

The komacv-multilang Package*

Richard Gay
richard.gay@t-online.de

August 30, 2017

1 Introduction

The goal of `komacv-multilang` is to simplify the specification and maintenance of a CV in multiple languages. More concretely, `komacv-multilang` enables one

- to specify translations for pieces of content such that they are located closely together and easily identifiable as belonging together; and
- to change the document’s language as simple as changing the `babel` or `polyglossia` language.

That is, with `komacv-multilang`, you can continuously expand and improve your CV in more than one language and when you actually want to use the CV, you choose a language and restrict the content to what you want to show.

The `komacv-multilang` package is realized as proxy to the `komacv` class. The latter package enables one to typeset CVs in one of three styles provided by the package or in a custom-made style. `komacv-multilang` uses the `multilang` package to define a set of macros and environments that establish a multilingual interface to `komacv`’s macros and L^AT_EX’s sectioning and list macros and environments. The macros and environments of `komacv-multilang` provide a proxy to all macros of `komacv` that produce CV entries. That is, with `komacv-multilang`, it should not be necessary to use `komacv` directly for CV entries, even though it is still possible to use `komacv` macros.

The remainder of this introduction provides an example based on `komacv`’s “casual” style. The `komacv-multilang` package is not limited to a specific style but bases on the macro interface provided by the core `komacv` class (i.e., macros like `\cvitem` and `\cventry`). The example shows a toy excerpt of a multilingual CV with `komacv-multilang` (Figure 1). For comparison, the same CV is also produced with pure `komacv` (Figure 2) and an ad-hoc approach to specifying the translations.

The example shows the document preamble, sectioning, a simple as well as a more involved CV entry, date range specification, and list entries. The example can be used for getting an intuition about `komacv-multilang` as well as for deciding whether to use `komacv-multilang` (or rather an ad-hoc approach as in Figure 2).

*This document corresponds to `komacv-multilang` v0.9, dated 2017/08/30. The package is available online at <http://www.ctan.org/pkg/komacv-rg> and <https://github.com/Ri-Ga/komacv-rg>.

Education	
date of birth	January 1, 1970
08/2010 – 07/2015	M.Sc. Computer Science, UCSB, California, USA, 4.00 GPA. <ul style="list-style-type: none"> ○ cryptography ○ software engineering

```

% in preamble
\documentclass[style=casual]{komacv}
\usepackage[english]{babel}
\usepackage[languages={english,german}]{komacv-multilang}
% in document
\begin{Section}{
  title/english      = Education,
  title/german       = Schulische Ausbildung}
  \BasicEntry{
    header/english   = {date of birth},
    header/german    = {Geburtsdatum},
    text/date        = {1970-01-01},
  }
  \EducationEntry{
    when/daterange   = {2010-08-01}{2015-07-31},
    degree/english    = M.Sc.{ } Computer Science,
    degree/german     = Master Informatik,
    institution       = UCSB,
    where/english     = {California, USA},
    where/german      = {Kalifornien, USA},
    grade/english     = 4.00 GPA,
    grade/german      = {Gesamtnote 1,0},
    details/list      = {
      \Item{
        value/english = {cryptography},
        value/german  = {Kryptographie},
      }
      \Item{
        value/english = {software engineering},
        value/german  = {Software-Engineering},
      }
    },
  },
}
\end{Section}

```

Figure 1: Multilingual CV with komacv-multilang

Education	
date of birth	January 1, 1970
08/2010 – 07/2015	M.Sc. Computer Science, UCSB, California, USA, 4.00 GPA. <ul style="list-style-type: none"> ○ cryptography ○ software engineering

```

% in preamble
\documentclass[style=casual]{komacv}
\usepackage[english]{babel}
\newcommand\inEnglish[1]{#1}
\newcommand\inGerman[1]{}
% in document
\section{\inEnglish {Education}%
\inGerman {Schulische Ausbildung}}
\cvitem{\inEnglish {date of birth}%
\inGerman {Geburtsdatum}}
{\inEnglish {January 1, 1970}%
\inGerman {1.{} Januar 1970}}
\cventry
{\inEnglish {08/2010 — 07/2015}
\inGerman {M.Sc.{} Computer Science}%
\inGerman {Master Informatik}}
{UCSB}
{\inEnglish {California, USA}%
\inGerman {Kalifornien, USA}}
{\inEnglish {4.00 GPA}%
\inGerman {Gesamtnote 1,0}}%
{\begin{compactitem}
\item\inEnglish {cryptography}%
\inGerman {Kryptographie}
\item\inEnglish {software engineering}%
\inGerman {Software–Engineering}
\end{compactitem}}

```

Figure 2: Multilingual CV with komacv

2 Usage

2.1 Package Loading

The `komacv-multilang` package is loaded like any other package. The only requirements to pay attention to are, firstly, `komacv-multilang` must be loaded after `babel` or `polyglossia`, and secondly, `komacv-multilang` must be loaded after the `komacv` class. This ensures that `komacv-multilang` can detect the document's language and that it can apply fixes to `komacv`.

`languages` The `komacv-multilang` package has one package option: `languages`. Through this option, the set of languages known to `komacv-multilang` is specified, as a comma-separated list. [Section 2.2](#) includes an example of loading `komacv-multilang`.

2.2 Sections

The `komacv-multilang` package utilizes `multilang-sect` for sectioning environments. Under the `komacv` class, these environments produce sections in the respective loaded `komacv` style. The following documentation essentially repeats what is also stated in the `multilang` documentation.

```
\begin{Section}{<data>}
```

```
\end{Section}
```

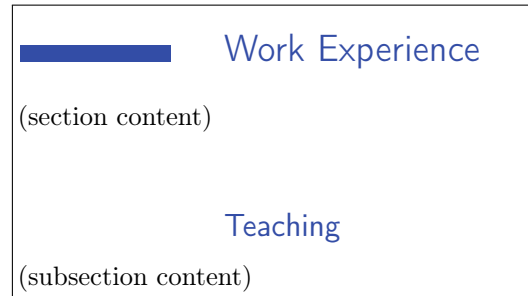
This environment shows a section. It has a single, mandatory argument, named `title`. It is a disableable environment, i.e., the argument `disabled` can be used in `<data>` to disable the display of the whole section. This environment acts as a proxy for the `\section` macro as it is used by `komacv` (i.e., without optional argument and without the star).

```
\begin{SubSection}{<data>}
```

```
\end{SubSection}
```

This environment is analogous to the `Section` environment, just for sub-sections. Examples:

```
% in preamble
\documentclass[style=casual]{komacv}
\usepackage[english]{babel}
\usepackage[languages={english,german}]
    {komacv-multilang}
% in document
\begin{Section}{
    title/english = Work Experience,
    title/german = Berufserfahrungen,
}
(section content)
\begin{SubSection}{
    title/english = Teaching,
    title/german = Lehre,
}
(subsection content)
\end{SubSection}
\end{Section}
```



The example shows several aspects: (1) how the `komacv-multilang` package is loaded and how its `languages` option is used and (2) how the `Section` and `SubSection` environments can be used.

2.3 CV Entries

The `komacv` provides a set of macros for typesetting CV entries. In the following, we introduce the proxies for these macros provided by `komacv-multilang`.

`\BasicEntry{⟨data⟩}`

This macro has two mandatory arguments, named `header` and `text`, and one optional argument, named `spacing`. The argument alias `both` can be used to specify the `header` and `text` simultaneously. It is a disableable macro, i.e., the argument `disabled` can be used in `⟨data⟩` to disable the display of the entry. This macro acts as a proxy for `komacv`'s `\cvitem` macro. That is, the value of the `header` argument is displayed in the hint column and the `text` is displayed in the main column. The value of the `spacing` argument specifies the vertical spacing below the CV entry. Examples:

```
\BasicEntry{
  header/english = date of birth,
  header/german  = Geburtsdatum,
  text/date      = 1970-01-01,
}
\BasicEntry{disabled,
  both/english   = {marital status}{single},
  both/german    = {Familienstand}{ledig},
}
```

date of birth	January 1, 1970
---------------	-----------------

The example shows several aspects: (1) how translated values for the arguments `header` and `text` are specified, (2) how a date can be specified (in the `yyyy-mm-dd` format) for a localized display, (3) how the display of an entry can be disabled, (4) how the `both` argument can be used, and (5) how the \LaTeX source code can be aligned such that it is easily readable despite the presence of translations.

`\CommentedEntry{⟨data⟩}`

This macro has three mandatory arguments, named `header`, `text`, and `comment`, as well as one optional argument named `spacing`. The argument alias `all` can be used to specify `header`, `text`, and `comment` simultaneously. It is a disableable macro, i.e., the argument `disabled` can be used in `⟨data⟩` to disable the display of the entry. This macro acts as a proxy for `komacv`'s `\cvitemwithcomment` macro. That is, the value of the `header` argument is displayed in the hint column, the `text` is displayed in the main column and the `comment` is displayed right-aligned in the main column. The value of the `spacing` argument specifies the vertical spacing below the CV entry. Examples:

```
\CommentedEntry{
  all/english   = {English}{fluent}
                {(oral and written)},
  all/german    = {Englisch}{flie{\ss}end}
                {(in Wort und Schrift)}
}
\CommentedEntry{
  header/english = German,
  header/german  = Deutsch,
  text/english   = mother tongue,
  text/german    = Muttersprache,
  comment        = {},
}
```

English	fluent	(oral and written)
German	mother tongue	

The example shows several aspects: (1) how the `all` argument alias can be used, (2) how arguments can be broken into multiple lines, (3) how translated values for the arguments `header`, `text`, and `comment` are specified, and (4) how an argument can be left empty in all languages (`comment`).

`\DoubleEntry{⟨data⟩}`

This macro has four mandatory arguments, named `header1`, `text1`, `header2`, and `text2`, as well as one optional argument named `spacing`. The argument aliases `first` and `second` can be used to specify `header1`, and `text1`, respectively, `header2` and `text2` simultaneously. It is a disableable macro, i.e., the argument `disabled` can be used in `⟨data⟩` to disable the display of the entry. This macro acts as a proxy for `komacv`'s `\cvdoubleitem` macro. That is, the value of the `header1` argument is displayed in the hint column, and the values of `text1`, `header2`, and `text2` are displayed in a columned fashion in the main column. The value of the `spacing` argument specifies the vertical spacing below the CV entry. Example:

<pre>\DoubleEntry{ first = {h1}{t1}, second = {h2}{t2}, }</pre>	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%; padding: 5px;">h1</td> <td style="width: 25%; padding: 5px;">t1</td> <td style="width: 25%; padding: 5px;">h2</td> <td style="width: 25%; padding: 5px;">t2</td> </tr> </table>	h1	t1	h2	t2
h1	t1	h2	t2		

The example shows how the `first` and `second` argument aliases can be used. Further possibilities for specifying the arguments, including translations, are analogous to the cases of `\BasicEntry` and `\DoubleEntry`.

`\EducationEntry{⟨data⟩}`

This macro has five mandatory arguments, named `when`, `degree`, `institution`, `where`, and `grade`, as well as two optional arguments, named `spacing` and `details`. It is a disableable macro, i.e., the argument `disabled` can be used in `⟨data⟩` to disable the display of the entry. This macro acts as a proxy for `komacv`'s `\cventry` macro. That is, the value of the `where` argument is displayed in the hint column, whereas the values for the other arguments are displayed in the main column (in the order they are listed). The value of the `spacing` argument specifies the vertical spacing below the CV entry. Example:

<pre>\EducationEntry{ when/daterange = {2010-08-01}{2015-07-31}, degree/english = M.Sc.{} Computer Science, degree/german = Master Informatik, institution = UCSB, where/english = {California, USA}, where/german = {Kalifornien, USA}, grade/english = 4.00 GPA, grade/german = {Gesamtnote 1,0}, details = {(e.g., honors, awards)}, }</pre>	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%; padding: 5px;">08/2010 – 07/2015</td> <td style="width: 70%; padding: 5px;"> M.Sc. Computer Science, <i>UCSB</i>, California, USA, 4.00 GPA. (e.g., honors, awards) </td> </tr> </table>	08/2010 – 07/2015	M.Sc. Computer Science, <i>UCSB</i> , California, USA, 4.00 GPA. (e.g., honors, awards)
08/2010 – 07/2015	M.Sc. Computer Science, <i>UCSB</i> , California, USA, 4.00 GPA. (e.g., honors, awards)		

The example shows a few notable aspects: (1) how the arguments can be used, (2) how a date range can be specified (for the `when` argument), and (3) how curly braces can be used to deal with commas in argument values (for the `where`, `grade`, and `details` arguments).

`\EmploymentEntry{⟨data⟩}`

This macro has four mandatory arguments, named `when`, `title`, `employer`, and

where, as well as two optional arguments, named `spacing` and `description`. It is a disableable macro, i.e., the argument `disabled` can be used in `<data>` to disable the display of the entry. This macro acts as a proxy for `komacv's \cventry` macro, as also `\EducationEntry`. The `\EmploymentEntry` specializes to detailed entries about professional/work experiences. Example:

```
\EmploymentEntry{
  when/daterange = {2015-08-01}{*},
  title/english  = Software engineer,
  title/german   = Softwareentwickler,
  employer       = XYZ Development,
  where/english  = {California, USA},
  where/german   = {Kalifornien, USA},
  description    = {Java, JavaScript},
}
```

```
08/2015 – Software engineer,
today    XYZ Development, Cali-
         fornia, USA.
         Java, JavaScript
```

`\AchievementEntry{<data>}`

This macro has two mandatory arguments, named `when` and `title`, as well as four optional arguments, named `spacing`, `occasion`, `where`, and `description`. It is a disableable macro, i.e., the argument `disabled` can be used in `<data>` to disable the display of the entry. This macro acts as a proxy for `komacv's \cventry` macro, as also `\EducationEntry` and `\EmploymentEntry`. The `\AchievementEntry` specializes to entries about achievements. Example:

```
\AchievementEntry{
  when/date      = 2016-05-01,
  title/english  = employee of the month,
  title/german   = Mitarbeiter des Monats,
  where          = XYZ Development,
}
```

```
May 1, 2016 employee of the
             month, XYZ Develop-
             ment.
```

`\EntryListItem{<data>}`

This macro has one mandatory argument, `item`, and one optional argument, `label`. It is a disableable macro, i.e., the argument `disabled` can be used in `<data>` to disable the display of the entry. This macro acts as a proxy for `komacv's \cvlistitem` macro.

`\EntryListDbItem{<data>}`

This macro has two mandatory arguments, `first` and `second`, and one optional argument, `spacing`.¹ The argument alias `both` can be used to specify `first` and `second` simultaneously. It is a disableable macro, i.e., the argument `disabled` can be used in `<data>` to disable the display of the entry. This macro acts as a proxy for `komacv's \cvlistdoubleitem` macro. Examples:

```
\BasicEntry{
  header/english = programming,
  header/german  = Programmiersprachen,
  text/english   = {I am proficient in:},
  text/german    = {Ich beherrsche:}}
\EntryListDbItem{both = {Java}{C/C++}}
\EntryListItem {item = PHP}
```

```
programming I am proficient in:
              o Java           o C/C++
              o PHP
```

¹Note that contrary to the `komacv` documentation, the optional argument to `\cvlistitem` and `\cvlistdoubleitem` is also the `spacing` and not a label.

2.4 Dates and Lists

The `komacv-multilang` package defines several datatypes, such as `date` and `daterange`, which are illustrated multiple times in [Section 2.3](#). The following is a complete list of all defined datatypes.

date: This datatype expects a date in the format `yyyy-mm-dd` and displays it in the regional format for dates.

daterange: This datatype expects two dates in the format `yyyy-mm-dd` and displays them in the same format as `shortdate`, with a “-” in between. If the second date is “*”, the text “today” (in the selected language) is displayed.

shortdate: This datatype expects a date in the format `yyyy-mm-dd` and displays it in the format “mm/yyyy”.

list: This datatype expects a list of items and displays the items in a compact form. Items can be specified with the standard `\item` macro as well as with the multilingual `\Item{<data>}` macro, whose `<data>` parameter expected one mandatory argument, `value`. The following example shows both possibilities.


```
\BasicEntry{
  header/english = programming languages,
  header/german  = Programmiersprachen,
  text/list      = {
    \item PHP
    \Item{value   = Java}
    \Item{
      value/english = C and C++,
      value/german  = C und C++}}}
```

programming	○	PHP
languages	○	Java
	○	C and C++

2.5 Tags

The `komacv-multilang` package utilizes `multilang-tags` for tag support. Tag support enables one to easily switch between granularities and focuses of a CV. The following just provides an example from the context of typesetting CVs. For more information, please consult the `multilang` documentation.

```
\DefineTagFilter{Programmer}{accept}{
  accept=PLDetails, deny=OfficeDetails}
\DefineTagFilter{Consultant}{accept}{
  accept=OfficeDetails, deny=PLDetails}
\UseTagFilter{Consultant}
\begin{Section}{
  title/english = Skills,
  title/german  = Qualifikationen}
\BasicEntry{
  header/english = Programming,
  header/german  = Programmiersprachen,
  text/list      = {
    \Item{value   = {Java, C/C++}}
    \Item{tags    = PLDetails,
      value       = {\LaTeX, Bash}}}}
\BasicEntry{header = Software,
  text/list      = {
    \Item{value   = MS Office}
    \Item{tags    = OfficeDetails,
      value       = Adobe InDesign}}}}
\end{Section}
```

	Skills
Programming	○ Java, C/C++
Software	○ MS Office
	○ Adobe InDesign

3 Implementation

The `multilang` package provides the core feature of `komacv-multilang`: defining multilingual macros and environments. We load the package with the same options as `komacv-multilang`. That is, the `languages` option of `komacv-multilang` is passed on to `multilang`. We also utilize the `multilang` extensions for tags and sectioning macros.

```
1 \RequirePackageWithOptions{multilang}
2 \RequirePackage{multilang-tags}
3 \RequirePackage{multilang-sect}
```

For the date datatypes (Section 3.1.1), we use the following packages:

```
4 \PassOptionsToPackage{useregional}{datetime2}
5 \RequirePackage{datetime2,datetime2-calc}
6 \RequirePackage{translations}
```

3.1 Datatypes

We define the datatypes before we define the multilingual macros and environments, such that the datatypes are available in the macros and environments.

3.1.1 Date Formatting

First, we define the three datatypes. The code largely speaks for itself.

```
7 \NewMultilangType{date}{\DTMdate{#1}}
8 \NewMultilangType{shortdate}{%
9   \DTMsetdatestyle{Myyyy}\DTMdate{#1}\DTMsetdatestyle{default}}
10 \NewMultilangType[2]{daterange}{%
11   \DTMsetdatestyle{Myyyy}\DTMdate{#1} --
```

If the second argument to `daterange` equals “*”, then the text “today” (in the translation for the current language) is displayed rather than a date.

```
12   \ifstrequal{#2}{*}%
13     {\GetTranslation{today}}%
14     {\DTMdate{#2}\DTMsetdatestyle{default}}
```

Next, we define the `Myyyy` date format, which we use for the `shortdate` and `daterange` style.

```
15 \DTMnewdatestyle{Myyyy}{%
16   \def\DTMdisplaydate##1##2##3##4{\DTMtwdigits{##2}/##1}}
```

Finally, we provide a small set of initial translations for the “today” in `daterange`.

```
17 \DeclareTranslation{english}{today}{today}
18 \DeclareTranslation{german}{today}{heute}
19 \DeclareTranslation{french}{today}{aujourd’hui}
```

3.1.2 List Formatting

First, we define the list type, `list`.

```
20 \NewMultilangType{list}{\kcvml@list{#1}}
```

`\kcvml@list` For the list datatype, the `\kcvml@list{<items>}` macro formats an itemized list of multiple items. The items of the list are specified by *<items>*. The items can be specified with `\item` macros as well as with multilingual `\Item` macros (i.e., `\kcvml@listitem`).

```
21 \newcommand\kcvml@list[1]{%
```

To ensure a proper spacing, we use `komacv`'s `compactitem` environment and encapsulate this environment inside a `minipage`. Only inside this environment, we enable the `\Item` macro.

```
22 \begin{minipage}[t]{\linewidth}%
```

```
23 \let\Item=\kcvml@listitem
```

```
24 \compactitem
```

```
25 #1%
```

For the case that the last line of the last list item has a small depth, the `minipage` might yield a bad spacing to content following the list. Hence, we add a `\strut` here.

```
26 \ifhmode\unskip\strut\fi
```

```
27 \endcompactitem
```

```
28 \end{minipage}}
```

`\kcvml@listitem` The `\kcvml@listitem{<data>}` enables multilingual items as a proxy to the `\item` macro. We treat `\item` here as a macro with one argument, as it usually does not harm to write “`\item{text}`” rather than “`\item text`”.

```
29 \NewMultilangCmd{\kcvml@listitem}{disablable,
```

```
30 command=\item, margs=value}
```

3.2 CV Entries

`\BasicEntry` The `\BasicEntry{<data>}` is a proxy to the `\cvitem` macro of `komacv`.

```
31 \NewMultilangCmd{\BasicEntry}{disablable,
```

```
32 command=\cvitem,
```

```
33 oargs={spacing},
```

```
34 margs={header,text},
```

```
35 alias/both={header,text},
```

```
36 }
```

`\CommentedEntry` The `\CommentedEntry{<data>}` is a proxy to the `\cvitemwithcomment` macro of `komacv`.

```
37 \NewMultilangCmd{\CommentedEntry}{disablable,
```

```
38 command=\cvitemwithcomment,
```

```
39 oargs={spacing},
```

```
40 margs={header,text,comment},
```

```
41 alias/all={header,text,comment},
```

```
42 }
```

`\DoubleEntry` The `\DoubleEntry{<data>}` is a proxy to the `\cvdoubleitem` macro of `komacv`.

```
43 \NewMultilangCmd{\DoubleEntry}{disablable,
```

```
44 command=\cvdoubleitem,
```

```
45 oargs={spacing},
```

```
46 margs={header1,text1,header2,text2},
```

```
47 alias/first={header1,text1},
```

```
48 alias/second={header2,text2},
49 }
```

`\EducationEntry` The `\EducationEntry{<data>}` is a proxy to the `\cventry` macro of `komacv`, specialized for education-related entries.

```
50 \NewMultilangCmd{\EducationEntry}{disablable,
51   command=\cventry,
52   oargs={spacing},
53   margs={when,degree,institution,where,grade,details},
54   defaults={details={}},
55 }
```

`\EmploymentEntry` The `\EmploymentEntry{<data>}` is a proxy to the `\cventry` macro of `komacv`, specialized for main work-related entries.

```
56 \NewMultilangCmd{\EmploymentEntry}{disablable,
57   command=\cventry,
58   oargs={spacing},
59   margs={when,title,employer,where,@grade,description},
60   defaults={@grade={}, description={}},
61 }
```

`\AchievementEntry` The `\AchievementEntry{<data>}` is a proxy to the `\cventry` macro of `komacv`, specialized for achievements.

```
62 \NewMultilangCmd{\AchievementEntry}{disablable,
63   command=\cventry,
64   oargs={spacing},
65   margs={when,title,occasion,where,@grade,description},
66   defaults={@grade={}, occasion={}, where={}, description={}},
67 }
```

`\EntryListItem` The `\EntryListItem{<data>}` is a proxy to the `\cvlistitem` macro of `komacv`.

```
68 \NewMultilangCmd{\EntryListItem}{disablable,
69   command=\cvlistitem,
70   oargs={spacing},
71   margs={item},
72 }
```

`\EntryListDbItem` The `\EntryListDbItem{<data>}` is a proxy to the `\cvlistdoubleitem` macro of `komacv`.

```
73 \NewMultilangCmd{\EntryListDbItem}{disablable,
74   command=\cvlistdoubleitem,
75   oargs={spacing},
76   margs={first,second},
77   alias/both={first,second},
78 }
```

Change History

v0.1
General: Initial version
(unpublished) 1

v0.9
General: Complete rewrite, splits
off multilang package 1

Index

A
\AchievementEntry 7, [62](#)

B
\BasicEntry 5, [31](#)
\begin 22

C
\CommentedEntry 5, [37](#)
\compactitem 24
\cvdoubleitem 44
\cventry 51, 57, [63](#)
\cvitem 32
\cvitemwithcomment 38
\cvlistdoubleitem 74
\cvlistitem 69

D
\DeclareTranslation . 17, 18, 19
\DoubleEntry 6, [43](#)
\DTMdate 7, 9, 11, 14
\DTMdisplaydate 16
\DTMnewdatestyle 15
\DTMsetdatestyle 9, 11, 14
\DTMtwodigits 16

E
\EducationEntry 6, [50](#)
\EmploymentEntry 6, [56](#)
\end 28
\endcompactitem 27
\EntryListDbItem 7, [73](#)
\EntryListItem 7, [68](#)

F
\fi 26

G
\GetTranslation 13

I
\ifhmode 26
\ifstrequal 12
\Item 23
\item 30

K
\kcvml@list 20, [21](#)
\kcvml@listitem 23, [29](#)

L
\let 23
\linewidth 22

N
\NewMultilangCmd . 29, 31, 37, 43,
50, 56, 62, 68, 73
\NewMultilangType .. 7, 8, 10, 20

P
\PassOptionsToPackage 4

R
\RequirePackage 2, 3, 5, 6
\RequirePackageWithOptions . 1

S
\strut 26

U
\unskip 26