werden, allerdings fehlen für den Anwender noch die wesentlichen Grundinformationen und die Extension hat auch noch keine Funktionalität.

Sofern sich bei der Vorgabe der Deklaration des Namespace im `<description>`-Tag ein Fehler (z.B. ein Tippfehler) eingeschlichen hat, erhalten Sie eventuell beim Versuch der Installation der Extension eine Fehlermeldung wie im Bildschirmfoto unten.

2 Der Container OXT und sein Inhalt



Abbildung 2.1: Fehlermeldung wegen fehlerhafter Namespace-Deklaration

Identifizier

Als erste Ergänzung fügen wir in das Description-Tag einen sogenannten Identifier ein. Dies ist eine eindeutige Bezeichnung der Extension für den LibreOffice Extensionmanager. Es ist möglich, dass zwei inhaltlich bzw. auf Dateiebene identische Extensions vom Extensionmanager geladen werden, sofern sie unterschiedliche Identifier haben, nicht aber anders herum. Deshalb ist es wichtig, dass sie einen Identifier wählen, auf den nur Sie Zugriff haben bzw. der regelmäßig nur von Ihnen in dieser Form gewählt werden wird.

Der Identifier spielt dabei auch bei einem eventuell vorgesehenen Online-Update der Extension später eine wesentliche Rolle. Für unser Beispiel verwenden wir:

```
<identifier value="org.mycompany.myfirstextension" />
```

Version

Als nächstes fügen wir die Angabe der Versionsnummer (Release-Nummer) der Extension in das Description-Tag als neues Tag ein:

```
<version value="0.1" />
```

Als Werte (Value) kommen hier positive Zahlen in Betracht, die eine Versionsnummer der Extension darstellen können. Die Versions-Nummer kann aus lediglich einer positiven Ziffer (größer Null) oder einer Kolonne von mehreren durch Punkt getrennten Ziffern bestehen, z.B. 0.1.3. Das Benennungsschema sollte dabei den üblichen Geflogenheiten folgen. Die erste Ziffer bezeichnet insoweit Major-(Haupt-)Releases (mit eventueller Inkompatibilität zu bisherigen Haupt-Releases). Die zweite Ziffer steht für Änderungen im Sinne von Minor-Releases (mit kompatiblen Änderungen), während die dritte Ziffer Micro-Releases anzeigt (mit beispielsweise Fehlerbereinigungen).

Plattform

2.1 Die Datei *description.xml*

Eine LibreOffice Extension kann sowohl für alle Plattformen von LibreOffice lauffähig sein, als auch lediglich für einzelne. Dies wird über das Platform-Tag gesteuert. Wir gehen davon aus, dass die Extension für alle Plattformen geeignet ist und fügen daher folgendes Zeile ein:

```
<platform value="all" />
```

Der Wert ("value") kann entweder eine oder mehrere Plattformen mit Komma getrennt enthalten. Beispielsweise in folgender Form:

```
<platform value="windows_x86, linux_x86_64, linux_x86" />
```

Die Extension kann nur installiert werden, wenn eine der angegebenen Plattformen dem laufenden Betriebssystem entspricht. Falls als Wert keine Plattform angegeben wird, kann die Extension unter keinem laufenden Betriebssystem installiert werden.

```
<platform value=" " />
```

Der Extensionmanager von LibreOffice kann hier keine Plattform (Betriebssystem) ausfindig machen, auf dem die Extension lauffähig ist, und bricht daher die Installation ab.

Die folgende Tabelle listet alle möglichen Plattformen auf, die als Wert verwendet werden können, unabhängig davon, ob sie von LibreOffice unterstützt werden.

Platform Token (Plattform-Schlüssel)

Token (Schlüssel)	Beschreibung
all	Umfasst alle Plattformen
freebsd_x86	FreeBSD Betriebssystem auf X86-Plattform
freebsd_x86_64	FreeBSD Betriebssystem auf 64bit-Plattform
linux_arm_eabi	(Aktuell nicht unterstützt) Linux Betriebssystem auf einer ARM-CPU unter Verwenden von EABI"
linux_arm_oabi	(Aktuell nicht unterstützt) Linux Betriebssystem auf einer ARM-CPU unter Verwenden von ÖABI"
linux_ia64	Linux Betriebssystem auf einer ia64 CPU
linux_mips_eb	(Aktuell nicht unterstützt) Linux Betriebssystem auf einer MIPS CPU unter verwenden von 'EB' ABI
linux_mips_el	(Aktuell nicht unterstützt) Linux Betriebssystem auf einer MIPS CPU unter verwenden von 'EL' ABI.
linux_powerpc	Linux Betriebssystem auf einer POWERPC CPU.
linux_powerpc64	(Aktuell nicht unterstützt) Linux Betriebssystem auf einer POWERPC 64bit CPU.
linux_s390	Linux Betriebssystem auf einer s390 CPU.
linux_s390x	(Aktuell nicht unterstützt) Linux Betriebssystem auf einer s390x CPU.
linux_sparc	Linux Betriebssystem auf einer SPARC CPU.
linux_x86	Linux Betriebssystem auf einer x86 CPU.
linux_x86_64	Linux Betriebssystem auf einer x86 64 bit CPU
macosx_powerpc	Mac X Betriebssystem auf einer POWERPC CPU
macosx_x86	Mac X Betriebssystem auf einer x86 CPU
macosx_x86_64	Mac X Betriebssystem auf einer x86-64 CPU
os2_x86	OS/2 Betriebssystem auf einer x86 CPU
solaris_sparc	Solaris Betriebssystem auf einer SPARC CPU
solaris_x86	Solaris Betriebssystem auf einer x86 CPU
windows_x86	Windows operating system running on a x86 CPU

Rückwärtskompatibilität von Plattformen

Teilweise wird ein LibreOffice-32-bit-Build auch auf 64-bit-Linux-Plattformen ausgeführt. In einem solchen Fall vergleicht der Extensionmanager die Plattform der Extension mit derjenigen, für die LibreOffice erstellt wurde. Die folgende Tabelle enthält einige Beispiele für Kombinationen und die jeweiligen Ergebnisse.

Extension Plattform	LibreOffice erstellt für Plattform	Plattform, auf der LibreOffice läuft	Resultat
Linux_x86	Linux_x86	Linux_x86	OK
Linux_x86	Linux_x86	Linux_x86_64	OK
Linux_x86	Linux_x86_64	Linux_x86_64	Scheitert
Linux_x86_64	Linux_x86	Linux_x86_64	Scheitert

Anzeigename der Extension

Als nächstes fügen wir dem Description-Tag einen Namen hinzu, der im Extensionmanager angezeigt wird. Dieser Name kann auch für die einzelnen Lokalisierungen vorgegeben werden. Der umschließende Tag lautet `<display-name>`. Der angezeigte Name wird dann in einem oder mehreren `<name>`-Tags vorgegeben.

Sofern der `<name>`-Tag einen Zusatz bzgl. der Sprache (`lang`) enthält, wird dieser Text für die entsprechende Lokalisierung angezeigt. Ohne einen solchen Zusatz bzw. für andere Lokalisierungen ohne eigenen `<name>`-Tag wird der Standard bzw. die englische Version angezeigt. Für unsere Beispiel-Extension geben wir dem Description-Tag folgendes hinzu:

```
<display-name>
  <name lang="de">Meine erste LibreOffice Extension</name>
  <name lang="en">My first LibreOffice extension</name>
</display-name>
```

Autor

Nun fügen wir noch den Autor über das `<publisher>`-Tag der Description hinzu. In unserem Beispiel lautet der Tag für mich selbst folgendermaßen:

```
<publisher>
  <name xlink:href="http://amantke.de/blog">Andreas Mantke</name>
</publisher>
```

Der `<publisher>`-Tag muss zumindest einen `<name>`-Tag haben. Letzterer Tag kann wieder für unterschiedliche Lokalisierungen über einen `lang`-Zusatz vorgegeben werden. Der `<name>`-Tag selbst enthält eine URL zusammen mit dem ggfs. lokalisierten Namen des Autors.

Die Datei description.xml sieht nun wie folgt aus:

```
<?xml version="1.0" encoding="UTF-8"?>
<description
```

2 Der Container OXT und sein Inhalt

```
xmlns="http://openoffice.org/extensions/description/2006"
xmlns:xlink="http://www.w3.org/1999/xlink">

<identifier value="org.mycompany.myfirstextension" />
<version value="0.1" />
<platform value="all" />

<display-name>
  <name lang="de">
    Meine erste LibreOffice Extension</name>
  <name lang="en">
    My first LibreOffice extension</name>
</display-name>
<publisher>
  <name xlink:href="http://amantke.de/blog">
    Andreas Mantke</name>
</publisher>
</description>
```

2.2 Meta-Informationen

Eine LibreOffice Extension benötigt zumindest eine Datei manifest.xml, die weitere Verwaltungsinformationen enthalten kann. Ohne diese Datei ließe sich die Extension zwar installieren, aber eine Deinstallation würde scheitern mit einer Fehlermeldung wie im folgenden Bildschirmfoto.



Abbildung 2.2: Fehlermeldung wegen fehlender Datei manifest.xml beim Entfernen der Extension

Wir legen dazu ein Unterverzeichnis „META-INF“ und in diesem eine neue Datei manifest.xml an. Die Datei bekommt zunächst nur folgenden Inhalt:

```
<?xml version="1.0" encoding="UTF-8"?>
<manifest:manifest
  xmlns:manifest="http://openoffice.org/2001/manifest">
</manifest:manifest>
```


Zu den möglichen weiteren Inhalten dieser Datei wird an späterer Stelle in diesem Dokument Näheres erläutert.

2.3 Erstellen der Extension-Datei

Mit diesen ersten Versionen der Dateien `description.xml` und `manifest.xml` können wir nun eine Testversion der Extension-Datei erstellen und versuchen, die Extension noch ohne Funktionalität mit dem Extensionmanager zu installieren (und wieder zu deinstallieren) Bei der Extension-Datei (*.`oxt`) handelt es sich um einen komprimierten Container. Um diesen zu erstellen, können Sie dem für die Test-Extension erstellten folgendes Python-Buildskript hinzufügen. Dieses Skript hat Björn Michaelsen erstellt. Es steht unter der Mozilla Public License (MPL) und ist beispielsweise hier zu finden: <https://gerrit.libreoffice.org/gitweb?p=sdk-examples.git;a=blob;f=BundesGit/build;h=4e29b2addb081599100c5e8b48c384bbfff427ca;hb=HEAD>

Das Skript setzt voraus, dass im Verzeichnis eine weitere Textdatei mit dem Namen `extensionname.txt` vorhanden ist, die den Namen der Extension enthält, hier den Text `myfirstextension.oxt`

Das Build-Skript mit dem Namen `build` sieht folgendermaßen aus:

```
#!/usr/bin/env python
# This Source Code Form is subject to the terms of the
# Mozilla Public License, v. 2.0. If a copy of the MPL was
# not distributed with this file, You can obtain one at
# http://mozilla.org/MPL/2.0/.

from zipfile import ZipFile
import os, os.path, sys
scriptdir = os.path.dirname(os.path.abspath(sys.argv[0]))
extensionname = open(
    os.path.join(scriptdir,
        'extensionname.txt')).readlines()[0].rstrip('\n')
with ZipFile(extensionname, 'w') as tuesdayzip:
    os.chdir(scriptdir)
    for root, dirs, files in os.walk('.'):
        for name in files:
            if not name == extensionname:
                tuesdayzip.write(os.path.join(root, name))
```

Da das Skript in der Programm-Sprache Python geschrieben ist, müssen Sie darauf achten, dass die Einrückungen korrekt übernommen werden. Das Skript selbst rufen Sie später einfach mit

```
python ./build
```

auf. Die Extension-Datei wird dadurch in dem Verzeichnis der Test-Extension erstellt.

2.4 Erste Testinstallation der Extension

Die gerade erstellte OXT-Datei installieren wir nun testweise mit dem Extensionmanager von LibreOffice. Wir starten dazu LibreOffice und wählen im Menü Extras den Eintrag "Extension Manager". In der sich öffnenden Dialogbox aktivieren wir die Schaltfläche "Hinzufügen" und navigieren in dem nun geöffneten Datei-Dialog zu der gerade erstellten OXT-Datei und bestätigen die Auswahl über die Schaltfläche "Öffnen". Die neue Extension sollte anschließend im Dialogfenster gelistet sein.

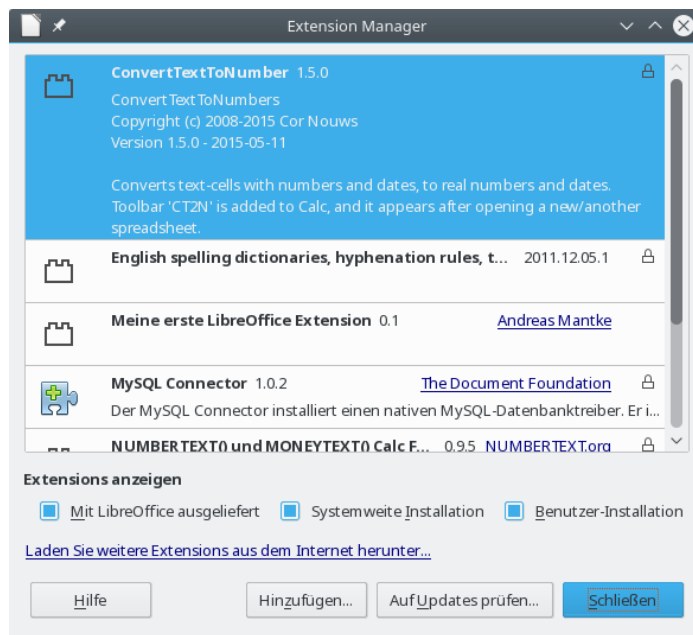


Abbildung 2.3: Der LibreOffice Extension Manager nach Installation der neuen Extension

Auf der rechten Seite des Eintrags für die neue Extension befindet sich der Name des Autors verbunden mit einem Link zu der URL, die in dem `<name>`-Tag innerhalb des `<publisher>`-Tags angegeben wurde. Dieser kann mit einem Mausklick im Standard-Browser aufgerufen werden.

Markieren Sie nun den neuen Eintrag im LibreOffice Extension Manager und deinstallieren Sie die Extension wieder über die nun angezeigte Schaltfläche "Entfernen". Die Extension sollte sich damit nach Bestätigen des folgenden Benutzerdialogs fehlerfrei entfernen lassen. Sofern dies nicht der Fall ist, könnte dies beispielsweise an einer fehlenden Datei `manifest.xml` (siehe dazu oben).

2.5 Weitere Einträge in der Datei description.xml

Auf der Abbildung im letzten Abschnitt ist zu sehen, dass für die Test-Extension im LibreOffice Extension Manager keine Grafik bzw. ein Logo angezeigt wird, wie beispielsweise für den MySQL Connector. Um ein solches Logo einzubinden, existiert der `<icon>`-Tag.

Erstellen Sie nun ein neues Unterverzeichnis „images“ und anschließend eine neue Grafik bzw. ein neues Logo im Dateiformat PNG und der Dimension 42 mal 42 Pixel. Die Grafikdatei speichern Sie dann unter dem Namen `"extensionicon_42.png"` im neuen Unterverzeichnis. Um das Logo einzubinden, fügen Sie dem `<description>`-Tag folgenden Eintrag hinzu:

```
<icon>
  <default xlink:href="images/extensionicon_42.png" />
</icon>
```

In dem Tag hinter „xlink:href“ wird die relative URL zu der Grafikdatei angegeben. Neben dem Link zu der Default-Grafik ist es möglich, parallel eine alternative Grafik mit hohem Kontrast, mit einem weiteren Tag anzugeben:

```
<high-contrast xlink:href="(...)" />
```

Minimale LibreOffice Version

Es ist möglich, für die Extension eine minimale Version von LibreOffice anzugeben. Dies geschieht über den `<dependencies>`-Tag. Dieser Tag braucht eine Namespace-Deklaration, die hinter „xmlns:lo“ erfolgt. Innerhalb der `<dependencies>`-Tag wird dann mit einem weiteren Tag die minimale LibreOffice-Version, für das Beispiel Version 4.2.

```
<dependencies xmlns:lo=
  "http://libreoffice.org/extensions/description/2011">

  <lo:LibreOffice-minimal-version name="LibreOffice 4.2"
    value="4.2"/>

</dependencies>
```

Die Vorgabe einer auch maximalen Version von LibreOffice, wie dies früher für OpenOffice.org möglich war, steht nicht mehr zur Verfügung. Der Tag für die Angabe der minimalen LibreOffice-Version erfordert neben der Übergabe des Wertes („value“) mit der Nummer des Releases (nur zweistellig, nicht ein etwaiges Micro-Release) auch die Vorgabe des Namens als Zeichenkette. Für die Anzeige der Meldung über die minimal erforderliche LibreOffice-Version im Extension Manager wird allerdings nur auf das angegebene Value zurückgegriffen.

Beschreibung der Extension

Der Extension kann eine Erläuterung Ihrer Funktionsweise über den `<extension-description>`-Tag hinzugefügt werden, wobei dieser mehrere Lokalisierungen dieser Beschreibung aufzunehmen.

Hierzu legen wir zunächst ein weiteres Unterverzeichnis „description“ und darin zwei neue -zunächst leere Textdateien `description_de.txt` und `description_en.txt` an. Diese können später mit Informationen in deutscher und englischer Sprache befüllt werden. Wir binden diese beiden Beschreibungsdateien über folgende Ergänzung des `<description>`-Tag ein:

```
<extension-description>
  <src xlink:href="description/description_en.txt"
    lang="en" />
  <src xlink:href="description/description_de.txt"
    lang="de" />
</extension-description>
```

Der in ihnen abgelegte Text wird nach der Installation im Extension Manager als Erläuterung angezeigt. Er sollte daher möglichst kurz und präzise formuliert werden und gegebenenfalls für ausführlichere Informationen auf eine externe Ressource (z.B. eine Webseite oder ein Wiki) verweisen.

Die Bezeichner für die Lokalisierung richten sich nach RFC 3066. Es sind daher sowohl Lokalisierungsbezeichner in der zuvor benutzten Form als auch spezifischere zulässig wie beispielsweise:

```
<src (...) lang="en-GB" />
<src (...) lang="en-US" />
```

Release Notes

Es ist zwar möglich, in der Datei `description.xml` ein Tag für Release-Notes vorzugeben, allerdings hat dies für den Anwender keinen praktischen Nutzen, da ihm die Verbindung zu einer in dem `<release-notes>`-Tag eingestellten Verbindung zu einer URL im Extension Manager nicht angezeigt wird. Sofern für den Anwender wichtige Informationen zur Funktionsweise der Extension und zu Änderungen des oder der Release angezeigt werden sollten, eignet sich daher allein der zuvor beschriebene `<description>`-Tag und eine darin verlinkte externe Ressource am besten.

Registration / Lizenz

Das Registration-Element ermöglicht im Moment allein die Vorgabe der Lizenz der Extension, wobei auch hier wieder die Möglichkeit der parallelen Übermittlung von unterschiedlichen Lokalisierungen besteht.

2.5 Weitere Einträge in der Datei `description.xml`

Das `<registration>`-Element enthält ein `<simple-license>`-Element, welches wieder für jede Lokalisierung ein `<license-text>`-Element enthält. Das `<simple-license>`-Element enthält ein notwendiges Attribut „accept-by“. Optional sind zwei weitere Attribute in diesem Element möglich: „suppress-on-update“ und „suppress-if-required“.

Attribut	Beschreibung
accept-by	Erforderlich. Der Wert kann entweder „user“ oder „admin“ sein. „user“ bedeutet, dass jeder Benutzer der Lizenz zustimmen hat. Dies heißt dann, dass die Extension nur vom Benutzer und nicht vom Administrator als systemweite (shared) Extension installiert werden kann. Sofern der Wert „admin“ gesetzt ist, kann die Extension auch als shared (geteilt / systemweit) installiert werden. In diesem Fall muss lediglich die Person, die die Extension installiert, der Lizenz zustimmen. Die einzelnen Benutzer werden nicht gefragt, ob sie die Lizenz akzeptieren. Sie können die Extension einfach benutzen.
suppress-on-update	Optional. Falls das Attribut nicht gesetzt ist, wird als Wert „false“ angenommen. Sofern das Attribut mit dem Wert „true“ vorgegeben ist, wird bei einem Update der Extension (gleiche ID, aber eventuell andere Version) der Lizenzdialog nicht erneut angezeigt.
suppress-if-required	Optional. Falls das Attribut nicht gesetzt ist, wird als Wert „false“ angenommen. Der Wert „true“ weist darauf hin, dass der Lizenzdialog während der Installation nicht angezeigt wird, sofern dies gewünscht wird. Dies kann dann durch den Schalter <code>--suppress-license</code> von <code>unopkg</code> vorgegeben werden und wird implizit für mitgelieferte (gebündelte) Extensions vorausgesetzt.

Legen Sie nun ein weiteres Unterverzeichnis „registration“ und darin die Textdateien `license_de.txt` und `license_en.txt` an. Es reicht zunächst, wenn die Dateien existieren, auch wenn sie aktuell noch keinen Inhalt haben. Ihr späterer Inhalt wird, sofern nicht in der Einstellung unterdrückt, dem Benutzer im Installationsdialog angezeigt. In dem Dialog wird dann abgefragt, ob er die Lizenz akzeptiert.

```
<registration>
  <simple-license accept-by="admin">
    <license-text xlink:href="registration/license_de.txt"
      lang="de" />
    <license-text xlink:href="registration/license_en.txt"
      lang="en" />
  </simple-license>
</registration>
```

2 Der Container OXT und sein Inhalt

```
</simple-license>
</registration>
```

Die Datei description.xml hat mit den weiteren vorgenommenen Ergänzungen nun folgenden Inhalt:

```
<?xml version="1.0" encoding="UTF-8"?>
<description
  xmlns="http://openoffice.org/extensions/description/2006"
  xmlns:xlink="http://www.w3.org/1999/xlink">

  <identifier value="org.mycompany.myfirstextension" />
  <version value="0.1" />
  <platform value="all" />
  <display-name>
    <name lang="de">
      Meine erste LibreOffice Extension</name>
    <name lang="en">
      My first LibreOffice extension</name>
  </display-name>
  <publisher>
    <name xlink:href="http://amantke.de/blog">
      Andreas Mantke</name>
  </publisher>
  <icon>
    <default xlink:href="images/extensionicon_42.png" />
  </icon>
  <dependencies xmlns:lo=
    "http://libreoffice.org/extensions/description/2011">
    <lo:LibreOffice-minimal-version name="LibreOffice 4.2"
      value="4.2"/>
  </dependencies>
  <extension-description>
    <src xlink:href="description/description_en.txt"
      lang="en" />
    <src xlink:href="description/description_de.txt"
      lang="de" />
  </extension-description>
  <registration>
    <simple-license accept-by="admin">
      <license-text
        xlink:href="registration/license_de.txt"
        lang="de" />
      <license-text xlink:href="registration/license_en.txt"
        lang="en" />
    </simple-license>
```

2.5 Weitere Einträge in der Datei *description.xml*

```
</registration>  
</description>
```

Eine entsprechende Basisvorlage für eine LibreOffice Extension finden Sie im Github-Repository unter folgender Adresse: https://github.com/andreama/extensionbook/tree/master/extensiontemplates/extension_basic_design/example01.

3 Auslieferung von Inhalten mit Extensions

Mit einer LibreOffice Extension lassen sich verschiedene Arten von Inhalten dem Standard-Programm hinzufügen. So ist es beispielsweise möglich, der Gallery, einem Verwaltungsmodul für Grafiken, Zeichnungen, Musik und andere Multimediadateien, neue Bereiche und Dateien hinzuzufügen. Wie Sie eine solche Extension erstellen, soll im folgenden Abschnitt erläutert werden.

3.1 Gallery-Extension

Bevor wir eine Gallery-Extension erstellen können, müssen wir zunächst mit LibreOffice ein entsprechendes neues Gallery-Thema erstellen, das wir dann mit neuen Inhalten (z.B. Grafiken) befüllen und die entstandenen Dateien exportieren.

3.1.1 Neues Gallery-Thema erstellen

Beginnen wir also mit dem Erstellen eines neuen Gallery-Themas in LibreOffice. Starten Sie das Programm. Die Gallery rufen Sie über das Menü „Ansicht“ und den dortigen Eintrag „Gallery“ auf. Die Gallery wird damit in der Seitenleiste geöffnet (früher wurde sie oberhalb des Bearbeitungsfensters unterhalb der Menüleiste angezeigt). Im oberen Bereich werden die vorhandenen Themen in Ihrem LibreOffice angezeigt Erstellen Sie nun über die Schaltfläche „Neues Thema“ und den folgenden Dialog ein solches Thema. Auf dem ersten Register („Allgemein“) des Dialogs geben Sie einen Namen für das Thema vor. Sofern Sie bereits zuvor Grafiken, Zeichnungen oder Multimedia-Dateien erstellt haben, können Sie diese über das Register „Dateien“ direkt dem Thema hinzufügen. Wir nennen das Thema „Tiere“, da wir einige Grafiken mit offenen Lizenzen aus dem Openclipart-Projekt in unserer Beispiel-Extension in der LibreOffice-Gallery verfügbar machen wollen.

3 Auslieferung von Inhalten mit Extensions

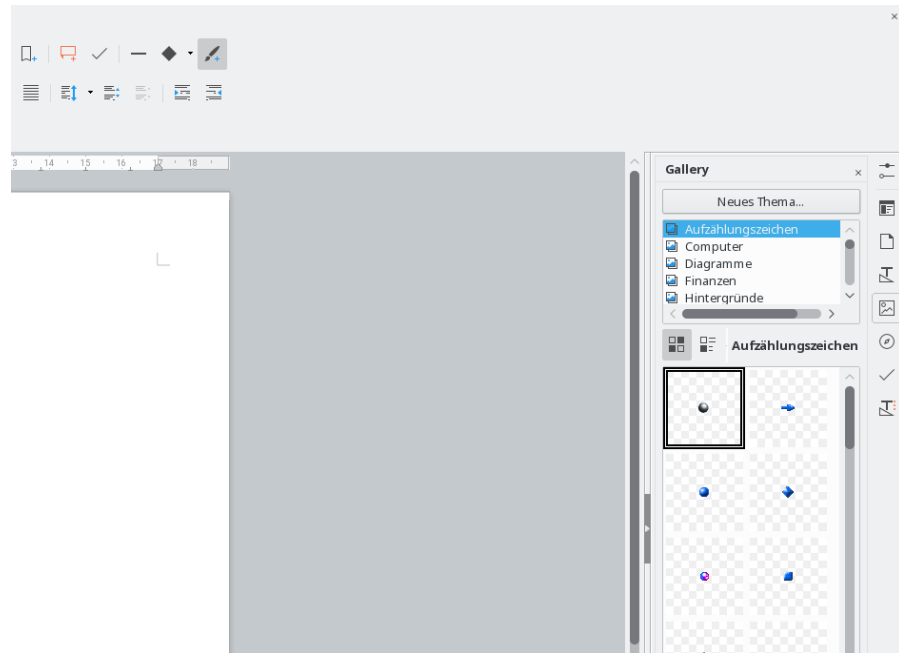


Abbildung 3.1: Die LibreOffice Gallery

Die Grafiken auf der Seite von OpenClipart (<https://openclipart.org>) werden dort unter Public Domain veröffentlicht. Dort wird dazu die Creative Commons Zero 1.0 Public Domain License (<https://creativecommons.org/publicdomain/zero/1.0/>) verwendet.

Für unser gerade erstelltes Tier-Thema in der Gallery laden wir mehrere Tiergrafiken von der OpenClipart-Webseite möglichst im Vektorformat (SVG) herunter, z.B. einen Elefanten (Elephant), einen Tiger und einen Wolf. Wir können diese Grafiken nun im SVG-Dateiformat in das gerade erstellte Thema der Gallery einfügen. Dazu rufen wir mit einem rechten Mausklick auf das Thema das Kontextmenü auf und dort wählen dort den Eintrag „Eigenschaften“ aus. Damit gelangen wir wieder in den Dialog, in dem wir auf dem Register „Dateien“ neue Grafiken oder andere Dateien dem Thema hinzufügen können. Oben im Dialog können wir in dem Ausklappmenü hinter Dateitypus vorgeben, welche Art von Datei wir hinzufügen wollen. Nach einem Mausklick auf die Schaltfläche „Hinzufügen“ öffnet sich der Dateidialog, mit dem wir zu dem Verzeichnis und der Datei navigieren, die wir einfügen wollen. Über die Schaltfläche „Öffnen“ wird das Hinzufügen sodann ausgelöst.

Zwar ist es möglich, SVG-Grafiken in die Gallery einzufügen, allerdings führt dies aktuell regelmäßig dazu, dass LibreOffice längere Zeit für das Hinzufügen benötigt, sehr viel Systemressourcen (CPU-Power) beansprucht und sich später auch beim Einfügen von Grafiken aus der Gallery in ein Dokument sehr viel Zeit nimmt bzw. scheitert. Die Grafiken sollten daher zunächst in ein Pixel-Dateiformat wie beispielsweise PNG umge-

3.1 Gallery-Extension

wandelt werden. Dafür kann z.B. das Programm Inkscape (<https://inkscape.org/de/>) verwandt werden, sofern eine Umwandlung nicht auf der Kommandozeile erledigt werden soll.

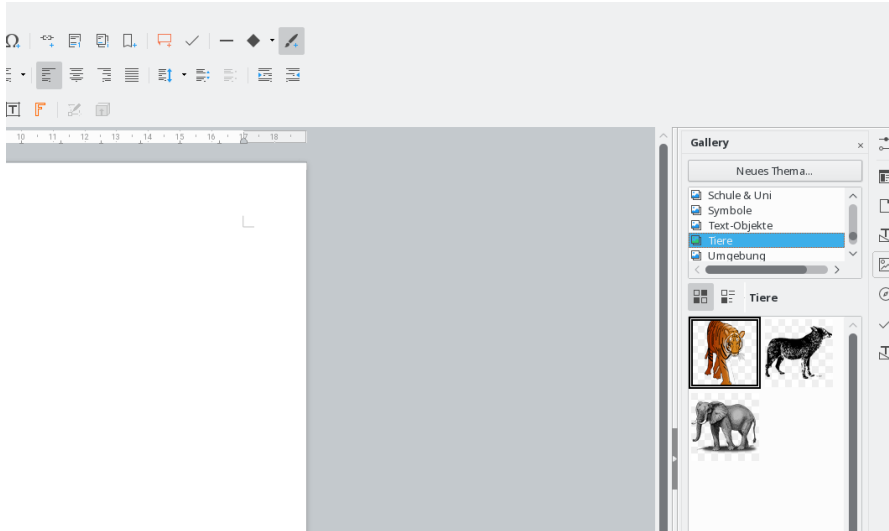


Abbildung 3.2: Das neue Gallery-Thema mit einigen Grafiken

Exkurs

Auf der Kommandozeile lassen sich Grafiken mit dem Programm „convert“ aus dem Programmpaket ImageMagick konvertieren. Soll nur von einem Dateiformat in ein anderes konvertiert werden, reicht ein einfaches Kommando nach folgendem Muster:

```
convert datei1.svg datei1.png
```

Die Datei „datei1.svg“ wird damit in die das Dateiformat PNG umgewandelt. Das Programm convert bietet darüber hinaus aber sehr viele Möglichkeiten, Grafiken schnell und komfortabel auf der Kommandozeile und viele Dateien in einem Durchgang zu bearbeiten. Ein Überblick zu den vielfältigen Optionen lässt sich auf der Seite <https://www.imagemagick.org/script/convert.php> gewinnen. Das Programmpaket ImageMagick ist sowohl für das Linux Betriebssystem und andere Betriebssysteme der Nix-Welt als auch für das Betriebssystem Windows verfügbar. Es kann von der Projektseite (<https://www.imagemagick.org>) heruntergeladen oder beispielsweise unter dem Betriebssystem Linux als Paket der jeweiligen Distribution installiert werden.

3.1.2 Die Dateien des neuen Gallery-Themas

Das neue Gallery-Thema ist nun erstellt und einige Dateien eingefügt worden. Es wurde von LibreOffice unter den Konfigurationsdateien des Benutzers abgelegt. Der

3 Auslieferung von Inhalten mit Extensions

genaue Speicherort und der Dateiname wird angezeigt, wenn im Kontextmenü des neuen Gallery-Themas der Befehl „Eigenschaften“ aufgerufen wird. Auf dem Register „Allgemein“ wird hinter „Ort“ der Pfad und Name des neuen Themas angezeigt. Ein Gallery-Thema besteht immer aus drei Dateien mit den Dateierweiterungen „sdg“, „sdv“ und „thm“. LibreOffice vergibt für ein neues Thema aktuell immer Dateinamen mit der Zeichenkette „neues thema“. Die Dateinamen lassen sich ändern. Es ist allerdings darauf zu achten, dass alle drei zusammengehörenden Dateien eines neuen Themas den gleichen Dateinamen vor der Dateierweiterung erhalten. Wir benennen die Dateien des neuen Themas also um in „tiere.sdg“, „tiere.sdv“ und „tiere.thm“. Damit haben wir die Gallery-Dateien für die neue Beispiel-Gallery-Extension fertig gestellt und können damit die Extension erstellen.

3.1.3 Bauen der Gallery-Extension

Im Kapitel Der Container OXT und sein Inhalt haben wir ein erste LibreOffice Extension in Ihrer Grundstruktur erstellt. Diese Extension soll nun um die weitere Funktionalität einer Gallery-Extension erweitert werden. Dazu legen wir im Grundverzeichnis einen weiteren Ordner „gallery“ an und kopieren in diesen die vorhergehenden Abschnitt erstellten drei Gallery-Dateien. Im Grundverzeichnis erstellen wir weiterhin eine neue Textdatei „paths.xcu“.

Die Datei paths.xcu

Die Datei paths.xcu beginnt mit einer XML-Definition, gefolgt von einer XML-Struktur:

```
<?xml version="1.0" encoding="UTF-8"?>

<oor:component-data oor:name="Paths"
  oor:package="org.openoffice.Office"
  xmlns:install="http://openoffice.org/2004/installation"
  xmlns:oor="http://openoffice.org/2001/registry"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <node oor:name="Paths">
    <node oor:name="Gallery" oor:op="fuse">
      <node oor:name="InternalPaths">
        <node oor:name="%origin%/gallery"
          oor:op="fuse"/>
      </node>
    </node>
  </node>
</oor:component-data>
```

Zunächst wird ein XML-Element für eine Extension definiert, die keinen Programmcode enthält. Darauf folgt der Pfad, in dem der Inhalt abgelegt werden soll. Mit dem Schlüsselwort „InternalPaths“ werden alle Verzeichnisse gekennzeichnet, die etwas enthalten können, die der Benutzer nicht ändern, aber beispielsweise durch Extension-Pakete erweitern, also mit eigenen Inhalten bevölkern kann. Hier ist es die LibreOffice Gallery. Das Schlüsselwort „fuse“ bewirkt das Vereinigen von bisherigem Inhalt der Gallery mit neuem Inhalt. Sofern bereits ein Verzeichnis mit dem entsprechenden Namen vorhanden ist, wird der Inhalt der Extension hinzugefügt, andernfalls wird ein neues Verzeichnis mit dem entsprechenden Namen angelegt.

Wenn Sie die Datei `paths.xcu` mit dem zuvor dargestellten Inhalt erstellt haben, wird diese allerdings noch nicht in der Extension aktiv eingebunden. Dafür ist eine kleine Ergänzung in der in Kapitel 2 erstellten Datei „`manifest.xml`“ um folgende Zeile erforderlich:

```
<manifest:file-entry manifest:media-type=
"application/vnd.sun.star.configuration-data"
manifest:full-path = "paths.xcu"/>
```

Diese Zeile ist zwischen:

```
<manifest:manifest
  xmlns:manifest="http://openoffice.org/2001/manifest">
  und
</manifest:manifest>
```

einzuführen. Anschließend kann die LibreOffice Extension mit dem in Kapitel 2 erstellten Buildskript gebündelt werden. Nach einer Installation der Extension mit dem LibreOffice Extension Manager sollte danach in der Gallery ein neues Thema „Tiere“ mit den entsprechenden Grafiken angezeigt werden.

Während der Installation werden Sie feststellen, dass Sie lediglich eine weiße Seite im Dialog „Lizenzvertrag für die Extension-Software“ angezeigt bekommen. Dies beruht darauf, dass die im Kapitel 2 erstellten Dateien für die Lizenzinformationen noch ohne Inhalt sind. Bevor Sie eine Extension veröffentlichen müssen Sie diese Dateien bearbeiten und dort die erforderlichen Informationen ablegen.

Eine entsprechende Vorlage für eine LibreOffice Gallery Extension finden Sie im Github-Repository unter folgender Adresse: https://github.com/andreasma/extensionbook/tree/master/extensiontemplates/gallery_extension/galleryextensionexample01.

3.2 Template-Extension

Mit einer LibreOffice Extension lassen sich den Anwendern auch Vorlagen (Templates) für ihre täglichen Arbeiten oder bestimmte Aufgaben zur Verfügung stellen. Die Vorlagen werden dann, sofern sie für einzelne Benutzer installiert werden, in seinem Benutzerprofil abgelegt. Wo sich diese Einstellungen und insbesondere die Vorlagen

3 Auslieferung von Inhalten mit Extensions

befinden (oder später befinden werden, sofern bisher noch keine vorhanden sind), erfahren Sie bei einem Aufruf von „Optionen“ im Menü „Extras“. Dort befindet sich unter „LibreOffice“ der Eintrag „Pfade“. Wenn Sie diesen auswählen, sehen Sie auf der rechten Seite im Dialog die einzelnen Pfade (Paths), so auch den für die Vorlagen („Dokumentvorlagen“).

Gegenüber der Gallery-Extension aus dem vorhergehenden Abschnitt ändern sich für eine LibreOffice Extension mit Dokumentvorlagen nur das Verzeichnis, in dem die Vorlagen abgelegt werden, und wenige Einträge in der Datei „paths.xcu“. Sofern der Eintrag zu den Pfaden für die Vorlagen in der Benutzerkonfiguration auf ein Verzeichnis „templates“ zeigt, ist lediglich das Wort „gallery“ durch das Wort „template“ zu ersetzen. Die Datei paths.xcu sieht dann wie folgt aus:

```
<?xml version="1.0" encoding="UTF-8"?>

<oor:component-data oor:name="Paths"
  oor:package="org.openoffice.Office"
  xmlns:install="http://openoffice.org/2004/installation"
  xmlns:oor="http://openoffice.org/2001/registry"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <node oor:name="Paths">
    <node oor:name="Template" oor:op="fuse">
      <node oor:name="InternalPaths">
        <node oor:name="%origin%/template"
          oor:op="fuse"/>
      </node>
    </node>
  </node>

</oor:component-data>
```

Nun fehlt noch das Verzeichnis mit den Dokumentenvorlagen. Dazu legen wir ein solches mit dem Namen „template“ im Grundverzeichnis an. Je nachdem welche Arten von Vorlagen mit der Extension zur Verfügung gestellt werden sollen, ist im Unterverzeichnis template sinnvoller Weise eine weitere Struktur mit eigenen Unterverzeichnissen anzulegen. Diese in der Dateihierarchie tieferen Unterverzeichnisse enthalten dann Dokumentenvorlagen zu einzelnen Themengebieten, beispielsweise Vorlagen für das Schreibprogramm Writer oder Präsentationsvorlagen.

Es gibt bereits einige von LibreOffice für die von ihm mitgelieferten Dokumentenvorlagen verwendete Unterverzeichnisnamen, die bei der Benennung der Unterverzeichnisse im Verzeichnis template der Extension berücksichtigt werden sollten. Sie können sich diese Struktur im Programmverzeichnis von LibreOffice im Unterverzeichnis /share/template/common ansehen. Einige Beispiele:

- officore = Geschäftliche Korrespondenz

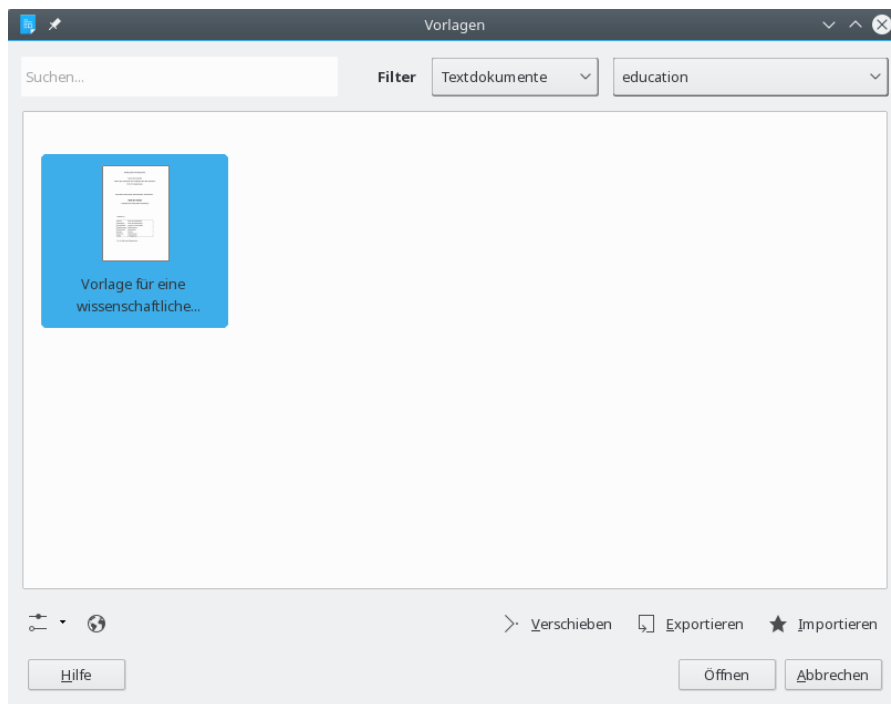
3.2 Template-Extension

- offimisc = Sonstige geschäftliche Dokumente
- personal = Private Korrespondenz und Dokumente
- presnt = Präsentationsvorlagen
- layout = Präsentationshintergründe
- misc = Diverses

Um die Beispiel-Extension für Dokumentenvorlagen mit entsprechenden Dokumenten zu bevölkern, laden Sie von der LibreOffice Template-Seite <https://extensions.libreoffice.org/templates> Vorlagendokumente aus verschiedenen Themenbereichen herunter, beispielsweise eine oder mehrere Briefvorlagen, Präsentationsvorlagen und Vorlagen für Lebensläufe.

Sie können auch weitere Unterverzeichnisse für themenspezifische Vorlagen anlegen, beispielsweise „education“ für Bildung oder „magazine“ für Zeitschriften- bzw. Magazin-Vorlagen. Entsprechende Vorlagendokumente finden Sie auf der LibreOffice Template Webseite.

Sofern Sie Ihre Beispiel-Vorlagen-Extension mit solchen weiteren Unterverzeichnissen mit entsprechenden Dokumentvorlagen ausgestattet haben, können Sie diese Themenbereiche in der Vorlagenverwaltung von LibreOffice gefiltert anzeigen lassen. Diesen Dialog rufen Sie über das Menü „Datei - Vorlagen - Vorlagen verwalten“ auf. Im Dialog können Sie dann filtern nach der Art der Dokumente, z.B. Textdokument, und nach Kategorien, z.B. Education. Sofern entsprechende Vorlagen vorhanden sind, werden Sie unten im Dialog angezeigt.



3 Auslieferung von Inhalten mit Extensions

Abbildung 3.3: Der Dialog Vorlagen verwalten mit neuer Kategorie education

Die Benennung der Kategorie mit „education“ im Filter der Vorlagenverwaltung lässt sich wie auch die Benennung anderer Kategorien sprachlich anpassen. Dazu legen Sie im Verzeichnis „template“ der Extension eine neue Datei „groupuinames.xml“ an. Mit dieser werden dann für die einzelnen Kategorien die Namen gesetzt. Die Datei kann dann beispielsweise folgenden Inhalt bekommen:

```
<?xml version="1.0" encoding="UTF-8"?>
<groupuinames:template-group-list
  xmlns:groupuinames=
    "http://openoffice.org/2006/groupuinames">
  <groupuinames:template-group groupuinames:name="education"
    groupuinames:default-ui-name="Bildung" />
  <groupuinames:template-group groupuinames:name="presnt"
    groupuinames:default-ui-name="Präsentationen" />
</groupuinames:template-group-list >
```

Es ist nicht nötig, Verzeichnisnamen für Verzeichnisse aus der Standardinstallation von LibreOffice in der neuen XML-Datei erneut vorzugeben. Der zweite groupuinames-Eintrag war daher nicht erforderlich. Es wäre aber möglich gewesen, den mit der Extension neu im Unterverzeichnis „presnt“ verfügbar gemachten Präsentationsvorlagen einen abweichenden Kategorienamen zuzuordnen, beispielsweise „Community-Präsentationen“. Die von LibreOffice mitgelieferten Präsentationsvorlagen würden dann weiterhin mit der Kategorie „Präsentationen“ angezeigt, während die neuen der Extension unter „Community-Präsentationen“ eingeordnet wären.

Eventuell wollen Sie eine Extension mit Vorlagen für unterschiedliche Sprachen bauen, wobei dem Anwender nur die Vorlagen aufgelistet werden, die zu seiner Spracheinstellung passen. Dies ist mit den Einstellungen für eine Multilanguage-Template- bzw. Mehrsprachen-Vorlagen-Extension möglich. Zunächst werden im Unterverzeichnis „template“ für die vorgesehenen Lokalisierungen neue Unterverzeichnisse angelegt. Die Benennung hat dabei der ISO-Sprachkodierung zu folgen, also beispielsweise „en-US“ für englischsprachige, „de“ für deutschsprachige und „fr“ für französischsprachige Vorlagen. In diesen Unterverzeichnissen für die einzelnen Lokalen, werden dann jeweils wieder Unterverzeichnisse für die einzelnen Kategorien angelegt. Um deren Darstellung bzw. Benennung im Kategorien-Filter des Vorlagenverwaltungs-Dialogs zu steuern, werden in den Unterverzeichnissen der Lokalen jeweils wieder „groupuinames.xml“ Dateien nach dem zuvor dargestellten Muster angelegt.

Danach bedarf es noch einer kleinen Änderung in der Datei „paths.xcu“. Dort ist die Zeile:

```
<node oor:name="%origin%/template" oor:op="fuse"/>
```

durch die Zeile:

```
<node oor:name="%origin%/template/$(vlang)" oor:op="fuse"/>
```


zu ersetzen. Die Ergänzung „\$(vlang)“ bewirkt, dass nur auf die Vorlagen der Extension in dem zur Spracheinstellung der Programmoberfläche passenden Unterverzeichnis zugegriffen und diese dem Anwender präsentiert werden. Die Vorlagen aus den Unterverzeichnissen mit anderen Sprachkodierungen werden ausgeblendet.

Es ist nicht möglich in der Extension zusätzlich noch einen Bereich mit sprachunabhängigen Vorlagen anzulegen. Sofern solche Vorlagen - beispielsweise Präsentationshintergründe - allen Benutzern, unabhängig von der Oberflächen-Spracheinstellung verfügbar sein sollen, ist ihre Ablage folglich in dem entsprechenden Unterverzeichnis jeder Lokale vorzunehmen.

Eine entsprechende Vorlage für eine LibreOffice Template(Vorlagen)-Extension finden Sie im Github-Repository unter folgender Adresse: https://github.com/andreasma/extensionbook/tree/master/extensiontemplates/template_extension/templateextensionexample01.

3.3 AutoText-Extension

Das Erstellen von Textdokumenten lässt sich insbesondere auch im beruflichen Umfeld mit dem Aufruf von Textbausteinen deutlich beschleunigen und erleichtern. LibreOffice bietet dazu die Möglichkeit, sogenannte Autotexte zu speichern. Dies sind im Regelfall häufiger verwendete Texte, die in LibreOffice auch mit Grafiken, Tabellen und Feldbefehlen verbunden sein können. Mit solchen Autotexten lassen sich dann beispielsweise große Teile von Schriftsätzen erstellen und diese auch mit individualisierten Daten wie Aktenzeichen oder Datum des Bezugsschreibens versehen.

Im beruflichen Umfeld werden oft eine Vielzahl von Standardtexten entwickelt, mit denen sich Anfragen von Kunden und Lieferanten schnell, effektiv und zielgenau beantworten lassen. Solche Standardtexte können, nachdem sie von einem Anwender erstellt worden sind, mit Hilfe einer Extension verteilt und so allen Benutzern zugänglich gemacht werden.

Erster Schritt auf dem Weg zu einer AutoText-Extension ist das Erstellen von Autotexten. Diese werden dann von LibreOffice in einem gezippten Archiv mit der Dateierweiterung „.bau“ im Benutzerverzeichnis gespeichert. Wo dieses für den Benutzer abgelegt wird, erfahren Sie bei Aufruf von „Optionen“ im Menü Extras von LibreOffice in dem Dialog unter „LibreOffice - Pfade“. Dort ist der Pfad (oder mehrere) hinter dem Eintrag „AutoText“ gelistet.

Welche Textbausteine in der installierten Version von LibreOffice für den Benutzer bereits vorhanden sind, lässt sich folgendermaßen prüfen:

- Rufen Sie im Menü Extras den Befehl „Makros - Makro ausführen“ auf.
- Öffnen Sie im Dialog links „LibreOffice Makros“ und unter diesen „Gimmicks“ sowie danach „AutoText“.
- Im Dialog auf der rechten Seite wählen Sie mit einem Mausklick „Main“.
- Schließen Sie den Dialog über die Schaltfläche „Ausführen“ ab.

3 Auslieferung von Inhalten mit Extensions

- Nach Ausführen des Makros wird Ihnen ein neues Dokument mit den verfügbaren Textbausteinen angezeigt, das Sie sich ausdrucken oder speichern können.

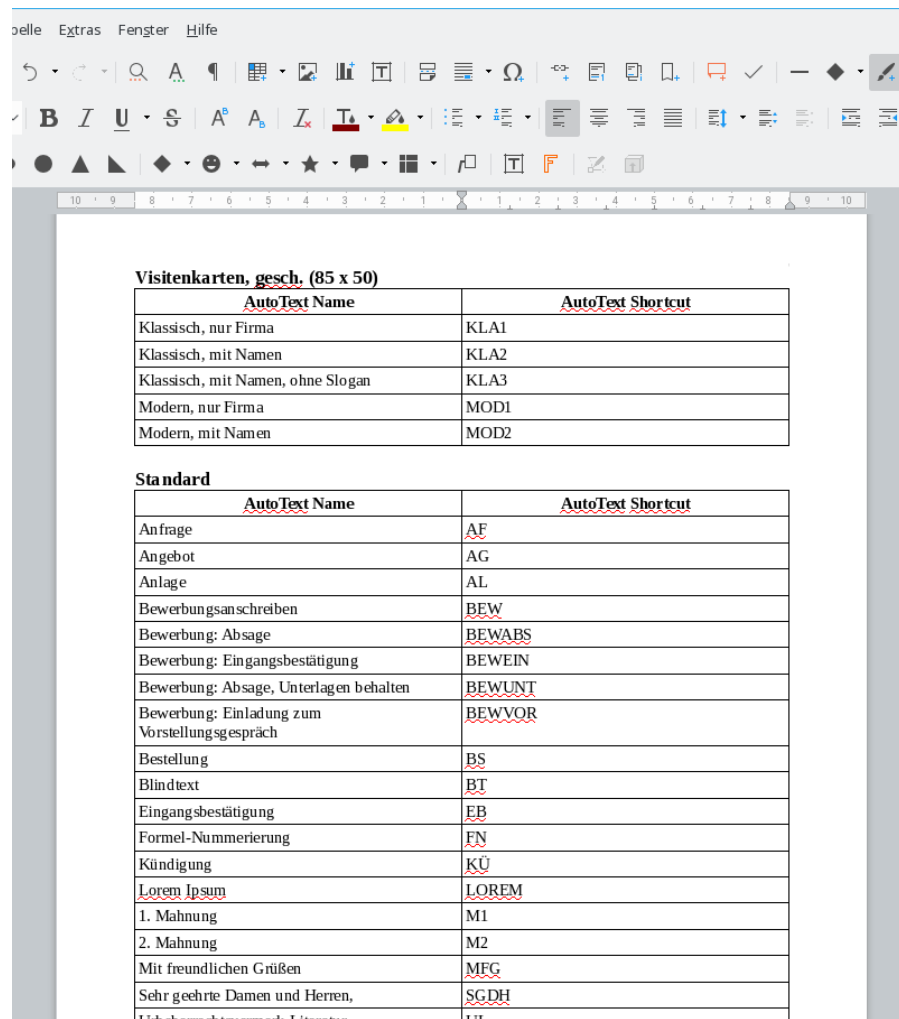


Abbildung 3.4: Anzeige der verfügbaren Textbausteine

Auch wenn auf dem System Autotexte in anderen Sprachen vorhanden sind, werden mit dem Makro nur diejenigen ausgegeben, die der Spracheinstellung der Oberfläche von LibreOffice entsprechen. Diese werden sortiert nach unterschiedlichen Kategorien aufgelistet, beispielsweise solche für Visitenkarten und solche für normale Textdokumente. Am Ende der Liste befindet sich ein Bereich bzw. eine Kategorie, in der Ihre eigenen Autotexte aufgelistet werden.

Nachdem Sie nun ermittelt bzw. angezeigt haben, welche Textbausteine bereits in Ihrem System vorhanden sind, geht es nun darum, neue Bausteine für eine AutoText-

Extension zu erstellen. Dazu rufen wir in LibreOffice im Menü „Extras“ den Eintrag „AutoText“ auf. In dem aufgerufenen Dialog klicken Sie auf die Schaltfläche „Pfad“ und schauen Sie, ob der Pfad zum AutoText-Verzeichnis in den Konfigurations-Einstellungen Ihres Benutzers bereits existiert. Sofern er nicht existiert, fügen Sie diesen nun hinzu und schließen den Dialog über die Schaltfläche „OK“ ab. Danach rufen Sie im AutoText-Dialog die Kategorien über die entsprechende Schaltfläche auf. Fügen Sie dort oben im Feld „Kategorie“ den Namen Ihrer neuen Kategorie ein, z.B. „Standardantworten“. Unter Pfad wählen Sie den zuvor unter Pfaden eventuell hinzugefügten Pfad zu den Autotexten für Ihren Benutzer und schließen den Dialog über die Schaltfläche „OK“ ab. Sie haben damit im Unterverzeichnis „autotext“ eine neue gezippte bau-Datei mit dem Namen Ihrer Kategorie, also beispielsweise „Standardantworten.bau“.

Schließen Sie den AutoText-Dialog wieder und schreiben Sie Ihren ersten Textbaustein in das geöffnete Textdokument. Diesen Text markieren Sie danach und rufen wieder den AutoText-Dialog auf. Geben Sie im Feld Name nun die gewünschte Bezeichnung für Ihren Textbaustein und rechts ein Kürzel ein. Danach ordnen Sie den neuen Textbaustein noch einer Kategorie zu. Dazu wählen Sie die von Ihnen zuvor erstellte Kategorie im Feld unten aus. Danach klicken Sie auf die Schaltfläche „AutoText“ und wählen im Ausklappenmenü „Neu“ aus. Damit haben Sie Ihren Textbaustein der Kategorie hinzugefügt. Nach Schließen des AutoText-Dialogs verfahren Sie entsprechend mit weiteren Textbausteinen, die Sie für die neue Extension verfassen wollen. Die AutoTexte sind dann jeweils unter Eingabe des vorgegebenen Kürzels und die Taste „F3“ einzufügen.

Nachdem Sie nun die Bau-Dateien mit den Textbausteinen fertig gestellt haben, müssen diese nur noch in eine LibreOffice extension eingebaut werden. Sie erstellen dazu zunächst ein Unterverzeichnis „autotext“ im Grundverzeichnis der Extension. Danach erstellen Sie eine neue oder bearbeiten Sie die Datei „paths.xcu“, die sich ebenfalls im Extension-Grundverzeichnis befinden muss. Die Datei sollte danach folgenden Inhalt haben:

```
<?xml version="1.0" encoding="UTF-8"?>

<oor:component-data oor:name="Paths"
  oor:package="org.openoffice.Office"
  xmlns:install="http://openoffice.org/2004/installation"
  xmlns:oor="http://openoffice.org/2001/registry"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <node oor:name="Paths">
    <node oor:name="AutoText" oor:op="fuse">
      <node oor:name="InternalPaths">
        <node oor:name="%origin%/autotext"
          oor:op="fuse"/>
      </node>
    </node>
  </node>
</oor:component-data>
```

3 Auslieferung von Inhalten mit Extensions

```
</node>  
</node>  
</oor:component-data>
```

Gegenüber der im vorhergehenden Abschnitt erläuterten Vorlagen-Extension ändern sich in der Autotext-Extension das Schlüsselwort Name. Dieses lautet nunmehr „AutoText“. Außerdem ändert sich natürlich auch der Pfad zu dem Verzeichnis, in dem die Textbaustein-Zip-Dateien liegen. Dieser verweist auf das Unterverzeichnis „autotext“. In letzteres Unterverzeichnis kopieren Sie nun die von Ihnen erstellten Textbaustein-Zip-Dateien. Danach bauen Sie die Extension mit dem Python-Build-Skript. Entfernen Sie aus dem Autotext-Verzeichnis ihrer Benutzerkonfiguration die erstellten Textbaustein-Zip-Archive und kontrollieren Sie, dass Ihre Textbausteine sich nicht mehr in der Liste der Autotexte befinden. Im Anschluss installieren sie mit dem Extension-Manager ihre AutoText-Extension. Sie sollten nun wieder ihre Textbausteine in der Liste der Autotexte angeboten bekommen.

Die Textbausteine werden regelmäßig nur für eine Spracheinstellung von LibreOffice sinnvoll verwendbar sein, so dass die Autotexte nach Lokalisierungen gruppiert im Unterverzeichnis „autotext“ abgelegt werden sollten. Dazu erstellen Sie für jede Lokalisierung, für die Sie eine Textbaustein-Zip-Datei generiert haben, ein Unterverzeichnis, dessen Benennung der ISO-Sprachkodierung folgt, also beispielsweise „en-US“ für englischsprachige, „de“ für deutschsprachige und „fr“ für französischsprachige Vorlagen. Die angelegten Unterverzeichnisse dürfen später nicht leer sein, d.h., sie müssen jeweils zumindest eine Textbaustein-Zip-Datei enthalten. Andernfalls wird nach der Installation der Extension bei einem Zugriff auf das Verzeichnis im Rahmen des Aufrufs der AutoText-Funktion eine Fehlermeldung angezeigt.

Sofern Sie die Textbausteine nach Lokalisierungen gruppiert in Unterverzeichnissen ablegen, müssen Sie die Datei „paths.xcu“ anpassen. Dort ist die Zeile:

```
<node oor:name=\"%origin%/autotext \" oor:op=\"fuse\"/>
```

durch die Zeile:

```
<node oor:name=\"%origin%/autotext/$(vlang)\" oor:op=\"fuse\"/>
```

zu ersetzen.

Eine entsprechende Vorlage für eine LibreOffice AutoText-Extension finden Sie im Github-Repository unter folgender Adresse: https://github.com/andreama/extensionbook/tree/master/extensiontemplates/auto-text_extension/autotextextensionexample01.

3.4 Autokorrektur-Extension

Eine weitere schöne Möglichkeit besteht darin, LibreOffice eigene automatische Korrekturen (Autokorrektur) beizubringen und diese anschließend in eine Extension verpackt an andere Benutzer auszuliefern. Diese Autokorrektur-Funktion hat einen Beitragenden zu LibreOffice dazu gebracht, eine Erweiterung namens PCSteno für LibreOffice zu erstellen. Diese besteht aus einer selbst erstellten Autokorrektur-Datei

`acor_de-DE.dat`, die vom Anwender in das Unterverzeichnis „autocorr“ kopiert werden muss. Nach einem Neustart von LibreOffice kann damit mit wenigen Tastaturanschlägen Text schnell eingegeben werden. Der Benutzer muss allerdings die entsprechenden Kürzel/Abkürzungen erlernen; z.B. „nmh“ für „nicht mehr“ oder „svj“ für „soviel“.

Das Ganze lässt sich allerdings auch eleganter mit einer LibreOffice-Extension an den Anwender ausliefern. Wie dies funktioniert, soll nun kurz unter Verwenden des Beispiels PCSteno gezeigt werden. Zunächst muss ein neues Verzeichnis „testpcsteno“ erstellt werden. Darin legen wir eine Datei `description.xml` etwa mit folgendem Inhalt an:

```
<?xml version="1.0" encoding="UTF-8"?>
<description
  xmlns="http://openoffice.org/extensions/description/2006"
  xmlns:xlink="http://www.w3.org/1999/xlink">
  <identifier value="org.libreoffice.testpcsteno" />
  <version value="0.2" />
  <platform value="all" />
  <display-name>
    <name lang="de">Test PC Steno für Libreoffice </name>
    <name lang="en">Test PC Steno for LibreOffice </name>
  </display-name>
  <publisher>
    <name xlink:href="http://amantke.de/blog">
      Andreas Mantke</name>
  </publisher>
</description>
```

Für das Beispiel habe ich im Wert (value) des Identifiers „libreoffice“ als Namen des Unternehmens eingesetzt und als Veröffentlicher (Publisher) meinen eigenen Namen und mein Blog eingetragen. Für die Veröffentlichung der entsprechenden Extension sind diese Werte durch die Daten des Extension-Autors zu ersetzen.

Damit die Extension auch deinstalliert werden kann, muss noch eine Datei `manifest.xml` in einem Unterverzeichnis `META-INF` hinzugefügt werden:

```
<?xml version="1.0" encoding="UTF-8"?>
<manifest:manifest
  xmlns:manifest="http://openoffice.org/2001/manifest">
  <manifest:file-entry
    manifest:media-type="application/
vnd.sun.star.configuration-data"
    manifest:full-path="paths.xcu"/>
</manifest:manifest>
```

In dieser Datei wird eine Datei `paths.xcu` referenziert, die nun noch zu erstellen ist mit folgendem Inhalt:

```
<?xml version='1.0' encoding='UTF-8'?>
```

3 Auslieferung von Inhalten mit Extensions

```
<oor:component-data
  oor:package="org.openoffice.Office"
  oor:name="Paths"
  xmlns:install="http://openoffice.org/2004/installation"
  xmlns:oor="http://openoffice.org/2001/registry"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <node oor:name="Paths">
    <node oor:name="AutoCorrect" oor:op="fuse">
      <node oor:name="InternalPaths">
        <node oor:name="%origin%/autocorr"
          oor:op="fuse"/>
        </node>
      </node>
    </node>
  </oor:component-data>
```

Nach den allgemeinen Definitionen werden in den Zeilen beginnend mit „node“ die speziellen Daten für eine Autokorrektur-Extension gesetzt. Der `oor:name` wird mit „AutoCorrect“ definiert und der Pfad wird auf das Verzeichnis „autocorr“ gesetzt. Die übrigen Vorgaben in den Node-Tags entsprechen denen, die bereits weiter oben im Abschnitt zur Gallery-Extension besprochen wurden.

Nun fehlt noch die Autokorrektur-Datei, die mit der Extension verteilt werden soll. Dazu erstellt man ein neues Unterverzeichnis `autocorr`. In dieses wird sodann die Datei `arcor_de-DE.dat` abgelegt.

Die Extension kann nun mit dem Build-Skript, das oben im Abschnitt „Erstellen der Extension-Datei“ beschrieben und erstellt wurde. Diese Datei `build` mit dem enthaltenen Python-Skript muss sich ebenfalls in diesem Abschnitt angelegten Verzeichnis „testpcsteno“ befinden. Sein Aufruf ist ebenfalls oben beschrieben.

3.4 Autokorrektur-Extension

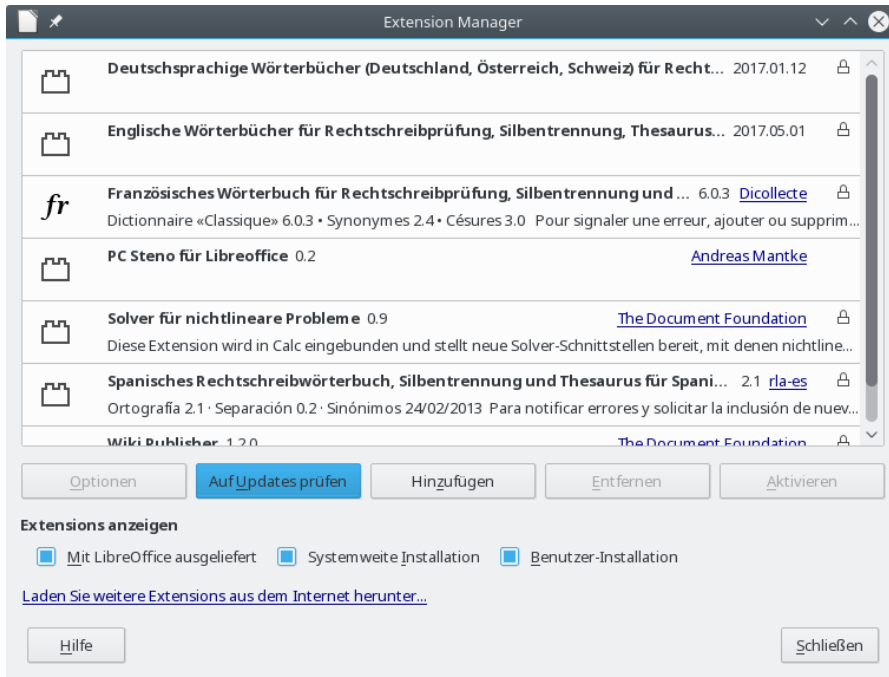


Abbildung 3.5: Anzeige Autokorrektur-Extension im Extension-Manager nach Installation

Es ist auch für Autokorrektur-Extensions möglich, die Autokorrekturen nach Lokalisierungen auszuliefern. Dazu ist die Zeile

```
<node oor:name="%origin%/autocorr" oor:op="fuse"/>
```

durch

```
<node oor:name="%origin%/autocorr/$(vlang)" oor:op="fuse"/>
```

zu ersetzen. In dem Unterverzeichnis ist für die auszuliefernden Lokalisierungen ein Unterverzeichnis anzulegen und die Autokorrektur-Datei dort abzulegen. Als Beispiel für die Lokalisierung Deutsch ist somit ein Unterverzeichnis „de“ anzulegen und darin die Autokorrektur-Datei für diese Lokalisierung abzulegen.

Mit der Installation der Autokorrektur-Extension wird allerdings für LibreOffice eine zuvor vom Benutzer erstellte eigene Autokorrektur-Liste(/-Datei) nicht mehr sichtbar. Sie bleibt zwar im Unterverzeichnis „autocorr“ erhalten, die dort enthaltenen Korrekturen sind allerdings für den Benutzer nicht mehr verfügbar. Sie gehen jedoch nicht komplett verloren, sondern sind nach der Deinstallation der Autokorrektur-Extension für den Benutzer wieder verfügbar. Die mit der Extension übermittelte Autokorrektur-Datei genießt somit Priorität vor der Datei im Unterverzeichnis „autocorr“.

Da sich bei der Einbindung von Extensions Änderungen ergeben haben, ist es wichtig, zumindest eine minimale Version von LibreOffice anzugeben. Da ältere Versionen

3 Auslieferung von Inhalten mit Extensions

von LibreOffice keine Sicherheitsupdates mehr erhalten, sollte als minimale Version zumindest LibreOffice 4.2 gewählt werden (mit dieser Version gab es einige Änderungen für Extensions). Der Datei `description.xml` werden hierfür folgende Zeilen hinzugefügt:

```
<dependencies xmlns:lo=
  "http://libreoffice.org/extensions/description/2011">
  <lo:LibreOffice-minimal-version name="LibreOffice 4.2"
    value="4.2"/>
</dependencies>
```

Der Extension fehlt nun noch eine Beschreibung, die sowohl eine Erläuterung des Zwecks als auch der Funktionsweise der Extension gibt. Hierzu gibt es das `Description`-Tag, das auf eine entsprechende Textdatei mit der Beschreibung verweisen kann. Es ist dabei möglich, für jede Lokalisierung auf eine eigene Datei zu verweisen. Die hier als Beispiel gewählte Autokorrektur-Erweiterung `PCSteno` ist nur in deutscher Sprache lokalisiert. Es soll jedoch dennoch bereits die Lokale im `Description`-Tag vorgesehen werden. Die Ergänzung zur Datei `description.xml` lautet dann folgendermaßen:

```
<extension-description>
  <src xlink:href="description/description_de.txt"
    lang="de" />
</extension-description>
```

Nun muss noch ein Unterverzeichnis „description“ erstellt und dort eine neue Textdatei `description_de.txt` angelegt werden. Diese Datei kann einen längeren Beschreibungstext enthalten, es ist allerdings nicht sinnvoll, dort eine vollständige Dokumentation unterzubringen, da das Rendern im Extension-Manager bei längeren Texten nicht optimal ist. Es sollten daher nur einige grundsätzliche Erläuterungen und Hinweise aufgenommen werden und im Übrigen auf eine Datei mit der vollständigen Dokumentation verwiesen werden, die beispielsweise auf der LibreOffice Extensions-Website <http://extensions.libreoffice.org/extensions> vom Autor der Extension zum Download (im Dateiformat ODT oder PDF) bereit gestellt werden kann.

Bevor die Extension auf der LibreOffice Extensions-Website veröffentlicht werden kann, ist die Datei `description.xml` noch um den Verweis auf die Lizenz zu ergänzen, unter der die Extension veröffentlicht wird und die vom Anwender bei der Installation akzeptiert werden muss. Hierzu wird zunächst ein weiteres Unterverzeichnis „registration“ angelegt, in dem, gegebenenfalls in unterschiedlichen Lokalisierungen Dateien mit den Lizenztexten abgelegt werden, z.B. für die deutsche Lokalisierung `license_de.txt`.

Anschließend wird die Datei `description.xml` um folgende Eintragung ergänzt:

```
<registration>
  <simple-license accept-by="admin">
    <license-text xlink:href="registration/license_de.txt"
      lang="de" />
  </simple-license>
</registration>
```


3.5 Farbpaletten-Extension

Unsere Beispiel-Autokorrektur-Extension ist zwar aktuell nur in deutscher Sprache lokalisiert, allerdings ist es immer sinnvoll, die Extension von Beginn an auf eine mögliche Lokalisierung in anderen Sprachen vorzubereiten.

Sofern in Bezug auf ein Release der Extension besondere Informationen zu geben sind, insbesondere wenn sich die Bedienung verändert oder weitere Funktionen hinzugekommen sind, können der Extension besondere Release-Notes hinzugefügt werden. Hierzu gibt es das entsprechende XML-Tag. Allerdings werden diese Release-Notes dem Anwender in LibreOffice selbst aktuell nicht angezeigt. Er kann sich diese nur ansehen, wenn er die Extension-Datei entpackt und dort die Release-Notes-Datei mit einem Anzeigeprogramm für Textdateien oder einem Texteditor öffnet. Insoweit sind entsprechende Beigaben der Extension für den Anwender nur etwas erschwert wahrzunehmen. Dies sollte den Autor einer Extension trotzdem nicht davon abhalten, entsprechende Release-Notes der Extension beizufügen.

Eine entsprechende Vorlage für eine LibreOffice Autokorrektur-Extension finden Sie im Github-Repository unter folgender Adresse: https://github.com/andreasma/extensionbook/tree/master/extensiontemplates/autocorrect_extension/autocorrectextensionexample01.

3.5 Farbpaletten-Extension

LibreOffice bringt eine Reihe von Farb-Paletten mit. Sie können sich die verfügbaren Paletten beispielsweise in dem Dialog „Fläche“ anzeigen lassen, den Sie im Programmmodul Draw im Menü „Format“ aufrufen können.

3 Auslieferung von Inhalten mit Extensions

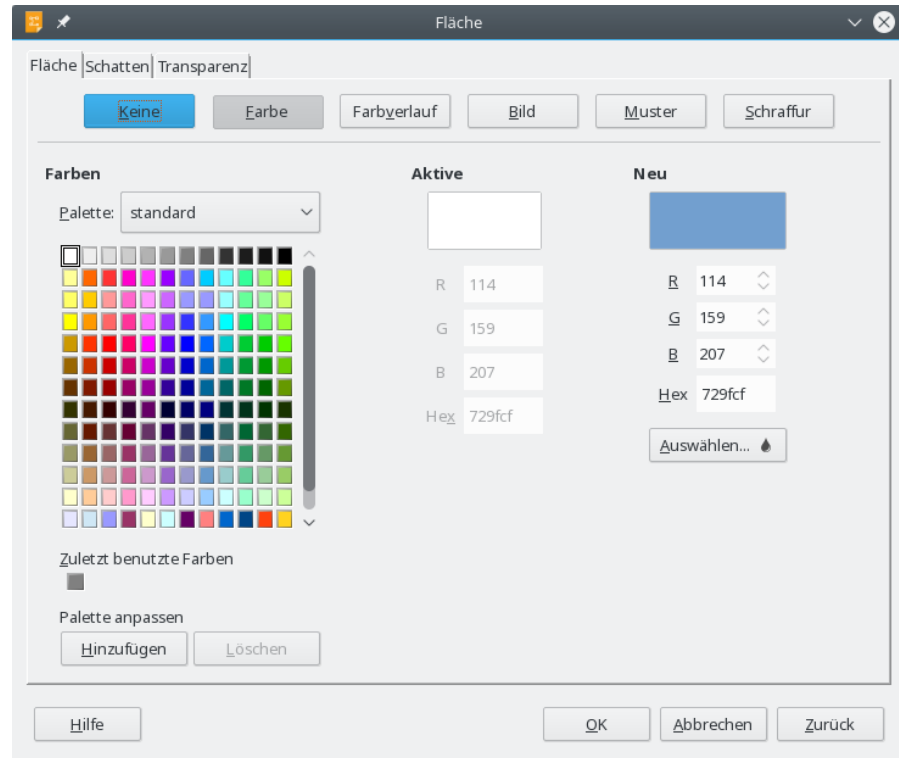


Abbildung 3.6: Der Dialog Fläche

Sie können nun eigene Farb-Paletten beispielsweise für spezielle Anwendungsfälle, etwa für die Corporate Identity eines Unternehmens, erstellen und anschließend mittels einer LibreOffice-Extension verteilen.

Zunächst erstellen Sie im Grundverzeichnis der Farbpaletten-Extension ein neues Unterverzeichnis „palette“. In diesem Unterverzeichnis erstellen Sie eine neue Datei mit dem Namen der Palette und der Dateiendung „.soc“, zum Beispiel „meinefirma.soc“. Rufen Sie die neue Datei in einem Texteditor auf und fügen Sie Folgendes dort ein:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- https://meinefirma.de -->
<ooo:color-table xmlns:office=
  "urn:oasis:names:tc:opendocument:xmlns:office:1.0"
  xmlns:draw="urn:oasis:names:tc:
    opendocument:xmlns:drawing:1.0"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:svg="http://www.w3.org/2000/svg"
  xmlns:ooo="http://openoffice.org/2004/office">
```

```
</ooo:color-table>
```

Innerhalb des XML-Tag `<ooo:color-table>` legen Sie danach die einzelnen Farben nach folgendem Muster an:

```
<draw:color draw:name="Farbname"
  draw:color="#Farbkodierung in Hexadezimal" />
```

Anstelle von `Farbname` tragen Sie dabei den individuellen Namen ihre benutzerdefinierten Farbe ein. Hinter „draw color“ folgt nach dem „#“ die Farbe in hexadezimaler Kodierung. Sofern Sie nur die Werte ihrer Farben nur im System RGB vorliegen haben, können Sie den Dialog „Fläche“ zum Umrechnen in das Hexadezimal-System verwenden. Geben Sie dazu auf der rechten Seite einfach die RGB-Werte ein und lesen Sie den Wert im Feld „Hex“ ab.

Wenn Sie die Datei mit der Farbpalette fertig gestellt haben, müssen Sie noch die Datei „paths.xcu“ im Grundverzeichnis neu anlegen oder ändern mit folgendem Inhalt:

```
<?xml version="1.0" encoding="UTF-8"?>

<oor:component-data oor:name="Paths"
  oor:package="org.openoffice.Office"
  xmlns:install="http://openoffice.org/2004/installation"
  xmlns:oor="http://openoffice.org/2001/registry"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <node oor:name="Paths">
    <node oor:name="Palette" oor:op="fuse">
      <node oor:name="InternalPaths">
        <node oor:name="%origin%/palette"
          oor:op="fuse"/>
      </node>
    </node>
  </node>
</oor:component-data>
```

Der Unterschied gegenüber den zuvor beschriebenen LibreOffice-Extensions besteht in der Bezeichnung hinter „oor:name“, der nun „Palette“ lautet, sowie dem Pfad in dem weiteren Knoten, der nun auf das Unterverzeichnis „palette“ verweist.

Mit der nun fertig gestellten Datei „paths.xcu“ können Sie die Extension wieder mit dem in der Vorlage enthaltenen Skript „build“ automatisch bauen lassen.

Eine entsprechende Vorlage für eine LibreOffice Farbpaletten-Extension finden Sie im Github-Repository unter folgender Adresse: https://github.com/andreasma/extensionbook/tree/master/extensiontemplates/color_extension/colorextensionexample01.

4 Ein neues Icon-Set mit einer Extension ausliefern

Das Extension-Management von LibreOffice bietet die Option, auch ein komplettes neues Set von Icons (für das Programm) auszuliefern. Dies setzt natürlich voraus, dass Sie ein entsprechendes neues Icon-Set für LibreOffice bereits selbst erstellt haben oder ein solches Set existiert, das sie in eine Extension verpacken wollen. Das Icon-Set selbst muss in der Form eines Zip-Archives vorliegen oder in ein solches gepackt werden.

4.1 Bauen der Icon-Set-Extension

Zunächst erstellen Sie im Grundverzeichnis der neuen Icon-Set-Extension ein Unterverzeichnis mit dem Namen „iconsets“. In dieses Unterverzeichnis kopieren Sie Ihr neues Icon-Set als Zip-Archiv. Des Weiteren erstellen Sie noch ein weiteres Unterverzeichnis mit dem Namen „icons“, in das Sie die Grafik im PNG-Dateiformat ablegen, die im Extension-Manager später angezeigt werden soll. Sinnvoller Weise sollte dies ein kleines Vorschaubildchen auf Ihr Icon-Set bzw. auf eine Auswahl Ihrer Icons sein, damit der Benutzer das Icon-Set visuell zuordnen kann.

In einem weiteren Unterverzeichnis, das Sie mit „description“ benennen können, erstellen Sie eine Textdatei im Dateiformat *.txt. In diese fügen Sie die Beschreibung Ihrer Icon-Set-Extension ein, die später im Extension-Manager-Dialog angezeigt wird. Sie sollte nicht zu lang sein, da sie ansonsten im Dialogfenster für den Benutzer unschön dargestellt wird.

Ein weiteres Unterverzeichnis mit dem Namen „META-INF“ legen Sie an, um dort die Datei „manifest.xml“ anzulegen. Diese Datei erhält folgenden Inhalt:

```
<?xml version="1.0" encoding="UTF-8"?>
<m:manifest xmlns:m="http://openoffice.org/2001/manifest">
  <m:file-entry m:media-type="application/
    vnd.sun.star.configuration-data"
    m:full-path="config.xcu"/>
</m:manifest>
```

In dieser Manifest-Datei wird auf eine weitere Datei mit dem Namen „config.xcu“ verwiesen. Diese erstellen Sie im Grundverzeichnis Ihrer Extension nun mit folgendem Inhalt:

```
<?xml version="1.0" encoding="UTF-8"?>
<oor:items xmlns:oor="http://openoffice.org/2001/registry">
  <item oor:path="/org.openoffice.Office.Paths/
    Paths/Iconset/InternalPaths">
```

4 Ein neues Icon-Set mit einer Extension ausliefern

```
<node oor:name="%origin%/iconsets" oor:op="fuse"/>
</item>
</oor:items>
```

In dieser Datei teilen Sie dem Extension-Manager mit, wo er das Icon-Set platzieren soll, und zwar in dem entsprechenden internen Pfad. Das Schlüsselwort „fuse“ bewirkt das Vereinigen von bisherigem Icon-Set mit Ihrem hinzugefügten Set von Icons.

Als nächstes müssen Sie noch eine Lizenz Ihrer Icon-Set-Extension hinzufügen, die später bei der Installation dem Benutzer angezeigt wird. Dazu erstellen Sie ein neues Unterverzeichnis „license“, in das Sie die Lizenzdateien im Dateiformat *.txt ablegen. Sie können hier eine Datei für jede von Ihrer Extension unterstützte Sprache hinterlegen, also beispielsweise „license_en.txt“ für die englische Version und „license_de.txt“ für die deutsche Version der Lizenz.

Zum Schluss erstellen Sie nun noch die Datei „description.xml“ im Grundverzeichnis. Diese hat beispielsweise folgenden Inhalt:

```
<?xml version='1.0' encoding='UTF-8'?>
<description
  xmlns="http://openoffice.org/extensions/description/2006"
  xmlns:dep="http://openoffice.org/extensions/description/2006"
  xmlns:lo="http://libreoffice.org/extensions/description/2011"
  xmlns:xlink="http://www.w3.org/1999/xlink">
  <identifier value="org.mycompany.myiconsetname"/>
  <version value="0.1"/>
  <display-name>
    <name lang="de">Meine Icon-Set-MeineExtension</name>
    <name lang="en">My Icon Set Extension</name>
  </display-name>
  <dependencies>
    <lo:LibreOffice-minimal-version
      d:name="LibreOffice 6.0" value="6.0"/>
  </dependencies>
  <publisher>
    <name xlink:href="http://amantke.de/blog">
      Andreas Mantke</name>
  </publisher>
  <icon>
    <default xlink:href="icons/iconsetname.png"/>
  </icon>
  <registration>
    <simple-license accept-by="admin">
      <license-text xlink:href="registration/license_de.txt"
        lang="de" />
      <license-text xlink:href="registration/license_en.txt"
        lang="en" />
    </simple-license>
```

4.1 Bauen der Icon-Set-Extension

```
</registration>
<extension-description>
  <src xlink:href="description/description.txt" />
</extension-description>
</description>
```

Eine Beschreibung der einzelnen XML-Tags der Datei „description.xml“ und ihrer Bedeutung finden Sie in den Abschnitten 2.1 und 2.5.

Zum Abschluss erstellen Sie im Grundverzeichnis Ihrer Extension noch eine Text-Datei mit dem Namen „extensionname.txt“, die den Namen der Extension enthält, hier z.B. den Text:

```
myiconsetextension.oxt
```

Danach fügen Sie noch das Build-Script aus dem Kapitel 2.3 in das Grundverzeichnis ein und lassen Ihre Extension durch dieses Skript wie dort beschrieben bauen.

Eine entsprechende Vorlage für eine LibreOffice Iconset-Extension finden Sie im Github-Repository unter folgender Adresse: https://github.com/andreama/extensionbook/tree/master/extensiontemplates/iconset_extension/iconsetextensionexample01.