Towards a New Bibliography Format

Jean-Michel Hufflen

Abstract

\textsc{mibibt\textsc{ex}}, our reimplementation of \textsc{bibt\textsc{ex}}, uses an enriched format for bibliography database files. Due to some features of this format, it is not backward-compatible with the conventions recognised by `old' \textsc{bibt\textsc{ex}}. We explain why and make precise the requirements for a modern bibliography format and the specification of our new format.

**Keywords** \textsc{mibibt\textsc{ex}}, bibliography processor, \textsc{unicode}, character encodings, bibliography format.

**Sommario**

\textsc{mibibt\textsc{ex}}, la nostra reimplementazione di \textsc{bibt\textsc{ex}}, usa un formato ampliato per il database bibliografico. A causa di alcune particolarità di questo formato, non è retrocompatibile con le convenzioni del `vecchio' \textsc{bibt\textsc{ex}}. Spieghiamo perché e mostriamo con precisione i requisiti per un formato bibliografico moderno e le specifiche del nostro nuovo formato.

**Parole chiave** \textsc{mibibt\textsc{ex}}, elaboratore bibliografico, \textsc{unicode}, codifiche dei caratteri, formato bibliografico.

**Preliminary note** The bibliography of the present article has been processed with \textsc{mibibt\textsc{ex}}, as a showcase for this program.

**Introduction**

As mentioned in Mittelbach and Goossens (2004, § 13.1), the \LaTeX\ typesetting system has deeply evolved since its first versions whereas \textsc{bibt\textsc{ex}}, the bibliography processor usually associated with \LaTeX, has remained stable for many years. Let us recall that this program reads an auxiliary (.\texttt{aux}) file built when \LaTeX\ typesets a source text (.\texttt{tex} file). The citation keys—used by the \texttt{\textbackslash cite} command throughout .\texttt{tex} files—are extracted from this auxiliary file, and \textsc{bibt\textsc{ex}} looks in bibliography database (.\texttt{bib}) files for these keys. The auxiliary file also gives the pathnames of the .\texttt{bib} files to be searched and the information needed for the layout of a `References' section. With \textsc{bibt\textsc{ex}}, such layouts are put into action by means of bibliography styles.

\textsc{bibt\textsc{ex}} is ageing, and we observe some possible replacement for several years. In particular, the \texttt{biber} program (Kim and Charette, 2014), generating references suitable for the \texttt{biblatex} package (Lehman et al., 2014). Let us recall that such references are marked up by means of \LaTeX\ commands; accurate redefinitions of these commands allow bibliography styles to be put into action. Another example of a possible replacement is our reimplementation, \textsc{mibibt\textsc{ex}} (Hufflen, 2015). Since there is a huge number of .\texttt{bib} files—especially on the Web—\textsc{bibt\textsc{ex}}'s successors still use this format for bibliography database files.

In Hufflen (2015) we introduced a new version of \textsc{mibibt\textsc{ex}}, presently in beta-test. We recalled some syntactic extensions for .\texttt{bib} files. But these extensions induce some backward-compatibility problems: if end-users of \textsc{mibibt\textsc{ex}} have written .\texttt{bib} files with this enriched syntax, what happens if they have to revert to `old' \textsc{bibt\textsc{ex}}? Such a scenario is possible nowadays, the whole process of publishing research papers in conference proceedings is often controlled by Web programs. On such sites, all of the successive steps are controlled: electronic submission, acceptance or rejection, and depositing the final version if the paper has been accepted. Concerning this last step, often \textsc{bibt\textsc{ex}} is the only usable bibliography processor when articles are typeset with \LaTeX.

This is clearly a drawback of using \textsc{mibibt\textsc{ex}}\footnote{Multi\textsc{lingual} \textsc{bibt\textsc{ex}}.}, but in our personal opinion this drawback is the price to pay to get more expressive power within .\texttt{bib} files. We think that this format should be redefined and extended in next years, since there are requirements—especially modern ones—difficultly implementable within `original' .\texttt{bib} format. In fact, this article aims to show that most of our syntactic extensions are debatable solutions out of .\texttt{bib} format expressive power. A comparable work about abbreviating authors’ and editors’ first names within bibliographies has already been presented in Hufflen (2016). The present work aims to give a more general view. In the first section, we recall this .\texttt{bib} format’s origin and show how some \LaTeX\ commands may be used as workarounds in quite simple cases. That is suitable for a `basic' use of \textsc{bibt\textsc{ex}}, but makes other applications more complicated: for example, formatting the contents of .\texttt{bib} files and displaying the result on the Web by means of \textsc{html}\footnote{The most famous site for Computer Science conferences is indisputably \url{http://www.easychair.org}.} pages. Section 2 shows other examples where putting \LaTeX\ commands does not apply or is unsatisfactory. As an alternative, the technique used by the \texttt{biblatex} package\footnote{3. Some analogous drawbacks exist for \texttt{biblatex} users, as shown in § 3.}}.
to increase .bib files’ expressive power is described in § 3. Then our choices are discussed in § 4. Reading this article requires some good knowledge of Bib\TeX; advanced technical details can be found in Mittelbach and Goossens (2004, Ch. 13).

1 The .bib format

An example of a bibliographical entry usable by Bib\TeX is shown in Fig. 1. In reality, this program (Pataki, 1988) was initially designed to work with Scribe (Reid, 1984). This historical point explains some features of Bib\TeX: Scribe has influenced \LaTeX—in particular, the notion of document style originates from Scribe—and was one of the first markup languages. However its syntax is close to \LaTeX’s but is not identical. In Scribe, the ‘$’ character is used at the beginning of a command name, such a command argument may be surrounded by braces, parentheses or double quote characters; ‘\’ is not a special character\footnote{As \LaTeX’s syntax is close to \LaTeX’s but is not identical. In Scribe, the ‘$’ character is used at the beginning of a command name, such a command argument may be surrounded by braces, parentheses or double quote characters; ‘\’ is not a special character.}, as in {\LaTeX}. We see the origin of the ‘$’ characters used within entry types such as @ARTICLE or @BOOK, the association of field names with corresponding values being surrounded by braces—mostly used—or parentheses. Even though Bib\TeX’s syntax is close to \LaTeX’s and braces can be used to group some consecutive characters, accent names are not recognised.

As \LaTeX has succeeded whilst Scribe has just had some historical value, the former has become the only word processor targeted by Bib\TeX; for many years, bibliography styles built ‘References’ sections for \LaTeX, not for Scribe. So writers get used to put \LaTeX commands inside values associated with fields, for example, to specify typographical effects:

\begin{verbatim}
TITLE = {\textbf{Babylon Babies}}
\end{verbatim}

(this work’s title uses italicised characters for foreign words, the book being written in French), or reach accented letters:

\begin{verbatim}
AUTHOR = {Herbert Vo\ss}
\end{verbatim}

(\ss command gives the German letter ‘ß’).

5. Analogous syntax is still used within Texinfo (Chassell and Stallman, 2008), the markup language used to document the products of the GNU (GNU’s Not UNIX) projects.

These two examples are suitable for \LaTeX, but not for Con\TeXt (Hagen, 2001), another format built out of \TeX: the \emph command is unknown and the \ss command causes a switch towards a sans-serif font\footnote{It is well-known that the AUTHOR and EDITORS fields are structured: co-authors or co-editors are separated by the ‘and’ keyword, as shown in Fig. 1, names are structured into four parts: First, von (a particle), Last and Junior\footnote{In the first case, you have to use the ‘...’ command to get it is \ss, the namesake command in \LaTeX causes the case of ‘ß’ to be raised and the result is ‘SS’.}. As suggested by our notation, the First, Last and Junior parts begin with an uppercase letter whereas the von part begins with an uppercase letter. The complete rules for parsing names in Bib\TeX are quite complicated\footnote{This part is also known as Lineage.}, but the first author’s name given in Fig. 1 is easily processed as follows: first => Lyon Sprague, von => de, last => Camp (by using MIBib\TeX’s alternative notation). If the particle begins with a lowercase letter, a workaround is needed as shown in Mittelbach and Goossens (2004, p. 766–768):

\begin{verbatim}
Maria {\MakeUppercase{d}e La} Cruz
\end{verbatim}

Maria \MakeUppercase{d}e La Cruz

Because of the ‘d’ letter, Bib\TeX considers that the second token begins with a lowercase letter—and the dummy \MakeUppercase command is only used to put an uppercase letter when the bibliography is typeset. Here also, we need a mini-\TeX parser to handle such cases for formats other than \LaTeX.

If you build a ‘References’ section where first names are to be abbreviated, let us recall that in most cases, only the first letter is retained, but some abbreviations use several letters, e.g., ‘Clive’, abbreviated into ‘Cl.’ In addition, some authors drop out their middle name, so ’Clive Eric Cussler’ should be abbreviated into ‘Cl. Cussler’. You can specify such modus operandi in Bib\TeX:

\begin{verbatim}
{\relax Cl}{\MakeUppercase{i}ve Eric} Cussler
\end{verbatim}

\begin{verbatim}
{\relax Cl}{\MakeUppercase{i}ve Eric} Cussler
\end{verbatim}

6. In the first case, you have to use the ‘\emph{...}’ construct. It exists in \LaTeX, but it is preferable to use the \emph command. Concerning the ‘ß’ letter, the Con\TeXt command to get it is \ss, the namesake command in \LaTeX causes the case of ‘ß’ to be raised and the result is ‘SS’.

7. This part is also known as Lineage.

8. You can find them in Hufflen (2006).
but from our point of view, it is actually a dirty trick. In such a case, additional syntax to specify a non-standard abbreviation is missing in the .bib format. The solution of MiBiTTeX is more readable:

\begin{verbatim}
AUTHOR = {Clive Eric Cussler, abbr => Cl.}
\end{verbatim}

An alternative solution could be:

\begin{verbatim}
Cussler, Clive Eric, Cl.
\end{verbatim}

—an empty Junior part being specified between the first two commas—but such notation using three commas is incorrectly processed by BiBT\TeX{} and causes \texttt{biber} to crash.

2 More features

2.1 Difficultly with \LaTeX{} commands

If we go on with fields for person names, it is impossible to specify additional collaborators if you use only standard fields\(^9\). Either collaborator names are dropped out, or viewed as co-authors. You can add specific fields, but they will not be recognised by standard bibliography styles. The syntax put into action by MiBiTTeX just uses another connector, ignored by standard bibliography styles:

\begin{verbatim}
AUTHOR = {Robert Silverberg with Karen Huber}
\end{verbatim}

(another example is given by Mittelbach and Goossens (2004)).

Sometimes there should be no space between a particle and the \textit{Last} part:

\begin{verbatim}
Guy d’Antin
\end{verbatim}

As far as we know, there is no satisfactory solution for specifying this French name in BiB\TeX{}, in order for it to be typeset nicely. Let us go back to abbreviating first names, some authors retain their middle name when their first name is abbreviated, e.g.:

\begin{verbatim}
Henry Rider Haggard \rightarrow H. Rider Haggard
\end{verbatim}

Some books about \LaTeX{} and BiB\TeX{}—e.g., Mittelbach and Goossens (2004, Ch. 13)—propose the use of square brackets:

\begin{verbatim}
H[enry] Rider Haggard
\end{verbatim}

but only a few styles are able to handle them. Here also, MiBiTTeX’s additional syntax is unusable with ‘old’ BiB\TeX{}, but seems to us to be clearer:

\begin{verbatim}
Henry Rider Haggard, abbr => H. Rider
\end{verbatim}

\(^9\) Such specification is allowed within the bibliographies managed by the DocBook system (Walsh, 2010).

\begin{figure}
\centering
\begin{verbatim}
<personname>
  <first>Guy</first>
  <von space-after-f="no">d’</von>
  <last>Antin</last>
</personname>
\end{verbatim}
\caption{MiBiTTeX’s internal format.}
\end{figure}

2.2 Considerations about sorting

In Hufflen (2014), we mentioned that BiB\TeX{} was only able to perform lexicographic sorts, the \texttt{YEAR} field is just concatenated to other information. An analogous problem complicates the use of organism names as authors or editors. The following specification, using only a \textit{Last} part:

\begin{verbatim}
AUTHOR = {{\GuIT}}
\end{verbatim}

aims to put the q\texttt{It} logo of the Italian \TeX{} Users Group as the author of a collective work. Unfortunately, the commands at the beginning of a BiB\TeX{} token are pruned when such a token is used as a sort key, as done about the \texttt{\MakeUppercase} command mentioned in § 1. As a consequence, the sort key derived from this specification is empty.

3 The solutions of biblatex

The \texttt{biblatex} package did not extend the look of existing fields, but added new ones. This is successful for sort operations since some information can be redefined at the sort step. For example, the fields \texttt{SORTNAME} and \texttt{SORTYEAR} take precedence over the fields \texttt{AUTHOR} and \texttt{YEAR}. In particular, the example given in § 2.2 can be extended as follows:

\begin{verbatim}
AUTHOR = {{\GuIT}},
SORTNAME = {GuIT}
\end{verbatim}

whereas MiBiTTeX’s solution is:

\begin{verbatim}
AUTHOR = {org => \GuIT, sortingkey => GuIT}
\end{verbatim}

The other problems mentioned above are not handled by the \texttt{biblatex} package. In addition, let us remark that \texttt{biblatex} end-users also know some backward-compatibility problems if they are to revert to ‘old’ BiB\TeX{}. A simple example is given by the \texttt{DATE} field, which extends the specification of dates—you can specify not only a simple date, but also a \textit{range} of dates—; this field takes precedence over the predefined fields \texttt{YEAR} and \texttt{MONTH}. Obviously, these end-users prefer to use this \texttt{DATE} field, which increases the expressive power of .\texttt{bib} files, but its drawback is that it is not recognised by standard bibliography styles.
4 Discussion and conclusion

Bib\TeX successors have developed interesting extensions, but often these extensions are mutually incompatible. That is regrettable, but we may think that these experimentation steps should lead to a new format for bibliographical entries in the near future. Maybe the .bib format reached its limits and it is now difficult to extend it. So, the best solution for a new bibliography format could be based on another syntax, e.g., XML. In particular, this format should implement some \TeX command used within .bib files, in order to ease their translation into other formats, as mentioned in § 1. Obviously, it should provide solutions to problems described in § 2.

When we designed MiBib\TeX, we decided for such an internal format, XML-based. We have been able to define our extensions, implement the features of interest for us. Since we would like to get access to these extensions, we defined some concrete syntax, as extensions within .bib files. Most often, but not always. For example, we can specify that there should be no space between a name’s \textit{von} and \textit{Last} part in our internal format—as shown in Fig. 2—but we have not proposed yet some concrete syntax for this point within .bib files. If a new format is defined with a comparable expressive power, we think that our functions could be working; we would have just to translate this new format into our internal one. We hope that such a new standard for bibliography database files will be carried out. In the meantime, we show that some syntactic extensions of .bib files—even if they are not backward compatible—can lead to interesting and useful results.

Acknowledgements

I thank Claudio Beccari for his Italian translations of the abstract and keywords.

References


10. e\X\textit{ensible Markup Language.}

11. More precisely, we have not found yet a satisfying solution. Let us go back to Fig. 2, the \textit{spacing-after-f} attribute obviously defaults to \textit{yes}, that is, if a bibliography style orders the insertion of a space character after a \textit{von} part, this space character should not be removed.