

# Making BachTeX Proceedings

Jean-Michel Hufflen

BachTeX 2024

2nd May 2024

Contents

Introduction

Our framework

Your article

Building a whole  
issue

Conclusion

Introduction

Our framework

Your article

Building a whole issue

Conclusion

# The origins

BachTeX 2023  $\Leftarrow$  I told you that I became the new editor  
of the journal of the French-speaking group *GUTenberg*.

# The origins

BachTeX 2023  $\Leftarrow$  I told you that I became the new editor of the journal of the French-speaking group *GUTenberg*.  
Development of programs for building the articles of a new issue.

# The origins

BachTeX 2023  $\Leftarrow$  I told you that I became the new editor of the journal of the French-speaking group *GUTenberg*.

Development of programs for building the articles of a new issue.

Using and enriching these programs for BachTeX proceedings.

# The situation

A long long time ago  $\Leftarrow$  *Plain* T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X and graphical extensions (sometimes anarchic).

Contents

Introduction

Our framework

Your article

Building a whole  
issue

Conclusion

# The situation

A long long time ago  $\Leftarrow$  *Plain* T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X and graphical extensions (sometimes anarchic).

Now  $\Leftarrow$  PDF has become standard for written documents, possibly including graphical effects.

Contents

Introduction

Our framework

Your article

Building a whole  
issue

Conclusion

# The situation

A long long time ago  $\Leftarrow$  *Plain* T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X and graphical extensions (sometimes anarchic).

Now  $\Leftarrow$  PDF has become standard for written documents, possibly including graphical effects.

But how to get such output?

Contents

Introduction

Our framework

Your article

Building a whole  
issue

Conclusion



# The situation

A long long time ago  $\Leftarrow$  *Plain* T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X and graphical extensions (sometimes anarchic).

Now  $\Leftarrow$  PDF has become standard for written documents, possibly including graphical effects.

But how to get such output?

pdfL<sup>A</sup>T<sub>E</sub>X    X<sub>3</sub>L<sup>A</sup>T<sub>E</sub>X    LuaL<sup>A</sup>T<sub>E</sub>X    ConT<sub>E</sub>Xt

(belonging to T<sub>E</sub>X's galaxy).

# The situation

A long long time ago  $\Leftarrow$  Plain T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X and graphical extensions (sometimes anarchic).

Now  $\Leftarrow$  PDF has become standard for written documents, possibly including graphical effects.

But how to get such output?

pdfL<sup>A</sup>T<sub>E</sub>X    X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X    LuaL<sup>A</sup>T<sub>E</sub>X    ConT<sub>E</sub>Xt

(belonging to T<sub>E</sub>X's galaxy).

*Each program might be used, excluding others.*

# Already?

TUGboat issues?  $\Leftarrow$  Some *scripts*.

Contents

**Introduction**

Our framework

Your article

Building a whole  
issue

Conclusion

# Already?

TUGboat issues?  $\Leftarrow$  Some *scripts*.

ConTeXt group?  $\Leftarrow$  done, but only from articles written  
using ConTeXt.

# Already?

TUGboat issues?  $\Leftarrow$  Some *scripts*.

ConTeXt group?  $\Leftarrow$  done, but only from articles written using ConTeXt.

*Cahiers GUTenberg*, at the XXIst century's beginning  $\Leftarrow$  *metadata* were collected automatically, but the *organisation* of a complete issue was done *manually*.

# Requirements/framework

- ▶ Each article could be given its *own processor*: pdfL<sup>A</sup>T<sub>E</sub>X, X<sub>Ǝ</sub>L<sup>A</sup>T<sub>E</sub>X, LuaL<sup>A</sup>T<sub>E</sub>X, ConT<sub>E</sub>Xt,

# Requirements/framework

- ▶ Each article could be given its *own processor*: pdf $\text{\LaTeX}$ , X $\text{\LaTeX}$ , Lua $\text{\LaTeX}$ , Con $\text{\TeX}$ t,
- ▶ each information computable *must* be *computed* or—at least—*checked*,

- ▶ Each article could be given its *own processor*: pdfL<sup>A</sup>T<sub>E</sub>X, X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X, LuaL<sup>A</sup>T<sub>E</sub>X, ConT<sub>E</sub>Xt,
- ▶ each information computable *must* be *computed* or—at least—*checked*,
- ▶ a complete issue is built using the pdfpages package,



- ▶ Each article could be given its *own processor*: pdfL<sup>A</sup>T<sub>E</sub>X, X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X, LuaL<sup>A</sup>T<sub>E</sub>X, ConT<sub>E</sub>Xt,
- ▶ each information computable *must* be *computed* or—at least—*checked*,
- ▶ a complete issue is built using the pdfpages package,
- ▶ Special feature: *two* tables of contents, in Polish (resp. English) at the beginning (resp. the end).

# When you are sending an article to me

The class to be used  $\Leftarrow$  `ltugproc.cls`.

# When you are sending an article to me

The class to be used  $\Leftarrow$  ltugproc.cls.

Multilingual aspects  $\Leftarrow$  resourcefulness, packages babel  
(polish/polish-compat) or polyglossia, package polski.

# When you are sending an article to me

The class to be used  $\Leftarrow$  ltugproc.cls.

Multilingual aspects  $\Leftarrow$  resourcefulness, packages babel  
(polish/polish-compat) or polyglossia, package polski.

Preferred encoding  $\Leftarrow$  UTF-8, but other choices are allowed.

# When you are sending an article to me

The class to be used  $\Leftarrow$  ltugproc.cls.

Multilingual aspects  $\Leftarrow$  resourcefulness, packages babel  
(polish/polish-compat) or polyglossia, package polski.

Preferred encoding  $\Leftarrow$  UTF-8, but other choices are allowed.

Graphical (PDF) output appreciated.

# The hbachotex package

It allows:

Making  
Bach $\text{\TeX}$   
Proceedings

Jean-Michel  
Hufflen

Contents

Introduction

Our framework

**Your article**

Building a whole  
issue

Conclusion

# The hbachotex package

It allows:

- ▶ the paper size to be unified (as A4),

# The hbachotex package

It allows:

- ▶ the paper size to be unified (as A4),
- ▶ headers and footers to be unified.



# The hbachotex package

It allows:

- ▶ the paper size to be unified (as A4),
- ▶ headers and footers to be unified.

Introducing new commands ‘\hb...’ or ‘\@hb@...’  
(internally).

# The hbachotex package

It allows:

- ▶ the paper size to be unified (as A4),
- ▶ headers and footers to be unified.

Introducing new commands ‘\hb...’ or ‘\@hb@...’  
(internally).

The lastpage package is loaded.

# The more-abstracts package

You can also load it, in which case *two* abstracts can be given:

```
\begin{abstract}
...           ⇐      in English (at first)
\end{abstract}
\begin{strszcz}
...           ⇐      in Polish (at second)
\end{strszcz}
```

# The more-abstracts package

You can also load it, in which case *two* abstracts can be given:

```
\begin{abstract}
...           ⇐      in English (at first)
\end{abstract}
\begin{strszcz}
...           ⇐      in Polish (at second)
\end{strszcz}
```

Watch out! (*Uwaga!*)

[Contents](#)

[Introduction](#)

[Our framework](#)

[Your article](#)

[Building a whole  
issue](#)

[Conclusion](#)

# The more-abstracts package

You can also load it, in which case *two* abstracts can be given:

```
\begin{abstract}
...           ⇐      in English (at first)
\end{abstract}
\begin{strszcz}
...           ⇐      in Polish (at second)
\end{strszcz}
```

Watch out! (*Uwaga!*) You are in charge of specifying the right language.

[Contents](#)

[Introduction](#)

[Our framework](#)

[Your article](#)

[Building a whole  
issue](#)

[Conclusion](#)

# How your article is changed

Putting commands `\hbsettitleinenglish` and  
`\hbsettitleinpolish`, used to build *tables of contents*.

Contents

Introduction

Our framework

**Your article**

Building a whole  
issue

Conclusion

# How your article is changed

Putting commands `\hbsettitleinenglish` and  
`\hbsettitleinpolish`, used to build *tables of contents*.  
Getting titles  $\Leftarrow$  commands `\hbttitleinenglish` and  
`\hbttitleinpolish`.

Contents

Introduction

Our framework

**Your article**

Building a whole  
issue

Conclusion

# How your article is changed

Putting commands `\hbsettitleinenglish` and  
`\hbsettitleinpolish`, used to build *tables of contents*.  
Getting titles  $\Leftarrow$  commands `\hbttitleinenglish` and  
`\hbttitleinpolish`.  
Putting the starting page number.

Contents

Introduction

Our framework

**Your article**

Building a whole  
issue

Conclusion



# How your article is changed

- Putting commands `\hbsettitleinenglish` and `\hbsettitleinpolish`, used to build *tables of contents*.
- Getting titles  $\Leftarrow$  commands `\hbttitleinenglish` and `\hbttitleinpolish`.
- Putting the starting page number.
- Adapting two abstracts with the `more-abstracts` package.

Contents

Introduction

Our framework

**Your article**

Building a whole  
issue

Conclusion

# How your article is changed

Putting commands `\hbsettitleinenglish` and `\hbsettitleinpolish`, used to build *tables of contents*.

Getting titles  $\Leftarrow$  commands `\hbttitleinenglish` and `\hbttitleinpolish`.

Putting the starting page number.

Adapting two abstracts with the `more-abstracts` package.

Solving some overfull and underfull boxes  $\Leftarrow$  sloppy mode, `microtype` package and `\textls` command, playing with letter spacing in Lua $\text{\LaTeX}$ .

Contents

Introduction

Our framework

**Your article**

Building a whole  
issue

Conclusion

# How your article is changed

Putting commands `\hbsettitleinenglish` and `\hbsettitleinpolish`, used to build *tables of contents*.

Getting titles  $\Leftarrow$  commands `\hbttitleinenglish` and `\hbttitleinpolish`.

Putting the starting page number.

Adapting two abstracts with the `more-abstracts` package.

Solving some overfull and underfull boxes  $\Leftarrow$  sloppy mode, `microtype` package and `\textls` command, playing with letter spacing in Lua $\text{\LaTeX}$ .

Reworking with double columns as far as possible, reworking the placement of figures or tables.

Contents

Introduction

Our framework

**Your article**

Building a whole  
issue

Conclusion

# A basic example

```
\hbsettitleinenglish{%  
  Rare books written and illuminated in scriptoria made  
  accessible}  
  
\hbsettitleinpolish{%  
  Pisane w~skryptoriach i~iluminowane książki opuszczają  
  półki bibliotecznych cymeliów}  
  
\title[\hbttitleinpolish]{%  
  \hbttitleinpolish\thanks{\seefigure{one}.}}  
  
\author{Andrzej TOMASZEWSKI}  
\address{...}  
\netaddress{\antispam{tomaszewski@\cdots.pl}}  
  
\setcounter{page}{68}
```

# Asking you

If I get something equivalent to what you joined (up to page numbers), it is OK. Otherwise, I contact author(s) for approval.

# And in ConTEXt?

New module: `tugboat-hbachotex`, available very soon.

# And in ConTEXt?

New module: `tugboat-hbachotex`, available very soon.  
`\StartAbstract...` `\StopAbstract`, but also:

# And in ConTEXt?

New module: `tugboat-hbachotex`, available very soon.

`\StartAbstract...` `\StopAbstract`, but also:

`\StartStreszczenie...` `\StopStreszczenie`



# The Swiss knife

Scheme function `g-retain-command`.

Show `hbachotex-demo.scm`.

Unix's `make` command:

```
target: dependency1 dependency2 ...  
command
```

Dependencies — which may be *Unix paths* or *phonies* — can be *generated*, that is current within programming languages such that C or C++.

# Building a volume

Show the file `h-all.tex`.

# Building a volume

Show the file `h-all.tex`.

In Scheme:

- ▶ generating dependencies and updating the Makefile file

# Building a volume

Show the file `h-all.tex`.

In Scheme:

- ▶ generating dependencies and updating the Makefile file  
(show `all-articles.scm`)

# Building a volume

Show the file `h-all.tex`.

In Scheme:

- ▶ generating dependencies and updating the Makefile file (show `all-articles.scm`);
- ▶ building the table(s) of contents;

# Building a volume

Show the file `h-all.tex`.

In Scheme:

- ▶ generating dependencies and updating the Makefile file (show `all-articles.scm`);
- ▶ building the table(s) of contents;
- ▶ checking page number' succession.

# Not inserted yet

Abstracts in Polish and English when complete papers are not provided  $\Leftarrow$  generated from an XML file by means of an XSLT stylesheet.



# Not inserted yet

Abstracts in Polish and English when complete papers are not provided  $\Leftarrow$  generated from an XML file by means of an XSLT stylesheet.

Used for this year's abstracts.

# Good — less good

No information redundancy: OK, but currently only at the top level  
(no for included files, in particular *graphic* files).

# Good — less good

No information redundancy: OK, but currently only at the top level

(no for included files, in particular *graphic* files).

Using the `lastpage` package  $\Leftarrow$  getting an article's last page,

# Good — less good

No information redundancy: OK, but currently only at the top level

(no for included files, in particular *graphic* files).

Using the `lastpage` package  $\Leftarrow$  getting an article's last page, but not for an output file generated by ConTeXt (although a program can compute the number of the pages of a PDF file).

# Conclusion

When I started this work, I have been slowed down by features related to Polish and by articles written using ConTeXt,

# Conclusion

When I started this work, I have been slowed down by features related to Polish and by articles written using ConTeXt, but adapting the framework put for *Cahiers GUTenberg* was not really difficult.

When I started this work, I have been slowed down by features related to Polish and by articles written using ConTeXt, but adapting the framework put for *Cahiers GUTenberg* was not really difficult.

If you have articles previously designed for BachTeX 2018, 2019, and 2023... please send them!

# Conclusion

When I started this work, I have been slowed down by features related to Polish and by articles written using ConTeXt, but adapting the framework put for *Cahiers GUTenberg* was not really difficult.

If you have articles previously designed for BachTeX 2018, 2019, and 2023... please send them!  
Before summer 2024.