

# THE LANGSELECT PACKAGE

vo.9.od 2025/08/10

Generierung verschiedener Sprachversionen aus einer Quelle

Matthias WERNER<sup>1</sup>

<https://github.com/tuc-osg/osglecture>

Dieses Paket unterstützt die Generierung von Dokumenten in mehreren Sprachvarianten aus einem gemeinsamen Quelldokument. Das Paket gehört zum osglecture-Bundle und wird in die osglecture-Klasse integriert und vom Buildsript `olm` unterstützt, kann aber auch eigenständig genutzt werden.

## Table of Contents

<b>1</b>	<b>Einleitung</b>	<b>1</b>	<b>3</b>	<b>Makros</b>	<b>4</b>
1.1	Alternativen . . . . .	1			
1.2	Sprachen . . . . .	2	<b>4</b>	<b>Beispiele</b>	<b>4</b>
<b>2</b>	<b>Anwendung</b>	<b>3</b>	<b>5</b>	<b>Implementation</b>	<b>4</b>
2.1	Auswahl der Zielsprache . . .	3			
2.2	Options . . . . .	4	<b>6</b>	<b>Anwendung</b>	<b>8</b>

## 1 Einleitung

Mitunter werden mehrere Sprachversionen des gleichen Dokuments benötigt. Im Anwendungsgebiet des osglecture-Bündel sind dies beispielsweise die Lehrskripte, Vorlesungsfolien und Handouts einer Lehrveranstaltung. Es fällt häufig schwer, mehrere parallel existierende Sprachversionen konsistent zu halten. Diese Aufgabe ist etwas einfacher, wenn alle Versionen in einem gemeinsamen Quelldokument enthalten sind.

Dieses Paket unterstützt die Generierung solcher Sprachversionen aus einem gemeinsamen  $\text{\LaTeX}$ -Quelldokument indem es Makros für Sprachvarianten und Mechanismen zur Auswahl der Zielsprache bereitstellt.

---

<sup>1</sup>. [matthias.werner@informatik.tu-chemnitz.de](mailto:matthias.werner@informatik.tu-chemnitz.de)

## 1.1 Alternativen

Es gibt eine Reihe von Alternativen zu langselect.

**Adhoc-Macros** Man kann relativ einfach adhoc Lösungen schaffen, wie z. B.

```

1      \newif\ifenglish
2      \englishtrue
3      \ifenglish
4          Welcome!
5      \else
6          Willkommen!
7      \fi
8

```

Tatsächlich folgt langselect grundsätzlich diesem Ansatz, automatisiert aber die Erstellung der Sprachmakros, vereinfacht die gemeinsame Nutzung von nichtsprachabhängigen Elementen wie z. B. Formeln und ermöglicht die Auswahl der Sprachvariante über verschiedene Methoden.

**multilang** Das ausgeklügelte Paket multilang<sup>2</sup> von Richard GREWE erstellt Sprachversionen von gegebenen Makros, wobei die aktive Sprache direkt aus Babel oder Polyglossia übernommen wird. langselect hat ein ähnliches Ziel, aber ein anderes (aus Sicht des Paketautoren: praktischeres) Interface. Falls Sie aber den Ansatz von langselect nicht mögen, sollten Sie sich unbedingt multilang anschauen.

**comment** Victor EIJKHOUT hat das Paket comment<sup>3</sup> geschrieben, das auf einfache Weise ermöglicht, nur bestimmte Abschnitte im Dokument auszugeben. Dies kann auch sehr gut für eine Sprachauswahl genutzt werden.

**translations/translator/xt\_capt** Diese drei Pakete translations<sup>4</sup>/translator<sup>5</sup>/xt\_capt<sup>6</sup> adressieren Programmierer von L<sup>A</sup>T<sub>E</sub>X-Paketen, um fixe Textelemente wie die Überschriften von Verzeichnissen zu internationalisieren. Für Autoren mehrsprachiger Dokumente sind diese Pakete nur bedingt geeignet.

## 1.2 Sprachen

Das Konzept von langselect beinhaltet die Nutzung von Sprachen auf verschiedenen Ebenen mit unterschiedlicher Wirkung. Um Verwechslungen zu vermeiden, wollen wir hier die Begriffe eindeutig definieren.

- 
- 2. on CTAN as multilang: <http://mirrors.ctan.org/macros/latex/contrib/multilang/>
  - 3. on CTAN as comment: <http://mirrors.ctan.org/macros/latex/contrib/comment/>
  - 4. on CTAN as translations: <http://mirrors.ctan.org/macros/latex/contrib/translations/>
  - 5. on CTAN as translator: <http://mirrors.ctan.org/macros/latex/contrib/translator/>
  - 6. on CTAN as xt\_capt: [http://mirrors.ctan.org/macros/latex/contrib/xtcapt/xt\\_capt/](http://mirrors.ctan.org/macros/latex/contrib/xtcapt/xt_capt/)

1. Die Menge *möglicher* Zielsprachen des Dokuments. Ein Dokument, das langselect nutzt, enthält typischerweise Abschnitte in verschiedenen Zielsprachen. Die möglichen Zielsprachen werden über eine Paketooption gesetzt, siehe Abschnitt 2.2. Im weiteren Text dieser Dokumentation nennen wir die möglichen Zielsprachen *Auswahlsprachen*, da für die  $\LaTeX$ -Übersetzung eine dieser Sprachen ausgewählt wird.
2. Die tatsächliche bei einer  $\LaTeX$ -Übersetzung gebrauchte Zielsprache. Diese muss eine der Sprachen aus 1. sein. Typischerweise wird der Übersetzungsvorgang für jede der tatsächlichen Zielsprachen einzeln ausgeführt. Für die Auswahl der tatsächlichen Zielsprache gibt es mehrere Methoden, siehe Abschnitt 2.1.
3. Die Sprache, in der Textteile oder einzelne Wörter tatsächlich geschrieben sind. Dies ist das Konzept von Sprache, das Pakete wie Babel oder Polyglossia haben.<sup>7</sup> Bitte beachten Sie, dass ein mehrsprachiges Dokument aus der Sicht von langselect ein monosprachiges Dokument aus der Sicht von Babel sein kann. Allerdings können durchaus noch weitere Sprachen hinzukommen: Beispielsweise kann ein Dokument mit den Auswahlsprachen Deutsch und Englisch im Text lateinische Wörter oder Sätze enthalten. Diese Auswahl wird über die üblichen Babelmechanismen wie `\selectlanguage`, `\foreignlanguage` oder `\text{<sprache>}` vorgenommen. Wenn Babel oder Polyglossia geladen wird, was auf Wunsch auch direkt über langselect erfolgen kann, wird als aktive Sprache die Zielsprache ausgewählt.

Sprachen werden in langselect über ISO 639-1-Codes (z. B. "en", "fr", "de", ...) bezeichnet. Wenn für die Sprachbehandlung auf der Ebene von Babel/Polyglossia ein bestimmter Dialekt oder eine bestimmte Variante gewünscht ist, kann ein Mapping erfolgen.

## 2 Anwendung

Das Paket wird auf die übliche Weise geladen:

```
\usepackage[<options>]{langselect}
```

Da die Auswahl der Zielsprache sowohl über Optionen als auch auf anderem Weg erfolgen kann, werden hier zunächst diese verschiedenen Wege beschrieben. Weitere Optionen wie das Setzen der Auswahlsprachen werden in Abschnitt 2.2 besprochen.

### 2.1 Auswahl der Zielsprache

Um eine Zielsprache für einen  $\LaTeX$ -Übersetzungslauf zu bestimmen, gibt es verschiedene Möglichkeiten. Die Reihenfolge in der folgenden Liste spiegelt auch die Reihenfolge der Auswertung. Sobald damit eine Zielsprache ermittelt wird, wird eine weitere Auswertung abgebrochen.

1. Definition eines Makros `\omdTargetLanguage` vor Laden des Pakets. Damit ist es beispielsweise möglich, über

---

<sup>7</sup> Wobei auch in diesen Paketen "Sprache" auch für mehrere unterschiedliche und z. T. voneinander unabhängige Konzepte wie Trennungsmuster, zu nutzende Fonts etc. steht

```
> latexmk -e"\def\olsTargetLanguage{fr}" document.tex
```

die Zielsprache Französisch für den Übersetzungslauf festzulegen.

2. Setzen der Paketoption `targetlang = {\langle Zielsprache \rangle}`
3. Wenn die Option `targetlang = {job=\langle n \rangle}` gesetzt wird, wird die Zielsprache aus dem  $n$ . Element des Jobnamens bestimmt. Ein Element ist ein mit "-" (Minus) abgetrennter Teil des Jobnamens, siehe die Dokumentation zum Paket `varsfromjobname`,<sup>8</sup> das dafür geladen wird. Beispielsweise kann bei `targetlang = {job=2}` über
 

```
> latex -jobname doc-ru doc.tex
```

 Russisch als Zielsprache eingestellt werden.
4. Mit Setzen von `targetlang = {meta}` wird die in `\DocumentMetaData` angegebene Sprache als Zielsprache verwendet.
5. Wenn alles andere versagt, wird Englisch als Zielsprache gesetzt.

## 2.2 Options

Es können folgende Optionen gesetzt werden:

`languages = {\langle Liste von Auswahl Sprachen \rangle}` (required)

Gibt zwei oder drei Auswahl Sprachen an. Die Reihenfolge ist relevant für die Generierung der Sprachmakros (siehe Abschnitt 3).

`targetlang = \langle Sprache \rangle | {job=\langle n \rangle} | meta` Default: meta

Siehe Abschnitt 2.1

`prefix = {\langle prefix \rangle}` Default: 1

Setzt einen Prefix für die Multisprachmakros. Es kann auch ein leerer Prefix gesetzt werden.

## 3 Makros

## 4 Beispiele

## 5 Implementation

```
\NeedsTeXFormat{LaTeX2e}[2022/06/01]
\def\packagename{langselect}
\def\packageversion{2025/08/10 v0.9.0d}
\ProvidesPackage{\packagename}[\packageversion\space support for different
languages versions of a document]
\ExplSyntaxOn
```

---

8. on CTAN as `varsfromjobname`: <http://mirrors.ctan.org/macros/latex/contrib/varsfromjobname/>

Im Moment brauchen wir noch Lua<sup>La</sup>TeX für die Sternvariante der Sprachmakros.

`\ExplSyntaxOn`

`\olsIsoTags` enthält alle gültigen ISO 639-1-Codetags.

```
\seq_const_from_clist:cn{\olsIsoTags}{aa,ab,ae,af,ak,am,an,ar,as,av,ay,az,ba,be,bg,
bh,bi,bm,bn,bo,br,bs,ca,ce,ch,co,cr,cs,cu,cv,cy,da,de,dv,dz,ee,el,en,eo,es,et,eu,
fa,ff,fi,fj,fo,fr,fy,ga,gd,gl,gn,gu,gv,ha,he,hi,ho,hr,ht,hu,hy,hz,ia,id,ie,ig,ii,
ik,io,is,it,iu,ja,jv,ka,kg,ki,kj,kl,km,kn,ko,kr,ks,ku,kv,kw,ky,la,lb,lg,li,ln,
lo,lt,lu,lv,mg,mh,mi,mk,ml,mn,mr,ms,mt,my,na,nb,nd,ne,ng,nl,nn,no,nr,nv,ny,oc,oj,
om,or,os,pa,pi,pl,ps,pt,qu,rm,rn,ro,ru,rw,sa,sc,sd,se,sg,si,sk,sl,sm,sn,so,sq,sr,
ss,st,su,sv,sw,ta,te,tg,th,ti,tk,tl,tn,to,tr,ts,tt,tw,ty,ug,uk,ur,uz,ve,vi,vo,wa,
wo,xh,yi,yo,za,zh,zu}
```

```
\newcommand\IfTagIsValidF[2]{%
  \seq_if_in:ceF{\olsIsoTags}{#1}{#2}
}
\RequirePackage{varsfromjobname}[2025/08/03]
```

Das Ergebnis von `\varsfromjobname` und von `\str_range:Nnn` hat Catcode 12. Wir brauchen einen Wrapper, damit wir einen einheitlichen Catcode (11) haben.

```
\newcommand\olsChangeCatEleven[1]{
  \def\tmpa{#1}
  \edef\tmpb{
    \noexpand\scantokens{
      \noexpand\edef\noexpand\olsCatElven{\tmpa}
    }
  }
  \tmpb
}
\DeclareKeys{
  languages.clist_gset:N = \ols@langs,
  languages.usage=load,
  targetlang.code = {
    \cs_if_exist:NF\olsTargetLanguage{
      \str_if_eq:eeTF{\str_range:Nnn{#1}{1}{3}}{job}{% targetlang =job n
        \olsChangeCatEleven{\getfromjobname{\str_range:Nnn{#1}{-1}{-1}}}
        \edef\olsTargetLanguage{\olsCatElven}
        % % \typeout{**** Set~olsTargetLanguage='\olsTargetLanguage'}
      }{
        \str_if_eq:nnTF{#1}{meta}{
          % %\typeout{**~SEARCH~FOR~META}
          \IfDocumentMetadataTF{
            \edef\tmpa{\GetDocumentProperties{document/lang}}
          }
        }
      }
    }
  }
}
```

## 5 Implementation

```

\edef\tmpb{\str_range:Nnn\tmpa{1}{2}}
\olsChangeCatEleven{\tmpb}
\typeout{*** Meta-lang: \GetDocumentProperties{document/lang},~**\tmpb}
\edef\olsTargetLanguage{\olsCatElven}
}{
\PackageWarningNoLine{\packagename}{No~meta~data~provided, falling
back to English.}
% %\edef\olsTargetLanguage{en}
}
}{
\edef\olsTargetLanguage{#1}
}
}
}{
},
targetlang.usage=load,
prefix.store=\ols@sprefix,
prefix.initial:n=l,
prefix.usage=load,
auto.bool_set:N=\ols@generate,
auto.usage=load,
auto.initial:n=true
}%
\ProcessKeyOptions\relax
\providecommand{\olsTargetLanguage}{en}
\typeout{*****~\string\olsTargetLanguage='\olsTargetLanguage'}

```

Wir führen verschiedene Checks durch. Als erstes prüfen wir, ob eine valide Zielsprache ermittelt werden konnte.

```

\IfTagIsValidF{\olsTargetLanguage}{
\PackageError{\packagename}{Couldn't~resolve~target~language~'\olsTargetLanguage'}{See
documentation.}
\aftergroup\endinput
}

```

Sind mindestens zwei Auswahl Sprachen angegeben?

```

\int_compare:nNt{\clist_count:N{\ols@langs}} < {2}{
\PackageError{\packagename}{No~sufficient~number~of~selectable~MessageBreak~languages~provided}{
You~have~to~provide~a~list~of~two~or~three~MessageBreak
languages~via~'language'~option.}
\aftergroup\endinput
}

```

Im Moment kann langselect nicht mehr als drei Auswahl Sprachen verarbeiten. Wir warnen daher, wenn mehr als drei Sprachen angegeben werden.

## 5 Implementation

```
\int_compare:nNtT{\clist_count:N\ols@langs} > {3}{
  \PackageWarningNoLine{\packagename}{Too~many~selectable~languages~provided.\MessageBreak
    I~will~ignore~the~superfluous~languages}
}{}
}
```

Als nächstes überprüfen wir, ob alle Auswahlssprachen gültige ISO 639-1 Codes sind.

```
\clist_map_inline:Nn\ols@langs {
  \typeout{***~CHECK~'#1'}
  \IfTagIsValidF{#1}{
    \PackageError{\packagename}{Couldn't~resolve~selectable~language~'#1'}{
      Use~valid~ISO-639-1~code~in~option~'languages'.}
    \aftergroup\endinput
  }
}
```

Die Zielsprache sollte in der Liste von Auswahlssprachen enthalten sein, sonst produzieren alle Sprachmakros nichts.

```
\clist_if_in:NVF\ols@langs{\olsTargetLanguage}{
  \PackageError{\packagename}{Target~language~'\olsTargetLanguage'~is~not~\MessageBreak
    in~the~list~of~selectable~languages}{
    Check~option~'language'~and~'targetlang'~.}
}
```

Damit es keine Konflikte gibt, überprüfen wir noch, ob eine ggf. existierende Spracheinstellung über `\DocumentMetadata` der Zielsprache entspricht

```
\IfDocumentMetadataT{
  \str_if_eq:eeF{\olsTargetLanguage}{\str_range:nnn{\GetDocumentProperties{document/lang}}{1}{2}}{
    \PackageWarningNoLine{\packagename}{Target~language~'\olsTargetLanguage'~doesn't~comply~with
      \MessageBreak\string\DocumentMetadata.~Prepare~for~trouble.}
  }
}
\bool_new:c{\ols@trilang}
\int_compare:nNtTF{\clist_count:N\ols@langs} > {2}{
  \bool_set_true:c{\ols@trilang}
}{
  \bool_set_false:c{\ols@trilang}
}
\def\olsFrstLanguage{\clist_item:Nn \ols@langs{1}}
\def\olsScndLanguage{\clist_item:Nn \ols@langs{2}}
\bool_if:cT{\ols@trilang}{
  \def\olsThrdLanguage{\clist_item:Nn \ols@langs{3}}
}
\typeout{***~First:~\olsFrstLanguage}
```

```

\typeout{***~Second:~\olsScndLanguage}
\NewDocumentCommand{\olsMakeBilingualMacro}{m m m}{
  \str_if_eq:eeT{\olsTargetLanguage}{#2}{
    \expandafter\long\expandafter\gdef\csname #1#2#3 \endcsname##1##2{##1}
  }
  \str_if_eq:eeT{\olsTargetLanguage}{#3}{
    \expandafter\long\expandafter\gdef\csname #1#2#3 \endcsname##1##2{##2}
  }
}
\NewDocumentCommand{\olsMakeTrilingualMacro}{m m m m}{
  \str_if_eq:eeT{\olsTargetLanguage}{#2}{
    \expandafter\long\expandafter\gdef\csname #1#2#3#4 \endcsname##1##2##3{##1}
  }
  \str_if_eq:eeT{\olsTargetLanguage}{#3}{
    \expandafter\long\expandafter\gdef\csname #1#2#3#4 \endcsname##1##2##3{##2}
  }
  \str_if_eq:eeT{\olsTargetLanguage}{#4}{
    \expandafter\long\expandafter\gdef\csname #1#2#3#4 \endcsname##1##2##3{##3}
  }
}
\bool_if:cT{\ols@generate}{
  \bool_if:cTF{\ols@trilang}{
    \olsMakeTrilingualMacro{\ols@sprefix}{\olsFrstLanguage}{\olsScndLanguage}{\olsThrdLanguage}
  }{
    \olsMakeBilingualMacro{\ols@sprefix}{\olsFrstLanguage}{\olsScndLanguage}
  }
}
\olsMakeBilingualMacro{\ols@sprefix}{\olsFrstLanguage}{\olsScndLanguage}
\ExplSyntaxOff

```

----- Ende Paketcode -----

## 6 Anwendung

Laden und verwenden:

```
\usepackage{mypkg}
```

Das war's.