MlBib\TeX{} 1.4: the New Version

Jean-Michel HUFFLEN
DISC — University of Franche-Comté
Trento, 17th October 2015
Contents

Bibliographies presently

\texttt{M\textsc{I}B\textsc{I}B\textsc{T}\textsc{E}X} experience

The new version

Customisation

Commands

Conclusion
Bibliographies presently

\texttt{BibTeX} is ageing. . .
Bibliographies presently

\texttt{BibTeX} is ageing. . .

biber usable with the biblatex package,
Bibliographies presently

\textsc{BibTeX} is ageing. . .

biber usable with the biblatex package,

interesting extensions, e.g., the DATE field:

\texttt{DATE = \{2015-10-17/2015-10-18\}}
Beyond Bib\TeX

What happens if an end-user of biblatex has to revert to Bib\TeX?
Beyond BibTEX

What happens if an end-user of biblatex has to revert to BibTEX?

What about ConTEXt?
Beyond Bib\TeX

What happens if an end-user of biblatex has to revert to Bib\TeX? What about Con\TeXt? After Bib\TeX $\leftarrow$ incompatible extensions.
MlBibTEX

Reimplementation of BIBTEX using the Scheme programming language,
MlBibTEX

Reimplementation of BIBTEX using the Scheme programming language, with particular focus on multilingual features, provides some syntactical extensions,
Reimplementation of Bib\TeX using the Scheme programming language, with particular focus on multilingual features, provides some syntactical extensions, based on an XML-like format for bibliographical items,
MlBibTEX

Reimplementation of BibTEX using the Scheme programming language,

with particular focus on multilingual features, provides some syntactical extensions,

based on an XML-like format for bibliographical items,

includes support for ConTEXT and biblatex,
MlBibTEX

Reimplementation of BIBTEX using the Scheme programming language, with particular focus on multilingual features, provides some syntactical extensions, based on an XML-like format for bibliographical items, includes support for ConTeXt and biblatex, and other applications.
MlBib\TeX

Reimplementation of BIB\TeX using the Scheme programming language, with particular focus on multilingual features, provides some syntactical extensions, based on an XML-like format for bibliographical items, includes support for Con\TeXt and biblatex, and other applications.

All the programs may take advantage of an extension (e.g., DATE).
Feedback

Not used widely, but users are satisfied, as far as I know.
Feedback

Not used widely, but users are satisfied, as far as I know.

Less permissive than BibTeX. For example, a YEAR field must be an integer,
Feedback

Not used widely, but users are satisfied, as far as I know.

Less permissive than \texttt{BIBTEX}. For example, a \texttt{YEAR} field \textit{must} be an integer, possibly negative,
Feedback

Not used widely, but users are satisfied, as far as I know.

Less permissive than \textsc{BibTeX}. For example, a \texttt{YEAR} field must be an integer, possibly negative, possibly \textit{inexact} (cf. \textsc{GUIT} 2014).
Why a new version?

Even if MiBibTeX 1.3 is based on a canonical representation of all accented letters, it only deals with Latin 1 encoding:
Why a new version?

Even if M\textsc{\textbf{BIB}}\textsc{T\textsc{E}}X 1.3 is based on a canonical representation of all accented letters, it only deals with Latin 1 encoding:

Łódź $\mapsto \{\texttt{L}\}\acute{o}\{z\}$
Why a new version?

Even if \texttt{M\/\textsc{libib}T\textsc{e}X} 1.3 is based on a canonical representation of all accented letters, it only deals with Latin 1 encoding:

\[
\text{\Lódź} \implies \text{"\L\'{o}\.{z}}
\]

whereas modern T\textsc{e}X engines—e.g., ConT\textsc{e}Xt—are based on \texttt{UTF-8}.
Why a new version?

Even if MiBIBTEX 1.3 is based on a canonical representation of all accented letters, it only deals with Latin 1 encoding:

\[ \text{Łódź} \implies \{\text{Ł}\}ó\{z\} \]

whereas modern TEX engines—e.g., ConTEXt—are based on UTF-8.

Now, Scheme’s new standard is Unicode-compliant, so MiBIBTEX can be, too.
Why a new version?

Even if MiBibTeX 1.3 is based on a canonical representation of all accented letters, it only deals with Latin 1 encoding:

Łódź $\Rightarrow \{\L\}ó\{z\}$

whereas modern \TeX engines—e.g., Con\TeX t—are based on UTF-8.

Now, Scheme’s new standard is Unicode-compliant, so MiBibTeX can be, too.

The interface with Scheme should be more customisable.
MlBibTEX 1.4 and encodings

First release \iffalse byte-based encodings will be processed: Latin-1, Latin-2, UTF-8. \fi
\texttt{M\rule{0pt}{1em}lBib\TeX} 1.4 and encodings

First release $\leftarrow$ byte-based encodings will be processed: Latin-1, Latin-2, UTF-8, ... but not UTF-16.
MlBib\TeX 1.4 and encodings

First release \iffalse byte-based encodings will be processed: Latin-1, Latin-2, UTF-8. . . but not UTF-16.

You can make precise the encoding at the beginning of a .bib file:

\%encoding = latin-1
M\textsc{mlBibTeX} 1.4 and encodings

First release \iffalse byte-based encodings will be processed: Latin-1, Latin-2, utf-8. . . but not utf-16.\fi

You can make precise the encoding at the beginning of a .bib file:

\texttt{\%encoding = latin-1}

(Another directive, \texttt{\%prefix}, allows name clashes to be avoided.)
Bibliography database files

.bib files, but also XML files.
Bibliography database files

.bib files, but also XML files.

JSON and Refer formats \iffalse\ equal \fi\ planned.
Rules for names

Accented letters will be allowed only in values associated with fields.
Rules for names

Accented letters will be allowed only in values associated with fields.

Field names (AUTHOR, ...) and entry types (@ARTICLE, ...) ⇒ the same rule holds.
Initialisation file

((encodings-pv 'set-default-4-bib-files)
 'utf-8)

(or accept the predefined default \[\Rightarrow\] latin-1).
Initialisation file

(((encodings-pv 'set-default-4-bib-files)
  'utf-8))

(or accept the predefined default \[\implies\] latin-1).

mlbibtex ← ~/.mlbibtexas
Initialisation file

((encodings-pv 'set-default-4-bib-files)
 'utf-8)

(or accept the predefined default \[=\] latin-1).

  mlbibtex ← ~/.mlbibtex
  mlbibcontext ← ~/.mlbibcontext

  ...
Revision of MLBibTeX’s commands

-encoding for the programs mlbibtex and mlbibtex2xml.
Revision of MlBibTEX’s commands

-encoding for the programs mlbibtex and mlbibtex2xml.

New argument for the program mlbiblatex.
Revision of \texttt{MlBibTeX}'s commands

- \texttt{-encoding} for the programs \texttt{mlbibtекс} and \texttt{mlbibtекс2xml}.

New argument for the program \texttt{mlbiblatex}.

No change for the program \texttt{mlbibcontext}, but the output defaults to the UTF-8 encoding.
Revision of \texttt{MlBib\LaTeX}'s commands

-encoding for the programs \texttt{mlbibtex} and \texttt{mlbibtex2xml}.

New argument for the program \texttt{mlbiblatex}.

No change for the program \texttt{mlbibcontext}, but the output defaults to the UTF-8 encoding.

The other options are still recognised, e.g., the -inexact option (cf. GUIT 2014).
What is to be done

Order relations.
What is to be done

Order relations.

Output routine, e.g.:

Łódź $\implies$ Łódź \hspace{1cm} (UTF-8)

$\implies$ {Łódź} (Latin-1)
What is to be done

Order relations.

Output routine, e.g.:

Łódź $\implies$ Łódź \hspace{1cm} (UTF-8)

$\implies$ \{Ł\}ó\{z\} \hspace{1cm} (Latin-1)

Parser.
Now... today

‘Semantic’ functions—including order relations—at the end of debugging.
Now... today

‘Semantic’ functions—including order relations—at the end of debugging.

Output routine ← in test.
Now... today

‘Semantic’ functions—including order relations—at the end of debugging.

Output routine ⇐ in test.

What is missing ⇒ parser and interface.
Now... today

‘Semantic’ functions—including order relations—at the end of debugging.

Output routine $\leftarrow$ in test.

What is missing $\rightarrow$ parser and interface.

Installation procedure $\rightarrow$ in refurbishment.
Conclusion

I have already reworked and extended M\texttt{BIBTEX}, . . . and succeeded. I am confident.

See you soon for 1.4’s first demonstration!