

Names in Bib_TE_X and MIBib_TE_X

Jean-Michel HUFFLEN

LIFC — University of Franche-Comté

TUG 2006

Marrakesh, November 2006

Contents

Why this talk

Name Components

Formatting Names

What to Know? What to Do?

Conclusion

Why this talk

AUTHOR = {Harry Harrison}

Why this talk

AUTHOR = {Harry Harrison}
First Last

Why this talk

AUTHOR = {Harry Harrison}
First Last

In practice:

users can deal with names' syntax intuitively in most cases

In other cases

Jean-Claude Smit-le-B\ ' {e}n\ ' {e}dicte

In other cases

Jean-Claude Smit-le-B\ ' {e}n\ ' {e}dicte

First *Last*

von

Handling person names

Not documented wholly.

Handling person names

Not documented wholly.

Uses workarounds: additional braces, insertion of \LaTeX commands.

Handling person names

Not documented wholly.

Uses workarounds: additional braces, insertion of \LaTeX commands.

Special characters vs other groups surrounded by braces.

Handling person names

Not documented wholly.

Uses workarounds: additional braces, insertion of \LaTeX commands.

Special characters vs other groups surrounded by braces.

Makes $\text{BIB}\TeX$ difficult to use for generating bibliographies w.r.t. output formats other than \LaTeX .

MLBibT_EX and BibT_EX

Reimplementation of BIBT_EX with particular focus on multilingual features.

MLBibT_EX and BibT_EX

Reimplementation of BIBT_EX with particular focus on multilingual features. Uses:

- XML as central formalism,

MLBibT_EX and BibT_EX

Reimplementation of BIBT_EX with particular focus on multilingual features. Uses:

- XML as central formalism,
- a variant of XSLT for bibliography styles.

MLBibT_EX and BibT_EX

Reimplementation of BIBT_EX with particular focus on multilingual features. Uses:

- XML as central formalism,
- a variant of XSLT for bibliography styles.

Aims to be a 'better' BIBT_EX.

MLBibT_EX and BibT_EX

Reimplementation of BIBT_EX with particular focus on multilingual features. Uses:

- XML as central formalism,
- a variant of XSLT for bibliography styles.

Aims to be a 'better' BIBT_EX.

'Old' bibliography styles still usable.

Implementing MIBibT_EX

Compatibility with BIBT_EX.

Implementing MIBibT_EX

Compatibility with BIBT_EX.

Studying BIBT_EX's behaviour.

Implementing MIBibT_EX

Compatibility with BIBT_EX.

Studying BIBT_EX's behaviour.

Improvements.

Tokens

End with a whitespace character or '-' or '~'.

Jean-Claude Smit-le-B\ '{e}n\ '{e}dicte

Tokens

End with a whitespace character or '-' or '~'.

Jean-Claude Smit-le-B\ '{e}n\ '{e}dicte

Tokens

End with a whitespace character or '-' or '~'.

Jean-Claude Smit-le-B\ '{e}n\ '{e}dicte

Additional separators are ruled out:

Jean- Pierre

Tokens

End with a whitespace character or '-' or '~'.

```
Jean-Claude Smit-le-B\ '{e}n\ '{e}dicte
```

Additional separators are ruled out:

```
Jean- Pierre  
-ky
```

Name fields

Syntax:

- *First von Last*
- *von Last, First*
- *von Last, Junior, First*

Additional rules

At least a *Last* part:

TUG y2006

tug y2006

Additional rules

At least a *Last* part:

TUG y2006

tug y2006

The *von* part is as large as possible:

Jean de la Fontaine du Bois Joli

Additional rules

At least a *Last* part:

TUG y2006

tug y2006

The *von* part is as large as possible:

Jean de la Fontaine du Bois Joli

_____von_____

Additional rules

At least a *Last* part:

TUG y2006

tug y2006

The *von* part is as large as possible:

Jean de la Fontaine du Bois Joli

_____von_____

No *von* part \implies no break before '-': Louis-Albert

Groups surrounded by braces

Special character \leftarrow the first character after the command:

```
{\relax max}
```

```
{\relax Max}
```

Groups surrounded by braces

Special character \Leftarrow the first character after the command:

```
{\relax max}
```

```
{\relax Max}
```

Other groups \Leftarrow supposed to begin with an uppercase letter:

```
{old chap}
```

Person names in MIBibT_EX

Using *keywords*:

`first => Gilles, von => d', last => Argyre`

Person names in MIBibT_EX

Using *keywords*:

`first => Gilles, von => d', last => Argyre`

Mixed notation allowed: `Robinson, first => Kim Stanley`

Person names in MIBibT_EX

Using *keywords*:

`first => Gilles, von => d', last => Argyre`

Mixed notation allowed: `Robinson, first => Kim Stanley`

Order is irrelevant, but repetition is not allowed:

`Kim Stanley, last => Robinson`

Organisation names in MIBibT_EX

org => \TUG 2006 at Marrakesh,
sortingkey => TUG 2006

. . . In XML

```
<!ENTITY % author-or-editor      "(name, (and, name)*,
                                   and-others?, (with, name)*,
                                   with-others?)">
<!ENTITY % sortingkey-possibly    "sortingkey CDATA #IMPLIED">
<!ENTITY % personname-components "(first?, von?, last, junior?)">
<!ENTITY % othername-components  "(#PCDATA | asitis | emph)*">

<!ELEMENT author      %author-or-editor;>
<!ELEMENT name        (personname | othername | ...)>
<!ELEMENT personname %personname-components;>
<!ELEMENT othername  %othername-components;>
<!ATTLIST othername  ... %sortingkey-possibly;>
```

Formatting names

$\mathcal{S}_1 \mathcal{I} \mathcal{S}_2$ format.name\$

\mathcal{S}_1 contains names,

\mathcal{S}_2 is a pattern,

\mathcal{I} is an index.

Subpatterns

f/ff, vv, ll, jj enclosed by braces.

Subpatterns

f/ff, vv, ll, jj enclosed by braces.

{...ff...} *Put before and after.*

{...} *Redefining internal separator.*

~ *One single space.*

~~ *Escape.*

Subpatterns

f/ff, vv, ll, jj enclosed by braces.

{...ff...} *Put before and after.*

{...} *Redefining internal separator.*

~ *One single space.*

~~ *Escape.*

Default separator: some whitespace characters replaced by '~'.

Examples

'Classical' way $\{ff\}\{vv\}\{11\}\{jj\}$

Examples

'Classical' way $\{ff\}\{vv\}\{11\}\{jj\}$

Building labels for
alpha' styles $\{v\}\{1\}$

Abbreviating

Retains the first letter or special character. Does not handle the other braced groups:

Philippe

Abbreviating

Retains the first letter or special character. Does not handle the other braced groups:

Philippe {Ph}ilippe

Abbreviating

Retains the first letter or special character. Does not handle the other braced groups:

```
Philippe    {Ph}ilippe    {\relax Ph}ilippe
```

Abbreviating

Retains the first letter or special character. Does not handle the other braced groups:

```
Philippe    {Ph}ilippe    {\relax Ph}ilippe
```

Does not put the last period.

Abbreviating a first name in MIBibT_EX

Using the abbr keyword:

```
first => Robert Ervin, abbr => Robert E.,  
last => Howard
```

Abbreviating a first name in MIBibT_EX

Using the abbr keyword:

```
first => Robert Ervin, abbr => Robert E.,  
last => Howard
```

with mixed notation:

```
Lyon Sprague de Camp, abbr => L. Sprague
```

Particles

Lyon Sprague de Camp

Gilles d'Argyre

Particles

Lyon Sprague de Camp

Gilles d'Argyre

Gilles d'\unskip Argyre \Leftarrow mind text.prefix\$!

Particles

Lyon Sprague de Camp

Gilles d'Argyre

Gilles d'\unskip Argyre \Leftarrow mind text.prefix\$!

Gilles d'\aftergroup\unskip Argyre \Leftarrow useless!

Gilles d'\aftergroup\ignorespaces Argyre \Leftarrow better but...

Particles

Lyon Sprague de Camp Gilles d'Argyre

Gilles d'\unskip Argyre \Leftarrow mind text.prefix\$!

Gilles d'\aftergroup\unskip Argyre \Leftarrow useless!

Gilles d'\aftergroup\ignorespaces Argyre \Leftarrow better but. . . let us consider '[Gilles d']Argyre'.

Point of view

The patterns of `format.name$` are very concise but may be unclear.

Point of view

The patterns of `format.name$` are very concise but may be unclear.

In fact, they serve two purposes:

- displaying names,
- building labels for some bibliography styles.

Point of view for MIBibT_EX

Simple cases \implies templates, like in XSLT.

Point of view for MIBibT_EX

Simple cases \implies templates, like in XSLT.

Tests \longleftarrow the last character of a string.

Point of view for $\text{MLBibT}_{\text{E}}\text{X}$

Simple cases \implies templates, like in XSLT.

Tests \longleftarrow the last character of a string.

Accented letters of Latin 1 directly handled. (Other special letters for next version.)

Point of view for MIBibT_EX

Simple cases \implies templates, like in XSLT.

Tests \longleftarrow the last character of a string.

Accented letters of Latin 1 directly handled. (Other special letters for next version.)

Writing Scheme function for most difficult cases

Point of view for MIBibT_EX

Simple cases \implies templates, like in XSLT.

Tests \longleftarrow the last character of a string.

Accented letters of Latin 1 directly handled. (Other special letters for next version.)

Writing Scheme function for most difficult cases (but I have not done that yet!)

Point of view for MIBibT_EX

Simple cases \implies templates, like in XSLT.

Tests \longleftarrow the last character of a string.

Accented letters of Latin 1 directly handled. (Other special letters for next version.)

Writing Scheme function for most difficult cases (but I have not done that yet!)

Modular approach \longleftarrow modules for handling names.

What to know

If braces are balanced within a string, `text.prefix$` function never yields unbalanced braces.

What to know

If braces are balanced within a string, `text.prefix$` function never yields unbalanced braces.

`change.case$` affects special characters:

$$\{\backslash 'e\} \implies \{\backslash 'E\}$$

What to know

If braces are balanced within a string, `text.prefix$` function never yields unbalanced braces.

`change.case$` affects special characters:

$$\{\backslash 'e\} \implies \{\backslash 'E\}$$

does not affect other 'braced' groups.

Warning

Characters strictly counted only by `substring$!`

Warning

Characters strictly counted only by `substring$!`

Do not rely the functions `text.prefix$` and `substring$`.

Beyond. . .

Defining other components for other kinds of names:
Russian names, Indian names. . .

Beyond. . .

Defining other components for other kinds of names:
Russian names, Indian names. . .

but only from the XML specification of MIBIBT_EX.

Ending

How to:

- display an XML tree resulting from parsing a .bib file,
- use the compatibility mode (for .bst files).

⇐ BachoT_EX 2006.

Using MIBibT_EX yourself

- under Linux,
- provided that you have a Scheme interpreter.