

Multilingual Bibliographies: Using and extending the `babelbib` package

Harald Harders

1 Introduction

When generating bibliographies using `BIBTEX`, style files (file extension `.bst`) are used to determine the appearance of the bibliographies. Most of the available `BIBTEX` styles are hardcoded to a specific language, often English. This is unsatisfactory in many cases. If you, for example, write a German document and use one of the standard `bst` files, English keywords as “edition”, “page”, etc. are used instead of their German translations »Auflage«, »Seite« etc.

Another limitation of most `BIBTEX` styles is the fact that even small changes of the bibliography’s layout are not possible without creating a new `bst` file. Since the syntax of `bst` files differs much from `LATEX` this is difficult for most `LATEX` users.

The package `babelbib` provides solutions for both problems. It is available from the CTAN network in the directory `CTAN:biblio/bibtex/contrib/babelbib/`.

2 Multilingual bibliographies

2.1 Available packages

The restriction to one bibliography language is avoided by the packages `bibgerm` and `babelbib`. Both use following approach: Their `BIBTEX` styles use `TEX` macros instead of hardcoded strings, e.g. the command `\btxeditorlong` instead of the string “editor”. These commands are defined within the packages for different languages differently, e.g. “editor” in English, »Herausgeber« in German, or «editore» in Italian.

`bibgerm` [4] is the older package and has served as basis for `babelbib`. It is restricted to English and German and works together with the `babel` [1], `german`, and `ngerman` [3] packages. `bibgerm` works both with plain `TEX` and `LATEX`. It does not provide commands to change the typography of bibliographies.

`babelbib` [2] has been developed in order to be extendable to more languages in cooperation with the `babel` package. Thus, it needs the `babel` package to be loaded, too. Version 0.40 of the package `babelbib` supports Afrikaans, Danish, Dutch, English, French, German, Italian, Portuguese, Swedish, and Spanish.¹ The author would be grateful for any offers of assistance with adding more languages. How this can be done is described in section 4. The

¹ In some languages, not all names for other languages are present, yet. For instance, the French name for new-norwegian (Nynorsk) is not defined.

current version only runs with `LATEX 2ε`. `babelbib` provides commands to change the typography of a bibliography within the `LATEX` source, described in section 3.2.

With `babelbib`, replacements for the standard `BIBTEX` styles (`bababrv`, `babalpha`, `babplain`, and `babunsrt`) as well as a replacement for the $\mathcal{A}\mathcal{M}\mathcal{S}$ `BIBTEX` style `amsplain` (`babamspl`) are shipped. In addition, two styles `bababbr3` and `babplai3` are included that convert a list of more than three authors to “first author *et al.*”. All styles have multilingual support, include additional field types (described in section 3.1), and allow easy layout changes (section 3.2).

2.2 Different approaches for multilingual bibliographies

Two approaches are possible for bibliographies with flexible languages:

Each citation can use the language of the cited document. Then, the keywords vary within one bibliography. This approach is used by `bibgerm`. The following example shows this behaviour:

References

- [1] Beitz, W. und K.-H. Küttner (Herausgeber): *Dubbel – Taschenbuch für den Maschinenbau*, Kapitel Werkstofftechnik, Seiten E 1–E 120. Springer-Verlag, Berlin, 17. Auflage, 1990, ISBN 3-540-52381-2.
- [2] Dieter, George E. *et al.* (editors): *Materials Selection and Design*, volume 20 of *ASM Handbook*, chapter Effects of Composition, Processing, and Structure on Properties of Engineering Plastics, pages 434–456. ASM International, 1997, ISBN 0-87170-386-6.

The second approach uses the document’s main language for the whole bibliography. Thus, the keywords are uniform. This means for example, that the “edition” of a cited document in a German text is named »Auflage« also for cited documents that are not German. Nevertheless, the data fields (title, authors, etc.) are typeset in the citation language given for the cited document in order to use the correct hyphenation patterns. For example, the above bibliography looks in an English text like this:

References

- [1] Beitz, W. and K.-H. Küttner (editors): *Dubbel – Taschenbuch für den Maschinenbau*, chapter Werkstofftechnik, pages

E 1–E 120. Springer-Verlag, Berlin, 17. edition, 1990, ISBN 3-540-52381-2.

- [2] Dieter, George E. *et al.* (editors): *Materials Selection and Design*, volume 20 of *ASM Handbook*, chapter Effects of Composition, Processing, and Structure on Properties of Engineering Plastics, pages 434–456. ASM International, 1997, ISBN 0-87170-386-6.

And in a German text like this:

Literatur

- [1] Beitz, W. und K.-H. Küttner (Herausgeber): *Dubbel – Taschenbuch für den Maschinenbau*, Kapitel Werkstofftechnik, Seiten E 1–E 120. Springer-Verlag, Berlin, 17. Auflage, 1990, ISBN 3-540-52381-2.
- [2] Dieter, George E. *et al.* (Herausgeber): *Materials Selection and Design*, Band 20 der Reihe *ASM Handbook*, Kapitel Effects of Composition, Processing, and Structure on Properties of Engineering Plastics, Seiten 434–456. ASM International, 1997, ISBN 0-87170-386-6.

Both approaches can be typeset with `babelbib`, described in the following sections.

2.3 Using the package `babelbib`

To use the features of `babelbib`, you have to load it by:

```
\usepackage{babelbib}
```

The default behaviour is to change the language settings depending on the cited document (first approach in section 2.2). If you want a unique language over the whole bibliography, use the option `fixlanguage`:

```
\usepackage[fixlanguage]{babelbib}
```

Then, the language is fixed to the main document language. If you want to use another language, you may change it by

```
\selectbiblanguage{<language>}
```

The following `BIBTEX` styles are available for `babelbib`: `bababbr3`, `bababbrv`, `babalpha`, `babamspl`, `babplai3`, `babplain`, and `babunsrc`. You select one of them by using:

```
\bibliographystyle{<style>}
```

If you use `babamspl`, you have to load the package with the option `languagesnames` because the `AMS` `BIBTEX` styles typeset the language of the cited document and `babelbib` has to define the names of the

languages, then. This is not done by default to save memory.

You can also use the `BIBTEX` styles of the package `bibgerm` (`gerabbrv`, `geralpha`, `gerapali`, `gerplain`, `gerunsrc`) but with fixed typography.

2.4 `BIBTEX` databasefiles (*.bib)

The `BIBTEX` database files (extension `.bib`) for usage with `babelbib` don't differ much from standard files. All document types have the additional field `language` which should be given for *each* cited document. The examples, given above, were generated using following bib file:

```
@InBook{dubbel1990a,
  editor = {Beitz, W. and K"uttner, K.-H.},
  title = {Dubbel~-- Taschenbuch f"ur den
           Maschinenbau},
  chapter = {Werkstofftechnik},
  publisher = {Springer"=Verlag},
  year = 1990,
  address = {Berlin},
  edition = {17.},
  pages = {E"1--E"120},
  isbn = {3-540-52381-2},
  language = {ngerman}
}

@InBook{dieter1997a,
  editor = {Dieter, George~E. and others},
  title = {Materials Selection and Design},
  chapter = {Effects of Composition,
            Processing, and Structure on
            Properties of Engineering
            Plastics},
  publisher = {ASM International},
  year = 1997,
  volume = 20,
  series = {ASM Handbook},
  pages = {434--456},
  isbn = {0-87170-386-6},
  language = {english}
}
```

Note that all extensions and shortcuts provided by `babel` for the different languages, e.g. "u instead of \u for "ü in German items, can be used in the fields of a document, if { and } are used as delimiters in the `bib` file.

If you leave out the specification of the language for a citation item this item is typeset in the document's main language if you use one of the `bab*.bst` styles. In addition, a warning is generated for the left out `language` field. If you use a `bibgerm` style (`ger*.bst`), no warning is produced, and—what is more important—the new item is typeset in the language of the preceding citation in the bibliography which may not be wanted.

Element	Fields	Default value	
		bababbr3, bababbrv babamspl babalpha, babplai3 babplain, babunsrt	
name	authors, editors		
title	title, series	\emph	\emph
etal	“ <i>et al.</i> ”	\emph	
journal	journal title		
volume	volume (journal)		\textbf
ISBN	ISBN	\MakeUppercase	\MakeUppercase
ISSN	ISSN	\MakeUppercase	\MakeUppercase
url	net address	\url	\url

Table 1: Default values of the fonts in bibliographies. A missing value means, that the standard font of the document is used.

3 Other extensions

The `babelbib` package as well as the associated `bst` files contain additional extensions that don’t concern the multilingual support.

3.1 More data fields supported by the `bst` files

The `bab*.bst` styles support three additional fields for most of the document types.

You are able to specify the ISBN resp. ISSN of the documents, using the equally named fields, as can be seen in the example above.

Using the field `url`, URLs can be given. If the command `\url` is available, e.g. by loading `url.sty`, they are printed using it. If not, `babelbib` defines a `\url` command that produces an error message when using the field URL.

3.2 Flexible typography of the bibliography

The standard `bst` files have a fixed typography of the bibliography. Even small changes (e.g. changing the font for author names to small caps) need to change the `bst` file. To avoid that, the `bab*.bst` files use user definable font commands for some elements of the bibliography.

The approach works as follows: If the user does not specify the font for an element of the bibliography, the `BIBTeX` style includes a default font that it uses. If, in contrast, the user specifies a font, this value is taken instead of the default.

Fonts of the bibliography are changed by using the

```
\setbibliographyfont{<element>}{<font command>}
```

command. The possible elements and font commands are listed in Table 1.

The font command has to be a `LATEX` command with exactly one argument, e.g. `\emph`, `\textbf`, or `\textsc`. You can also use commands as `\mbox`, that do not change a font but, for example, inhibit line breaks within one element. This can be interesting for ISBN and ISSN.

If you want to switch to a font for which no command is available, that accords to the rule, you have to specify a new command, e.g.

```
\newcommand\textitbf[1]{%
  {\bfseries\itshape #1\}}%
\setbibliographyfont{title}{\textitbf}%
```

In the argument of `\setbibliographyfont`, the font command is given without an argument, as shown in this example.

If you try to define a font for an element not listed in Table 1, an error message is generated. To define a font for a new element that is not known by `babelbib`, e.g. for a custom `BIBTeX` style, the `\setbibliographyfont*` command is available.

Internally, the `\setbibliographyfont` and `\setbibliographyfont*` commands define a command with a name built by `\bt<element>font`, e.g. `\btxtitlefont` for titles. This command can be used by the `bst` files.

The strings “ISBN” and “ISSN” are generated by the commands `\btISBN` and `\btISSN`. They don’t take an argument. By default, these commands just write the corresponding strings without a change of the font. In this article, they have been changed as follows:

```
\renewcommand\btISBN{\acro{ISBN}}
\renewcommand\btISSN{\acro{ISSN}}
```

where `\acro` prints the text slightly smaller. Another possibility could be to use small caps:

```
\renewcommand\btISBN{\textsc{isbn}}
```

```
\renewcommand\btxISSN{\textsc{issn}}
```

3.3 Changing keywords

If you don't like some of the keywords provided by `babelbib` you are able to change them using `\declarebtxcommands`. For example, it is possible to call Ph.D. theses »Dissertation« or »Doktorarbeit« in German, where the first name is used by `babelbib`. If you want to change it, type the following:

```
\declarebtxcommands{german}{%
  \def\btxphdthesis#1{%
    \foreignlanguage{german}{Doktorarbeit}}%
}
```

As it can be seen in the example, the command changes the keyword for the language specified in the first argument, while the second argument gives the (re)definition of the command. You may change more than one command within one call of `\declarebtxcommands`, but you have to avoid the insertion of unwanted spaces. Which `\btx...` command you have to change can be found out by searching for the unwanted keyword in the language-dependent `bdf` file (see the next section).

`\declarebtxcommands` can also be used to add new keyword commands, e.g. for newly developed `BIBTEX` styles.

4 Adding new languages to babelbib

The package `babelbib` includes a list of known languages. It determines automatically which of these have been loaded by `babel`. It then defines the bibliographic keywords for them. This is done by loading special files (extension `.bdf`) that provide the keyword definitions, similarly to the language definition files (extension `.ldf`) of `babel`.

If the user provides a new `bdf` file the package `babelbib` does not know about it and thus cannot load it automatically. Then, the user has to specify it as option when loading `babelbib`. Say, you have provided `norsk.bdf`. Then you have to load `babelbib` as follows:

```
\usepackage[norsk]{babelbib}
```

If you have generated a new `bdf` file or if you have extended one of the other files, please send them to my email address. Then, I can include your changes into the distribution.

4.1 Writing new bdf files

The `bdf` files are doing two things. First, they provide the commands that contain the keywords for bibliographies. And second, they append the call of these commands to the `\extras{language}` com-

mand of the loaded languages, if the option `fixlanguage` is not used. In the further text, the organisation of `bdf` files is described for the example of `english.bdf`.

The commands for the bibliographic keywords are called `\btx{keyword}` for keywords in the middle of a sentence (often starting with lowercase letters) resp. `\Btx{keyword}` for keywords at the beginning of a sentence (starting with uppercase letters). Many of these commands provide a long and a short (abbreviated) version, for which `long` resp. `short` is appended to the command name, e.g. `\btxeditorlong` for "editor" and `\btxeditorshort` for "ed".

The keyword definitions are put into a command `\bibs{language}`, e.g. `\bibsenglish`, which is called when the document language is changed by `\selectlanguage`, if `fixlanguage` is not set, or at `\begin{document}`, if `fixlanguage` is set.

A part of the command `\bibsenglish` looks like this:

```
\newcommand\bibsenglish[1][english]{%
  \def\biblanguagename{#1}%
  \def\btxetalshort##1{%
    \foreignlanguage{#1}{et~al~##1{}}}%
  :
  :
  \def\btxeditorshort##1{%
    \foreignlanguage{#1}{ed##1{}}}%
  \def\btxeditorlong##1{%
    \foreignlanguage{#1}{editor}}%
  \def\btxeditorssshort##1{%
    \foreignlanguage{#1}{eds##1{}}}%
  \def\btxeditorslong##1{%
    \foreignlanguage{#1}{editors}}%
  :
  :
  \def\Btxeditorshort##1{%
    \foreignlanguage{#1}{Ed##1{}}}%
  \def\Btxeditorlong##1{%
    \foreignlanguage{#1}{Editor}}%
  :
  :
  \ifbbblanguagenames
    \def\btxlanguagenameamerican{%
      \foreignlanguage{#1}{english}}%
    \def\btxlanguagenameaustrian{%
      \foreignlanguage{#1}{german}}%
    :
    \def\btxlanguagenamefrenchb{%
      \foreignlanguage{#1}{french}}%
    \def\btxlanguagenamegerman{%
      \foreignlanguage{#1}{german}}%
    :
    \def\btxlanguagenameUKenglish{%
      \foreignlanguage{#1}{english}}%
```

```

\def\btxlabelnameUSenglish{%
  \foreignlanguage{#1}{english}}%
\fi
}

```

The `\btxlabelname...` commands typeset different language names in the keyword language of citations. This is necessary if the `BIBTEX` style writes the language of the citation into the bibliography, as `babamspl` does it. In order to save memory, the language names are only defined if the option `language-names` is set when loading the `babelbib` package.

The `\bibsenglish` command takes one optional argument which specifies the language of the keywords. By default, it is `english`. This optional argument is useful for defining English dialects, that mostly use the same keywords. For example, American is defined like this:

```

\newcommand\bibsamerican{%
  \bibsenglish[american]}

```

`\bibsamerican` simply calls `\bibsenglish` with the keyword language changed to `american`. If, for example, there was an English dialect “myengl” where Master’s theses were called “Diploma thesis”, the definition could look like this:

```

\newcommand\bibsmyengl{%
  \bibsenglish[myengl]}
\def\btxmastthesis##1{%
  \foreignlanguage{myengl}{Diploma thesis}}%
}

```

This first would set `\btxmastthesis` to “Master’s thesis” and then redefine it to “Diploma thesis”. This approach wastes some time, but it avoids to repeat identical entries in the source code.

All commands defined by `\bibsenglish` take one argument `##1`, whose content is appended to the keyword text in some cases. This can be used by the `bst` files to append the dot for abbreviations. For uniformity, all commands take this argument even if they don’t need it.² All `\btx...` and `\Btx...` commands switch to the keyword language using `\foreignlanguage` and typeset the keyword as specified. Thus, the keywords are hyphenated correctly.

The second part of the `bdf` file appends the macro `\bibs⟨language⟩` to the `\extras⟨language⟩` command for all languages that are loaded by `babel`, if `fixlanguage` is not used. This is done by the command `\bbbbbaddto{⟨language⟩}` which is called at `\begin{document}` for all dialects defined in the `bdf` file (which are American, British, Cana-

² This is due to the fact that `bibgerm` does it like that. Both packages are intended to stay compatible to some extent.

dian, English, UK English, and US English³ for `english.bdf`):

```

\AtBeginDocument{%
  \ifbbbfixlanguage
  \else
    \bbbbbaddto{american}{bibsamerican}
    \bbbbbaddto{british}{bibsbritish}
    \bbbbbaddto{canadian}{bibs canadian}
    \bbbbbaddto{english}{bibsenglish}
    \bbbbbaddto{UKenglish}{bibsUKenglish}
    \bbbbbaddto{USenglish}{bibsUSenglish}
  \fi
  \bbbbbaddto{american}{btixifchange case on}
  \bbbbbaddto{british}{btixifchange case on}
  \bbbbbaddto{canadian}{btixifchange case on}
  \bbbbbaddto{english}{btixifchange case on}
  \bbbbbaddto{UKenglish}{btixifchange case on}
  \bbbbbaddto{USenglish}{btixifchange case on}
}

```

The switch `\ifbbbfixlanguage` ensures that this is only done if `fixlanguage` is not set.

The second part of this code snippet, after the `\fi`, is necessary, because the case of titles is changed in some languages and preserved in others. For example, in English, titles are printed lowercase, while in German, titles are printed as given. This is reached by following approach: The `BIBTEX` style prints the title twice as arguments of the `\btixifcasechange` commands. The first one is lowercase, the second with preserved case. The `LATEX` code then decides based on the language which version is typeset. There are two commands `\btixifchange case on` and `\btixifchange case off` that switch between both behaviours. Since in all English dialects the case of titles is changed, `\btixifchange case on` is appended to `\extras⟨language⟩`. If a language does not change the case, you have to append `\btixifchange case off` instead.

If you want to generate a `bdf` file for a new language you should copy an existing one to a new file and then change it. To test the new language, `babelbib.sty` does not have to be changed. Instead, specify the name of the new `bdf` file without extension as option to the `\usepackage[⟨filename⟩]{babelbib}` command.

4.2 Extending the package `babelbib`

The package file `babelbib.sty` provides the common commands for all languages and loads the necessary `bdf` files. Therefore, it contains a list of all known languages and dialects. `babelbib` version 0.40

³ For some dialects, different names are available (e.g. American and US English), since `babel` also supports different names for some dialects.

knows about the following languages and dialects: afrikaans, american, austrian, brazil, brazilian, british, canadian, canadien, danish, dutch, english, franceis, french, frenchb, german, germanb, italian, mexican, naustrian, ngerman, portuges, portuguese, UKenglish, USenglish, spanish, and swedish.

The language definitions are loaded by the command

```
\inputbdf{<language>}{<filename>},
```

where *<language>* is the dialect and *<filename>* is the name of the bdf file without the extension. If you add a new language, just add a new line containing an `\inputbdf` command to the list of `\inputbdf` commands.⁴

5 Adapt other BibTeX styles to babelbib

Using the example `amsplain.bst`, it is shown how other BibTeX styles can be adapted to `babelbib`. The resulting `bst` file is included in the `babelbib` distribution, called `babamspl.bst`.

5.1 Multilingual support

The $\mathcal{A}\mathcal{M}\mathcal{S}$ BibTeX styles are different to the standard styles in one aspect: They print the language of the citation for some document types. Thus, they already have the BibTeX field `language`. This can be seen in the list of supported fields:⁵

```
ENTRY
{ address
  author
  booktitle
  chapter
  edition
  editor
  howpublished
  institution
  journal
  key
  language
  month
  mrnumber
  note
  number
  organization
  pages
  publisher
  school
  series
  title
  type
  volume
  year
```

⁴ If you do this, you have to rename your style file.

⁵ In the source code snippets, newly inserted, changed, and important lines are marked by “←”.

```
}
{}
{ label bysame }
```

If the field `language` is missing in a BibTeX style, it has to be inserted.

As described in section 4.1, `bst` files print titles twice—with changed case and with preserved case—in order to enable the L^AT_EX code to decide which version will be typeset, using the macro `\btixifcasechange`. This is done by the function `language.change.case`:

```
FUNCTION {language.change.case}
{
  'change.temp :=
  't :=
  "\btixifcasecase{"
  t change.temp change.case$ *
  "}" *
  t *
  "}" *
}
```

In order for this to work probably, the string variables have to be defined beforehand. Therefore, the line `STRINGS { s t }` is changed to

```
STRINGS { s t language.state
          change.temp }
```

At the beginning of each citation, the language has to be switched to the citation language, if it is different from the preceeding citation. Therefore, some code is integrated into the function `output.bibitem`:

```
FUNCTION {output.bibitem}
{ newline$
  language empty$
  { "empty language in " cite$ * warning$
    language.state "nolanguage" =
    'skip$
    {
      "\expandafter\btixselectlanguage"
      "\expandafter{" *
      "\btxfallbacklanguage}" * write$
      newline$
    }
    if$
    "nolanguage" 'language.state :=
  }
  { language.state language =
    'skip$
    { "\btixselectlanguage{"
      language * "}" *
      write$ newline$
    }
    if$
    language 'language.state :=
  }
  if$
```

```

"\bibitem{" write$
cite$ write$
"}" write$
newline$
""
before.all 'output.state :=
}

```

This function also generates a warning, if the language is omitted. In addition, the language is changed to a fall-back language which is the document's main language.

Since this BIBTEX style prints the citation language, a function `format.language` is defined, that typesets the language name in brackets. Many styles don't need this function. Since it would not be nice, if in non-English texts the languages are called with English names, e.g. "german" or "french" instead of »deutsch« or »französisch« in German texts, L^AT_EX macros are used instead of the language names of `babel`. These macros print the language name in the correct language. Therefore, the function `format.language` is used:

```

FUNCTION {format.language}
{ language empty$
  { "" }
  { " (\btxlabelname{"
    language * "}") * }
  if$
}

```

The command `\btxlabelname` prints the language name using the keyword language of the citation. This only works, if the option `languagenames` is used when loading `babelbib`:

```
\usepackage[languagenames]{babelbib}
```

If a language name is not available, an error message is generated and the name in the source code is used, instead.

Since the change of case is only used in some languages, the call of `change.case$` has to be replaced by `language.change.case`, e.g.

```

FUNCTION {format.title}
{ title empty$
  { "" }
  { title "t" language.change.case
    emphasize }
  if$
}

```

This has to be done for all occurrences.

5.2 Flexible typography

In this section, it is described how the typography of BIBTEX styles can be made flexible.

Some functions are defined that allow to switch the fonts easily. They are similar to the existing `emphasize` function:

```

FUNCTION {emphasize}
{ duplicate$ empty$
  { pop$ "" }
  { "\emph{" swap$ * "}" * }
  if$
}

FUNCTION {namefont}
{ duplicate$ empty$
  { pop$ "" }
  { "\btxlabelname{" swap$ * "}" * }
  if$
}

FUNCTION {titlefont}
{ duplicate$ empty$
  { pop$ "" }
  { "\btxlabelname{" swap$ * "}" * }
  if$
}

FUNCTION {journalfont}
{ duplicate$ empty$
  { pop$ "" }
  { "\btxlabelname{" swap$ * "}" * }
  if$
}

FUNCTION {volumefont}
{ duplicate$ empty$
  { pop$ "" }
  { "\btxlabelname{" swap$ * "}" * }
  if$
}

FUNCTION {etalfont}
{ duplicate$ empty$
  { pop$ "" }
  { "\btxlabelname{" swap$ * "}" * }
  if$
}

```

More font-switching commands can be defined analogous. Since the style `babelbib` does not know them, the user has to use `\setbibliographyfont*` instead of `\setbibliographyfont` in the L^AT_EX file, if he wants to use another font than the default one. Also, the `bst` file has to use `\providebibliographyfont*` instead of `\providebibliographyfont`, as described below. Please tell me, if you add a new font command, because I can add it to the package, then.

The font functions are called in the further functions of the `bst` file. For example, names (authors,

editors) are typeset by `format.names`, which is defined as follows:

```
FUNCTION {format.names}
{ 's :=
  #1 'nameptr :=
  s num.names$ 'numnames :=
  numnames 'namesleft :=
  { namesleft #0 > }
  { s nameptr "{ff~}{vv~}{ll}{, jj}"
    format.name$ 't :=
    nameptr #1 >
    { namesleft #1 >
      { " " * t namefont * } ←
      { numnames #2 >
        { "\btxandcomma{" * } ←
        'skip$
        if$
        t "others" =
        { " " * "\btxetalshort{." ←
          etalfont * } ←
          { " \btxandlong{ " * ←
            t namefont * } ←
            if$
          }
        }
        if$
      }
    }
    { t nameptr "{ff~}{vv~}{ll}{, jj}" ←
      format.name$ namefont } ←
    if$
    nameptr #1 + 'nameptr :=
    namesleft #1 - 'namesleft :=
  }
  while$
}
```

Additionally, the L^AT_EX macros `\btxandcomma`, `\btxetalshort`, and `\btxandlong` have been added, that print language-dependent keywords.

For titles, `emphasize` is replaced by `titlefont`:

```
FUNCTION {format.title}
{ title empty$
  { "" }
  { title "t" language.change.case ←
    titlefont } ←
  if$
}
```

Similarly the title and volume of journals:

```
FUNCTION {format.journal.vol.year}
{ journal empty$
  { "journal name" missing.warning "" } ←
  { journal journalfont } ←
  if$
  volume empty$
  'skip$
  { " " * volume volumefont * } ←
  if$
  year empty$
  { "year" missing.warning } ←
}
```

Text	Macro
and	<code>\btxandlong{}</code>
ch.	<code>\btxchaptershort{.}</code>
ed.	<code>\btxeditorshort{.}</code>
ed.	<code>\btxeditionshort{.}</code>
eds.	<code>\btxeditorsshort{.}</code>
et al.	<code>\btxetalshort{.}</code>
in	<code>\btxinlong{}</code>
in	<code>\btxinserieslong{}</code>
Master's thesis	<code>\btxmastthesis{}</code>
no.	<code>\btxnumbershort{.}</code>
of	<code>\btxofserieslong{}</code>
p.	<code>\btxpageshort{.}</code>
Ph.D. thesis	<code>\btxphtdthesis{}</code>
pp.	<code>\btxpageshort{.}</code>
Tech. Report	<code>\Btxtechrephshort{.}</code>
vol.	<code>\btxivolumeshort{.}</code>
January	<code>\btxmonjanlong{}</code>
February	<code>\btxmonfeblong{}</code>
⋮	⋮

Table 2: Replacements for B_IB_TE_X styles

```
{ " (" * year * ")" * }
if$
}
```

In `format.incoll.inproc.crossref`, the original use of `\emph` is replaced by `titlefont`; in `format.article.crossref`, `journalfont` is added.

As shown for `format.names`, the hardcoded keywords have to be replaced by the corresponding L^AT_EX macros. Table 2 shows important replacements. Sometimes, equal keywords have to be replaced by different macros. That depends on the context.

If you want to use new keywords, that are not included in the existing `bdf` files, you have to define them in the L^AT_EX document using `\declarebtxcommands`, as described in section 3.3.

The used fonts for the data fields have to be initialized at the beginning of the bibliography. This is done by the `\providebibliographyfont` command, that only does an initialization, if the author has not done it before. Since the function `begin.bib` starts the bibliography, the initializations are added here

(shortened by a few lines, marked by “:”):

```
FUNCTION {begin.bib}
{ preamble$ empty$
  'skip$
  { preamble$ write$ newline$ }
  if$
}
```



```

"\providecommand{\bysame}{\leavevmode\hbox "          \btxeditionsshort{.}" * }
"to3em{\hrulefill}\thinspace}" *                    if$
write$ newline$                                     }
:                                                     if$
"\providecommand{\href}2{#2}"                         }
write$ newline$
"\begin{thebibliography}{
  longest.label * "}" *
  write$ newline$
  " \providebibliographyfont{name}{}% " ←
  write$ newline$ ←
  " \providebibliographyfont{title}{
    "\emph}{%" * ←
    write$ newline$ ←
    " \providebibliographyfont{journal}{}% " ←
    write$ newline$ ←
    " \providebibliographyfont{etal}{}% " ←
    write$ newline$ ←
    " \providebibliographyfont{volume}{
      "\textbf}{%" * ←
      write$ newline$ ←
      " \providebibliographyfont{ISBN}{
        "\MakeUppercase}{%" * ←
        write$ newline$ ←
        " \providebibliographyfont{ISSN}{
          "\MakeUppercase}{%" * ←
          write$ newline$ ←
          " \providebibliographyfont{url}{\url}{%" ←
          write$ newline$ ←
}

```

The lines containing `\providebibliographyfont` are put as the first lines of the `thebibliography` environment by the `BIBTEX` style file instead of defining them in the definition of the `thebibliography` environment for two reasons: First, it is possible then to use different font defaults with different `BIBTEX` styles. And second, other packages may redefine the `thebibliography` environment without problems.

5.3 Additional data fields

The `babelbib` `BIBTEX` styles support the additional data fields `isbn`, `issn`, and `url`. Now, they are added to the `bst` file.

Like the field `language`, the names `isbn`, `issn`, and `url` have to be added to the `ENTRY` definition at the beginning of the `bst` file (see section 5.1).

The new fields are formatted by following functions:

```

FUNCTION {format.edition}
{ edition empty$
  { "" }
  { output.state mid.sentence =
    { edition "1" language.change.case "
      \btxeditionsshort{.}" * }
    { edition "t" language.change.case "

```

```

FUNCTION {format.isbn} ←
{ isbn empty$ ←
  { "" } ←
  { "\btxISBN~\btxISBNfont{" isbn * ←
    "" } * } ←
  if$ ←
} ←
FUNCTION {format.issn} ←
{ issn empty$ ←
  { "" } ←
  { "\btxISSN~\btxISSNfont{" issn * ←
    "" } * } ←
  if$ ←
} ←
FUNCTION {format.url} ←
{ url empty$ ←
  { "" } ←
  { "\btxurlfont{" url * "" } * } ←
  if$ ←
} ←

```

The new fields have to be printed for all citations, where they are useful. For example, for books, an ISBN and maybe a URL is useful, while an ISSN is senseless. Thus, the function `book` looks like this:

```

FUNCTION {book}
{ output.bibitem
  author empty$
  { format.editors "author and editor"
    output.check }
  { format.authors output.nonnull
    crossref missing$
    { "author and editor" editor
      either.or.check }
    'skip$
    if$
  }
  if$
  format.title "title" output.check
  format.edition output
  crossref missing$
  { format.bookvolume.series.number output
    publisher "publisher" output.check
    address output
  }
  { format.book.crossref output.nonnull
  }
  if$
  format.date "year" output.check
  format.isbn output ←
  format.url output ←
}

```

```
format.language *  
note output  
fin.entry  
}
```

In the same way, the functions `booklet`, `inbook`, `incollection`, `inproceedings`, `manual`, `masterthesis`, `misc`, `phdthesis`, `proceedings`, `techreport`, and `unpublished` are extended.

6 Conclusions

This article has described how the `babelbib` package can be used to generate multilingual and flexible bibliographies. In addition, it has been shown, how the `babelbib` system can be extended to more languages and `BIBTEX` styles.

Since the package is still young, the number of supported languages and `BIBTEX` styles is limited, yet. Thus, there are two main topics: Both, the number of languages and `BIBTEX` styles has to be increased. But I need help for both tasks.

I hope, the package is useful for generating bibliographies in multilingual environments, already.

References

- [1] Braams, Johannes: *Babel, a multilingual package for use with L^AT_EX's standard document classes*, 2002. CTAN:macros/latex/required/babel/.
- [2] Harders, Harald: *The babelbib package*, 2003. CTAN:biblio/bibtex/contrib/babelbib/.
- [3] Raichle, Bernd: *Kurzbeschreibung german.sty und ngerman.sty*, 1998. CTAN:language/german/.
- [4] Wallmeier, M., A. Scherer, and H. Harders: *Macros for german BIBTEXing*, 2000. CTAN:biblio/bibtex/contrib/germbib/.

◇ Harald Harders
Nußbergstraße 48
38102 Braunschweig
Germany
h.harders@tu-bs.de