

# The **philex** Package

## version 1.01

Peter Pagin  
peter.pagin@philosophy.su.se

September 22, 2012, 20:34

### Changes since version 1.0

- Bugfix: eliminated unwanted space after `\rn{}` cross-references.

### Changes since version 0.9 (internal version)

- Adapted the package to version 4.1 of `linguex`.
- Added two commands for label separation.

### Changes since version 0.6

- Added support for `hyperref` for named environments.
- Added naming options for sub-environments (`\lba`, `\lbb`, `\lbz`).
- Added customizing of brackets and numbering-style for second-level sub-environments.
- Corrected errors in vertical spacing for `\lbu` and `\rff`.
- Corrected an error in the centering option, and added an alternative for centering.
- Replaced earlier (overly short) punctuation commands, and added a package option for using the old commands.
- Added customizing of vertical spacing and horizontal spacing as well as label length.
- Made the `\lbp` environments labels more customizable.
- Rewrote and reorganized the manual.

### Changes since version 0.5

Corrected vertical spacing between adjacent `\lb` and `\lbp` environments.

## Changes since version 0.3

In version 0.3, forward cross-reference was effected by means of writing command definitions to a separate file, *myfile-px.tex*, which was subsequently read-in at the beginning of the following run of typesetting. In version 0.5 the information is instead written to the .aux file and then read from the .aux file in the subsequent run. This method is more elegant and eliminates problems with the implementation of the old method. The improvement is due to Robin Fairbairns.

## Manual

The **philex** package is intended for use in topics such as philosophy and linguistics, where there is a need both of many example sentences and of named principles, and of cross-references to them in the text. Sometimes one wants cross-reference to environment numbers (or prefix plus number), sometimes to a non-numbered named environment, and sometimes to the content of the environment, for repeating the content in running text or in another environment. The package is a small addition to Wolfgang Sternefeld's **linguex** package and requires **linguex**. As of version 0.2 it also requires **calc**. **philex** will load **linguex** if it is installed. The purpose of **philex** is to add functions for more flexible cross-referencing, for repeating and embedding named or numbered environments, **hyperref** support and a some further formatting options.

As of version 0.5 **philex** writes cross-ref information to the .aux file and reads that information at the beginning of the subsequent run of typesetting. This allows forward cross-referencing to environments that are named or numbered+named, such as (PP) on p. 6, in analogy to the label-reference system of LaTeX itself. As of version 0.9 hyperlinks produced by **hyperref** are created also for named environments with the **<hyper>** package option.

## 1 Calling the package

**philex**

**philex** is called by adding `\usepackage[<package options>]{philex}` in the preamble. This will also automatically load the **linguex** package. If **linguex** is called separately, it should be called *before* **philex**, since **philex** (marginally) redefines some commands in **linguex**.

## 2 Package options

**<hyper>**

With the **<hyper>** option hyperlinks will be created by cross-references to named environments (such as `\lbp`). This requires that the **hyperref** package is called (should be called last). If **hyperref** is called without the

<hyper> option, hyperlinks will anyway be created from the usual `philex` cross-references that make use of the basic label- and cross-reference function in LaTeX.

<oldpunct>

With the <oldpunct> package option the old punctuation commands (see below) will still work. This is for the purpose of using more recent versions of `philex` with older files, without needing search-and-replace.

### 3 Top-level environments

\lb

The basic environment command of `philex` is `\lb{}{}`. It takes two obligatory arguments. The first argument is the label of the environment, and the second is the content of the environment itself. `lb` uses the `\ex.` formatting of `linguex`, and thereby the `ExNo` counter. Type

```
\lb{nice}{This is a nice day. \\ Not too hot.}
```

and the result will be

(1) This is a nice day.  
Not too hot.

in case it is the first `\lb`-environment in your document.

In `linguex` you leave a blank line after sentence to close the environment. The blank line gives a `\par`-command to TeX. In `philex` the `\par`-command is built into the top-level environments. So you leave a blank line after an `\lb{}{}`-environment only if you want to start a new paragraph. The reason for this change is that only this way will the `philex` package work together with the `extract` package by Hendri Adriaens, which can be used for extracting material, e.g. for generating a handout, from documents making use of `philex`. `linguex` comes together with Sternefeld's `linguho` package, which has the function of generating a handout from documents using `linguex`. `linguho` does not work with `philex`, since the `linguex` environments are not explicit in the document. The corresponding function, only more flexible, can be served with the `extract` package.

Update with \lbu

In many cases you will want to produce a variant of the original sentence, which will have the same name or number as the original, but with an added suffix, like a letter or a prime symbol. This is achieved with the `\lbu{}{}{}{}`-command. It takes four obligatory arguments. The first is the label of the new sentence, the second is the label of the environment you are giving a variant of, the third is the suffix you want to add, and the fourth is the new content itself.

```
\lbu{nicy}{nice}{\(')}{But that yesterday was even nicer.}
```

produces

(1') But yesterday was even nicer.

Named environ-  
ments with `\lbp`

You may also want to have named principles, such as in (W) below (exemplifying forward cross-reference)

(W) Life is wonderful!

These are produced with the `\lbp{}{}{}-command` which makes use of the optional argument to `\ex. []` in `linguex`. If you later change the name of the principle, or the formulation, the changes are reflected in the cross-references and repetitions after the next typesetting, as long as you do not change the labels.

And updates with `\lbu`, like

(W<sup>+</sup>) Life is terrific!

work as before.

Prefixed numbering  
with `\lbpx`

In case you wish a list of independently numbered principles with a shared name stem, there are four independent series. Use one of the `\lbpx-commands`, where `x` is one of `a, b, c, d`. You can format the listing with `\lbpxformat{}{}{}-command`, where again `x` has the corresponding value. The first argument sets the numbering style (see below), the second the prefix and the third argument the suffix (this is a change since version 0.6, where there was no suffix argument, and general bracket commands were used). Then the `\lbpx-command` picks it up:

```
\bpaformat{1}{(T){}}
\lbpa{kno}{John is in the know.}
\lbpa{hu}{Elsa is, too.}
\lbpa{kne}{Alfred agrees.}
```

produces

(T1) John is in the know.  
(T2) Elsa is, too.  
(T3) Alfred agrees.

These enumerations are independent of `ExNO` and of each other, and the counters (`bpx`) are reset with the relevant `\lbpxformat`. They still work with sub-clauses, cross-references and the rest of the apparatus. Note that these enumerations do not have to be contiguous:

(T4) But he shouldn't.

Centering

All the top-level environments take an optional argument `[c]` for centering the text. `\lb[c]{compo}{\[\mu(\sigma(t_{1}), \ldots, t_{n})) = r_{\sigma}(\mu(t_{1}), \ldots, \mu(t_{n}))\]}` produces

$$(2) \quad \mu(\sigma(t_1, \dots, t_n)) = r_{\sigma}(\mu(t_1), \dots, \mu(t_n))$$



empty. You can redefine it to provide a full stop by means of `\philfullstop` before the environment. The command `\philcomma` redefines it to produce a comma, `\philexclaim` produces an exclamation mark, `\philquestion` a question mark, and `\philempty` turns it back to empty. By means of `\renewcommand{\philpunct}{}` you can set it as you like.

In earlier versions of `philex` the corresponding commands were `\p`, `\s` (for full stop), `\km` (for comma) and `\q` (empty). Mostly because I have learned that it is bad design to use one-letter commands, these are now replaced by the new commands. In case of typesetting a file with the old commands, choose the package option `<oldpunct>`.

Footnotes `linguex` has sensitivity to footnote contexts, and provides a separate counter, `FnExNo`, that is reset in each footnote. This is inherited by `philex`.<sup>1</sup>

## 4 Sub-environments

`\lba`, `\lbb`, and `\lbz` `linguex` and hence also `philex` provides sub-environments (sublists) . To illustrate, the code

```
\lbp{clauses}{PP}{Some main words, followed by
\lba{first}{Time flies
\lba{firstnew}{Like an arrow}
\lbz{lastnew}{And much too fast}}
\lbb{second}{But never stops}
\lbz{last}{Which is lucky}
and a concluding comment.}
```

produces

```
(PP) Some main words, followed by
  a. Time flies
      (i) Like an arrow
      (ii) And much too fast
  b. But never stops
  c. Which is lucky
and a concluding comment.
```

As shown in the source example, the sub-environments must be embedded in the last argument of the top-level command. A second-level sub-environment need not be embedded in the first-level sub-environment (and so on).

---

<sup>1</sup>The result is illustrated by

- (i) A first footnote example:
- (ii) A second footnote example.

which shows that numbering is lowercase roman.

As exemplified, sub-environments can be nested up to three levels. The first two levels have their own counters, `SubExNo` and `SubSubExNo`, provided by `linguex`.

The `\lba` command restarts the counter of the relevant level. The `\lbb` command steps the counter, while the `\lbz` command steps the counter and closes the level, returning to the next higher level. It is not necessary to use `\lbz` for the last item at the first sub-level (i.e. at the end of a top-level environment). In case only a single sub-environment is used, the command `\z.` of `linguex` returns to the higher level.

The labels of the first two sub-levels can be customized in two ways, with respect to numbering style and with respect to brackets. The command `\subformat` takes three arguments.

The first argument determines the numbering style. The value ‘R’ displays the counter in upper-case roman numerals, ‘r’ in lower case roman, ‘1’ in arabic numerals, ‘A’ in upper-case alphabetic listing, and ‘a’ in lower-case alphabetic listing. Default for the first sub-level is lower-case alphabetic.

The second argument sets the prefix (opening bracket) and the third argument sets the suffix (closing bracket). Default for the first sub-level is no left-hand bracket and a period for the suffix, as shown in the example.

The corresponding command for the second sub-level is `\subsubformat`, with the same arguments and values. Default for the second sub-level is lower-case roman numerals for displaying the counter, and ordinary left- and right-hand round brackets, as shown in the example (these defaults are set by `linguex`). To exemplify, the source code

```
\lbu{clausesup}{clauses}{${}$}{
\subformat{A}{-}{-}}
\subsubformat{1}{[[]]}
Some introductory words, followed by
\lba{firstb}{Time flies}
\lba{firstnewb}{Like an arrow}
\lbz{lastnewb}{And much too fast}
\lbb{secondb}{But never stops}
\lbz{lastb}{Which is lucky}
and a concluding comment.}
```

produces

```
(PP') Some introductory words, followed by
  A) Time flies
    [1] Like an arrow
    [2] And much too fast
  B) But never stops
  C) Which is lucky
and a concluding comment.
```

Names for sub-environments

If the sub-formatting commands are given within the top-level environment, they control only the labels within that environment. If they are given outside, the commands are valid until superseded by later commands.

The sub-level commands have an optional argument (inherited from `linguex`) that can be used for naming sub-environments. For instance, the source

```
\lb{crane}{The saying
\lba[S]{squeak}{The squeaky wheel gets the grease}
\z.
is often confirmed.}
```

produces

```
(4)   The saying
      (S) The squeaky wheel gets the grease
      is often confirmed.
```

Punctuation for sub-environments

Round brackets are put in by default. This can be reset by the `\philbrackets` command.

Punctuation for sub-environments is controlled by means of the commands `\philsubpunct`, `\philsubstop`, `\philsubcomma`, `\philsubexclaim`, `\philsubquestion`, and `\philsubempty`, analogous to the corresponding top-level commands. The corresponding older commands where `\pt`, `\stp` (full stop), `\kmt` (comma), and `\qt` (empty). As before, they are made active by means of the `<oldpunct>` package option.

## 5 Cross-references

The first argument of all environment commands are used by `philex` to give a `\label` command. If the environment is numbered by a counter, the label makes the environment accessible for cross-reference by the standard cross-reference mechanism of LaTeX. For instance, `\ref{nice}` produces the cross-reference ‘1’ to the first `\lb` environment on page 3.<sup>2</sup>

`\rf`, `\rn`

However, `philex` provides its own cross-reference command, `\rf`, which has a wider function. It provides ordinary cross-references in accordance with the numbering style of the antecedent, and keeps track of the embedding. It also provides cross-references to *named* environments, where there is no counter value to pick up. The difference between `\rf` and `\rn` is that the latter leaves out the brackets. So we will have the productions

```
\rf{nice} → (1) (target on page 3)
```

<sup>2</sup>The reason for the fuss about brackets is that in order to modify the material within the brackets, e.g. with `\bu`, the brackets cannot be hard-coded into the cross-reference command, but must be put in afterwards.



	<code>\rf{kne} → ((T3))</code>	(target on page 4)
	<code>\rn{kne} → (T3)</code>	(target on page 4)
	<code>\rn{clauseup} → PP'</code>	(target on page 7)
	<code>\rf{lastnew} → (PPaii)</code>	(target on page 6)
	<code>\rf{lastnewb} → (PP'A2)</code>	(target on page 7)
	Both <code>\rf</code> and <code>\rn</code> have an optional argument which produces an optional suffix. <code>\rf[*]{nice}</code> produces (1*).	
Inner delimiters	Both <code>linguex</code> and <code>philex</code> provide customizing of inner delimiters in complex cross-references. <code>linguex</code> has changed in version 4.0 (see its manual). With <code>philex</code> , use the <code>\phildashes{ }{ }</code> command. This works together with the <code>\subformat</code> and the <code>\subsubformat</code> commands. The first argument sets the delimiter between the top-level reference and the reference to the first sub-environment, while the second argument sets the delimiter between references to the first and the second sub-level environments. To illustrate. We use the input	
<code>\phildashes</code>	<pre> \lb{cool}{We would like a colon \phildashes{:}{ } \subformat{a}{ }{ }} \lba{coola}{before reference to this.} \lba{coolb}{but nothing more before reference to this}} </pre>	
	to create and make a cross-reference with <code>\rf{coolb}</code> to (5:ai) below:	
	<pre> (5) We would like a colon     a) before reference to this       (i) but nothing more before reference to this. </pre>	
hyperref	<p>If the <code>hyperref</code> package is called (must be called last) in the preamble, all ordinary cross-references of the label+cross-reference mechanism of LaTeX are by default made into pdf hyperlinks. This includes all cross-references in <code>philex</code> that rely on counters. If the <code>&lt;hyper&gt;</code> package option is also called, cross-references to <i>named</i> environments are made into hyperlinks as well.</p> <p>This is useful for reading pdf:s on the screen, and also for <code>beamer</code> slide-show presentations: click on a cross-reference to a named principle, and you are taken to the page where that principle is stated. In <code>beamer</code> you are taken to the first slide of the relevant frame.</p>	
Cross-reference to sub-environments in footnotes	<p>Sub-environments can be used in footnotes as well. But here, to get the cross-references right, the <code>subformat</code> command must be used.<sup>3</sup> The example</p> <hr/> <pre> (i) A first footnote example:     a) Be careful in footnotes. (ii) A second footnote example. </pre> <p>This cross-reference to (ia) is correct.</p>	

is produced by the input

```
\lb{ffnote}{A first footnote example:
\subformat{a}{}{}}
\lba{fnota}{Be careful in footnotes.}}
\lb{sfnote}{A second footnote example.
This cross-reference to \rf{fnota} is correct.}
```

Repeating content

`philex` also has cross-reference functions for repeating the *contents* of environment. By means of `\rp{nicy}` you repeat the content of (1'): But yesterday was even nicer. If you want to embed the content in another sentence, you may want the initial capital letter to be made lowercase. Then use the `\ml` command, as in

(6) John says that this is a nice day.  
Not too hot.

produced by

```
\lb{jsays}{John says that \ml{nice}}
```

In case you want a full stop at the end of an earlier environment and want to embed the content without the stop, you can use the `\philfullstop` command to have it inserted for you:

```
\philfullstop
\lb{stop}{It is very cold}
\philexclaim
\lb{ifstop}{If \ml{stop}, you should put on a cap}
```

produces

(7) It is very cold.  
(8) If it is very cold, you should put on a cap!

Finally, one can repeat an entire environment with label by means of the `\rff` command. `\rff{nice}` produces

(1) This is a nice day.  
Not too hot.

## 6 Lengths and spaces

`linguex` provides a number length and spacing commands that can be used to customize the appearance of `philex` environments. Below are given the length name, brief explanation, and the default value set by `linguex`. ‘hspace’ is short for ‘horizontal space’ and ‘vspace’ analogously:

<code>\Exlabelsep</code> (hspace after label)	1.3em
<code>\Extopsep</code> (extra vspace above and below)	.66\baselineskip
<code>\SubExleftmargin</code> (hspace after sub-level label)	2em
<code>\SubSubExleftmargin</code> (hspace after sub-sub-level label)	2.4em
<code>\Exindent</code> (indent of the environment)	Opt

Beside these, `philex` modifies `linguex` by letting two re-definable commands replace hard-coded lengths:

<code>\phlabelwidth{}</code> (width of label)	3em
<code>\phlabeldefault</code> (returns to default value)	
<code>\phlabelsep{}</code> (sets hspace after label to the argument, e.g. 1cm)	
<code>\phlabelsepdefault</code> (sets default value of the label separation)	
<code>\Exredux</code> (vspace reduction between environments)	-\baselineskip

If one needs longer labels, e.g. as names of principles or for other purposes, one can use `\phlabelwidth` in combination with `\Exlabelsep`. For example

```
\setlength{\Exlabelsep}{2cm}
\phlabelwidth{4cm}
\broff
\lb{defe}{Compositionality:}{The meaning ... etc. }
```

has the following effect:

Compositionality:      The meaning of a complex expression is a function of  
the meanings of its parts and its mode of composition.

For two-column documents, the label separation should preferably be somewhat reduced. This has to be set manually by means of these commands, since there is no automatic adaptation to two-column options or environments.

Vertical spacing is controlled with `\Extopsep` and `\Exredux`. The second controls the reduction of inter-environment vertical space and needs to be adapted to the general control of vertical space. To illustrate, the input

```
\setlength{\Extopsep}{2\baselineskip}
\setlength{\Exredux}{-3\baselineskip}
```

has the following effect on vertical spacing:

(9)              First vertical space example

(10) Second vertical space example

(11) Third vertical space example

Next line comes here.

## 7 Troubleshooting

Infelicitous labels

The most common problem you can run into with **philex** derives from an infelicitous choice of label (first argument to the commands). **philex** creates new command names (control sequences) from the labels, and especially with short labels it sometimes happens that a generated label coincides with a control sequence defined by TeX, or LaTeX, or some document class or package. Typesetting will halt with an error message that does not give the appropriate information. The location of the error need not be right either.

To remedy, try replacing your latest label(s) by new ones that are unlikely to clash with other control sequences.

**hyperref** and the  
.aux file

Another type of error can occur in case the **hyperref** package is used, and **hyperref** does not find the information it needs in the .aux file, perhaps because of an error in a previous round.

The remedy is to enter non-stop typesetting mode by pressing ‘r’ in response to the error message. Typesetting will be completed, with incorrect results, but the next round of typesetting will then work as it should. If there is still a problem, abort typesetting, trash the auxiliary file and start over.

math environment  
in **lbu**

A problem may arise because LaTeX style math environment delimiters ( $\langle . \rangle$ ) are used in the third argument (the update symbol) of the **\lbu** command. To avoid this problem, use TeX style delimiters ( $\langle . \rangle$ ) instead.

Clash between  
**hyperref** and  
**linguex**

A further possible problem derives from a clash between **linguex** and fairly recent versions of **hyperref**. This (as has been explained to me by Heiko Oberdiek) depends on a conflict over the use of **\b**. The result will be that sub-environments will lack labels and be incorrectly formatted.

The remedy has been provided by Heiko Oberdiek, who has added a workaround in **hyperref**. To make use of this, make sure that your version of **hyperref** is 6.78o (from 2009.02.02) or later.

Clash between  
**setspace** and  
**linguex**

There is also a conflict between the **setspace** package and **linguex**. The problem seems to be that **setspace** redefines **\@footnotetext** in a way that interferes with the way **linguex** is meant to work in footnotes.

To avoid this problem, load **philex** (and **linguex** too, if called) *after* **setspace**. Also, don’t use the **\setstretch** command in the preamble for setting line spacing. Rather set a **spacing** environment in the main body, after **\begin{document}**. This environment may span the entire main document.

Comments

Finally, bug reports, comments or suggestions are welcome and should

be directed to Peter Pagin at [peter.pagin@philosophy.su.se](mailto:peter.pagin@philosophy.su.se).