The **langnames** package*

Alejandro García Matarredona
alejandrogarciaag41@gmail.com

August 25, 2022

Abstract

The **langnames** package provides a set of macros for formatting names of languages, as well as their identification (in the form of ISO 639-3 codes) and their classification (in the form of its top-level family). The datasets from WALS and Glottolog are included in the package. The package also allows users to rename and add new languages.

1   Introduction

The typing out of language names in academic papers, especially those in language typology or related fields where many names have to be typed many times, is often inconvenient and inconsistent. This package attempts to be a small help to writers, especially of large projects or of collaborative ones, to have a slightly easier time with names of languages. It does so by defining three main commands: `\lname`, `\liso`, and `\lfam`, which respectively print out the name, name and ISO 639-3 code, and name and family of the specified language. While the package comes with about 7500 pre-defined languages, with code, name, and family, the user may also define new ones through the `\newlang` command. The basic use of all four of these commands is explained below.

2   Usage

2.1   Installation

Download the package from wherever it was found to a place where LaTeX may see it, typically in $TEXMFHOME/tex/latex. **langnames** should automatically load the **xkeyval** package.

2.2   Package options

When calling `\usepackage{langnames}`, the user must specify one of three options: glottolog, wals, or none. The first option, glottolog, selects the naming conventions from the Glottolog database. The second option, wals, predictably selects the naming conventions of the WALS database. The names and the genetic classification differ in some languages, so the user may choose what convention to

---

*This document corresponds to **langnames** v1.0, dated 2022/08/25.

follow. During the preparation of the dataset, there were instances of languages which appeared in WALS but not in Glottolog, and vice-versa. In such cases, the missing information was added from the other database. For more details on how I built the dataset, one may consult the Python script made for it in the Github repository. The third option, `none`, tells the package not to load either of the datasets, and instead start off from an empty canvas. If one specifies this option, one will have to fill in the details of each language with the macro `\newlang` (see explanation in Section 2.3 below).

2.3   Macros

When referring to a language, the author may use one of three macros to print out different information about it. Languages are identified by their ISO 639-3 code. `\lname`   The simplest macro is `\lname`, which prints out the name of the specified language according to the code provided. The basic syntax is thus `\newlang` {⟨*ISO code*⟩}. This can be seen in example (1).

(1)   My native language is \lname{cat}.

My native language is Catalan.

`\liso`   One may also use the `\liso` macro to print out both the name and the ISO 639-3 code of the language specified in the macro in parenthesis , again according to its ISO code(`\liso` {⟨*ISO code*⟩}). Example (2) shows its behavior.

(2)   I have recently taken up \liso{brg}.

I have recently taken up Baure (Arawakan).

`\lfam`   A third macro for use is the `\lfam` command, which prints the name of the language and its family in parenthesis. Once again, the language is identified by its ISO 639-3 code. Example (3) shows how it works.

(3)   The tone system of \liso{ptk} is fascinating.

The tone system of Maleng (Austroasiatic) is fascinating.

`\newlang`   Finally, users may add their own languages (or change the code, name, or genetic affiliation of a language already in the dataset) via the use of the `\newlang` command, which takes the three arguments {⟨*code*⟩}, {⟨*name*⟩}, and {⟨*family*⟩}. This command may be used in the preamble, before `\begin{document}`. Example (4) shows its usage.

(4) `\newlang{boo}{Ameli}{Amelian}`
      `\begin{document}`
      `My new made up language is \lname{boo}.`
      `My new made up language is \liso{boo}.`
      `My new made up language is \lfam{boo}.`
      `\end{document}`

      My new made up language is Ameli.
      My new made up language is Ameli (ISO 639-3: boo).
      My new made up language is Ameli (Amelian).

Be aware that setting a new language overwrites any other language with the same code, as the package only listens to the language that is defined last.

3   Implementation

3.1   Dependencies

The langnames package needs to load the xkeyval package for its key-value pair setting functionality.

```
1 \usepackage{xkeyval}
```

3.2   Option setting

Options are set for what dataset to use. glottolog use Glottolog data; wals uses WALS data; none selects neither dataset and all languages are defined by the user. See langnames.py in the Github repository to see how I gathered and handled the data.

Options

```
2 \DeclareOption{glottolog}{\input{langs_glot.tex}
3                          \input{fams_glot.tex}}
4 \DeclareOption{wals}{\input{langs_wals.tex}
5                      \input{fams_wals.tex}}
6 \DeclareOption{none}{}
```

Language codes, names and families are set with the `\define@key{⟨family⟩}` `{⟨key⟩}` `{⟨value⟩}` macro from the xkeyval package(see the langs_.tex and fams_.tex files in the package folder). `{⟨family⟩}` is either names or fams. Thus, each language has two key-value pairs that refer to it, one defining its name and the other one defining its family, both using the ISO 639-3 code as their key.

3.3   Macro definitions

lname   The `\lname` macro takes the key specified in its mandatory argument to call its corresponding key value pair from the names family, and prints it. This is achieved through the use of the `\csname` and `\endcsname` macros. The `\unskip` macro is used in all the macro definitions to avoid the adding of an extra space after the macro has been printed.

```
7 \newcommand{\lname}[1]{%
```

```
8    \csname KV@names@#1\endcsname\unskip
9 }
```

liso   The \liso macro takes, like \lname, the value from the names family from the argument input, and prints the name as well as the ISO 639-3 code (which is the argument verbatim) between parenthesis.

```
10 \newcommand{\liso}[1]{%
11   \csname KV@names@#1\endcsname{} (ISO 639-3: #1)\unskip
12 }
```

lfam   The \lfam macro, like \lname and \liso, calls the key-value pair from the names family corresponding to the input of the mandatory argument, plus the key-value pair from the fams family which gives it the genetic affiliation, which is printed between parenthesis.

```
13 \newcommand{\lfam}[1]{%
14   \csname KV@names@#1\endcsname{} (\csname KV@fams@#1\endcsname{})\unskip
15 }
```

newlang   The macro \newlang defines new keys for a language from the three mandatory arguments with the \define@key {⟨family⟩} {⟨key⟩} {⟨value⟩} macro from xkeyval. The first argument of \newlang {⟨code⟩} defines the code which serves as identifier (the ISO code in the case of pre-defined key-value pairs). The second argument {⟨name⟩} defines the printed name of the language. The third argument {⟨family⟩} defines the family to which the language belongs.

```
16 \newcommand{\newlang}[3]{%
17   \define@key{names}{#1}{#2}
18   \define@key{fams}{#1}{#3}
19 }
```

ProcessOptions   This line of code simply tells the package to set the options specified above.

```
20 \ProcessOptions \relax
```