expl3 and LaTeX3

Will Robertson & Frank Mittelbach
And the LaTeX3 Project

July 29, 2014, Portland

Outline

LaTeX3

expl3

Case changing

Outline

LaTeX3

expl3

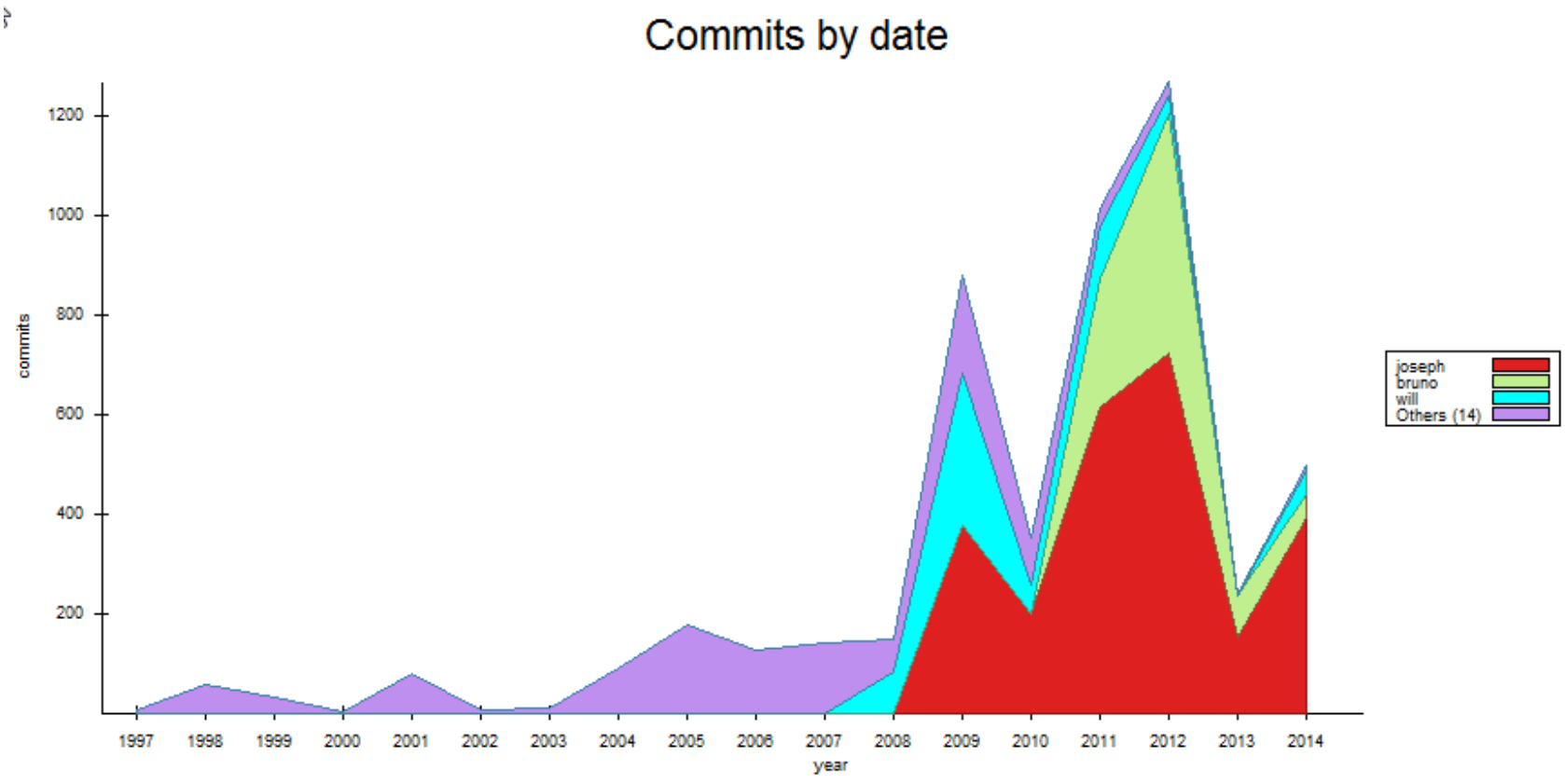
Case changing

LaTeX Project team members

Largely chronologically:

- Frank Mittelbach,
- Rainer Schöpf,
- Chris Rowley,
- David Carlisle,
- Michael Downes († 2003),
- Johannes Braams,
- Robin Fairbairns,
- Alan Jeffrey,
- Denys Duchier,
- Thomas Lotze,
- Morten Høgholm,
- Javier Bezos,
- Will Robertson,
- Joseph Wright, and
- Bruno Le Floch

LaTeX3 stats



What is LaTeX3?

- You know what LaTeX 2_ε is...(we assume)
- So LaTeX3 is the next version of LaTeX, right?
- Not so fast.

LaTeX 2ε status

- LaTeX 2ε must remain backwards compatible, warts and all.
- Many things that many people would change!
- Default document design:
  Some [many?] questionable/controversial aesthetics ...
- Programming:
  Not enough hooks, missing or unclean interfaces, separation of 'layers', default font encodings, ...

LaTeX 2ε status

- LaTeX 2ε must remain backwards compatible, warts and all.
- Many things that many people would change!
- Default document design:
  Some [many?] questionable/controversial aesthetics ...
- Programming:
  Not enough hooks, missing or unclean interfaces, separation of 'layers', default font encodings, ...

Explosion of packages doing similar things but each slightly differently and only parts of it...

LaTeX 2ε improvements?

- We *can/do* fix certain bugs in LaTeX 2ε — but not aspects that change layout or bugs that we know people worked around.
- More drastic changes can occur in `fixltx2e`, but that doesn't really work or solve the issue (see 'explosion of packages' earlier).
- But even seemingly 'harmless' changes have consequences.

LaTeX 2ε improvements?

- We *can/do* fix certain bugs in LaTeX 2ε — but not aspects that change layout or bugs that we know people worked around.
- More drastic changes can occur in `fixltx2e`, but that doesn't really work or solve the issue (see 'explosion of packages' earlier).
- But even seemingly 'harmless' changes have consequences.

Conclusion: In short, it just doesn't work.

What is LaTeX3?

- So we're not going to get rid of `latex` the format, and its interface is not going to change.
- That means whatever LaTeX3 is, it will be an alternative.

- The package concept means some LaTeX3 ideas can be layered on top of LaTeX 2_ε.

- Not everything can be layered (e.g. galley).
- In time, we will have a `latex3` format.

What is LaTeX3?

- So we're not going to get rid of `latex` the format, and its interface is not going to change.
- That means whatever LaTeX3 is, it will be an alternative.

- The package concept means some LaTeX3 ideas can be layered on top of LaTeX 2_ε.

- Not everything can be layered (e.g. galley).
- In time, we will have a `latex3` format.

N.B. LaTeX3 $\neq$ `expl3`

Outline

What is expl3?

- An interface to TEX programming, stabilised in the last five or so years.
- (Invented 1992.)
- It forms the programming/coding layer for LATEX3 but can be used independently:
  - for package writing on top of LATEX 2_ε,
  - for coding in other TEX formats; e.g., plain TEX, ConTEXt.

What is expl3?

Why not Lua?

- The first versions of expl3 appeared around the same time as Lua itself (1993).
- expl3 predates LuaTeX by some 20 years.
- expl3 supports pdfTeX, XeTeX, and LuaTeX, consistently.
- Also note that Lua doesn't always help.

What is expl3?

Why not Lua?

- The first versions of `expl3` appeared around the same time as Lua itself (1993).
- `expl3` predates LuaTeX by some 20 years.
- `expl3` supports pdfTeX, XeTeX, and LuaTeX, consistently.
- Also note that Lua doesn't always help.

And how would we use JSBox?

expl3 in LaTeX 2ε

The goal is to make it easier to write LaTeX packages:

- We eat our own dog food with `siunitx`, `fontspec`, etc. (this has formed the basis for iteration and solidification).
- More comprehensive than `etoolbox` &c.

`expl3` in LaTeX 2ε

The goal is to make it easier to write LaTeX packages:

- We eat our own dog food with `siunitx`, `fontspec`, etc. (this has formed the basis for iteration and solidification).
- More comprehensive than `etoolbox` &c.

All you plain users now in luck.

- `expl3` now loadable in plain TeX and even ConTeXt.
- This was done specially for 'generic' packages; specifically, Heiko Oberdiek asked us to provide this functionality to minimise variants of his packages.

expl3 is a success

acro	Interface for creating (classes of) acronyms
hobby	Hobby's algorithm in PGF/TiKZ for drawing optimally smooth curves.
chemmacros	Typesetting in the field of chemistry.
classics	Traditional-style citations for the classics.
conteq	Continued (in)equalities in mathematics.
ctex	A collection of macro packages and document classes for Chinese typesetting.
endiagram	Draw potential energy curve diagrams.
enotez	Support for end-notes.
exsheets	Question sheets and exams with metadata.
lt3graph	A graph data structure.
newlfm	The venerable class for memos and letters.
fnpct	Interaction between footnotes and punctuation.
GS1	Barcodes and so forth.
hobete	Beamer theme for the Univ. of Hohenheim.
kantlipsum	Generate sentences in Kant's style.
lualatex-math	Extended support for mathematics in LuaLaTeX.
media9	Multimedia inclusion for Adobe Reader.
pkgloader	Managing the options and loading order of other packages.
substances	Lists of chemicals, etc., in a document.
withargs	Ephemeral macro use.
xecjk	Support for CJK documents in XeLaTeX.
xpatch, regexpatch	Patch command definitions.
xpeek	Commands that peek ahead in the input stream.
xpinjin	Automatically add pinyin to Chinese characters
zhnumber	Typeset Chinese representations of numbers
zxjatype	Standards-conforming typesetting of Japanese for XeLaTeX.

expl3 is a success

acro	Interface for creating (classes of) acronyms
hobby	Hobby's algorithm in PGF/TiKZ for drawing optimally smooth curves.
chemmacros	Typesetting in the field of chemistry.
classics	Traditional-style citations for the classics.
conteq	Continued (in)equalities in mathematics.
ctex	A collection of macro packages and document classes for Chinese typesetting.
endiagram	Draw potential energy curve diagrams.
enotez	Support for end-notes.
exsheets	Question sheets and exams with metadata.
lt3graph	A graph data structure.
newlfm	The venerable class for memos and letters.
fnpct	Interaction between footnotes and punctuation.
GS1	Barcodes and so forth.
hobete	Beamer theme for the Univ. of Hohenheim.
kantlipsum	Generate sentences in Kant's style.
lualatex-math	Extended support for mathematics in LuaLaTeX.
media9	Multimedia inclusion for Adobe Reader.
pkgloader	Managing the options and loading order of other packages.
substances	Lists of chemicals, etc., in a document.
withargs	Ephemeral macro use.
xecjk	Support for CJK documents in XeLaTeX.
xpatch, regexpatch	Patch command definitions.
xpeek	Commands that peek ahead in the input stream.
xpinjin	Automatically add pinyin to Chinese characters
zhnumber	Typeset Chinese representations of numbers
zxjatype	Standards-conforming typesetting of Japanese for XeLaTeX.
copyediting	New!

What's new in the last six months?

- Joseph wrote `l3build`, which Frank covered yesterday.
- (Already mentioned that `expl3` now loads on plain.)
- Joseph and Bruno implemented expandable case switching.
- Will played around with something and Frank complained about it (auxiliary data).

Outline

LaTeX3

expl3

Case changing

Case changing

1. There is more to case changing than meets the eye:
   - Uppercase, lowercase
   - Titlecase (with language-dependent rules)
   - Case folding
2. Simple `\uppercase` and `\lowercase` are not sufficient!
   - Can have one-to-many mappings (ß → SS).
   - Can have many-to-one mappings (i, ı → I but also i → İ)
3. Unicode provides data, but is not providing a solution.

Case changing in regular TEX

TEX provides `\uppercase` and `\lowercase`:

```
\uppercase{%
  \def\mytitle{Some normal text}%
}
\mytitle
```

→ SOME NORMAL TEXT

The characters are not uppercased until the stomach.

I.e., case changing is not expandable.

This is the basis for `\MakeUppercase` in LATEX 2_ε, which has extra LICR-related code.

Case changing in LaTeX 2ε

From `source2e`:

> *These commands have some nasty features, such as uppercasing mathematics, environment names, labels, etc. A much better long-term solution is to use all-caps fonts, but these aren't generally available.* *

* A problem for `fontspec`?

Case changing in LaTeX 2ε

From `source2e`:

> *These commands have some nasty features, such as uppercasing mathematics, environment names, labels, etc. A much better long-term solution is to use all-caps fonts, but these aren't generally available.* *

* A problem for `fontspec`?

For `expl3`, we're not yet tackling this problem either.

The case-changing is intended to operate on 'characters' in token lists without discrimination.

What else are `\uppercase` & `\lowercase` used for?

`expl3` has long had `\tl_to_(upper/lower)case:n` and we needed to deprecate them!

We need to distinguish three main features:

1. Text manipulation in section titles, running headers, &c.
2. Normalizing (folding) text for sorting or filename searching etc.
3. Doing tricks with TEX programming.

Only one of these relates to typesetting!

Case changing for 'real' text input is a **hard** problem; not yet addressed.

Subsection 1

Case changing for programming

Case ‘folding’

We’ll cover programming first because it’s simplest. Quoting unicode.org:

> *Case folding is primarily used for caseless comparison of text, such as identifiers in a computer program, rather than actual text transformation.*
>
> *Case folding in Unicode is based on the lowercase mapping, but includes additional changes to the source text to help make it language-insensitive and consistent.*
>
> *As a result, case-folded text should be used solely for internal processing and generally should not be stored or displayed to the end user.*

Case folding examples

ASCII:

`\str_fold_case:n { ABCdef }` → abcdef

Greek sigma variants:

`\str_fold_case:n { σςΣ }` → σσσ

Deprecated ligature glyphs:

`\str_fold_case:n { ﬁ ﬅ }` → `fi st`

Implementation detail

Can't blindly compare for the 1000s of characters in Unicode.

From l3unicode-data.def:

```
\tl_const:cn { c__tl_lower_2_3_tl } { ŃńƧƨɅʌƐɛꞪɦΆάⓏⓩKk }
\tl_const:cn { c__tl_lower_2_4_tl } { ȌȍMµÈèѤѥḬḭẐẑỴỵẠạΠπꙢꙣꝈꝉⅬⅼ }
\tl_const:cn { c__tl_lower_2_5_tl } { ŅņƩʃƝνËëḨḩΫϋMm }
\tl_const:cn { c__tl_lower_2_6_tl } { ÔôΞξЂђѦѧȊȋẒẓỶỷꝐꝑꞨꞩѲѳNn }
\tl_const:cn { c__tl_lower_2_7_tl } { ŇňOoҐґӋӌΫϋOo }
\tl_const:cn { c__tl_lower_2_8_tl } { ƬƭȐȑПпƐɛѨѩḰḱẔẕỸỹꞒꞓӜӝƠơPp }
\tl_const:cn { c__tl_lower_2_9_tl } { PρSsӍӎѠѡΫϋQq }
\tl_const:cn { c__tl_lower_3_0_tl } { ŊŋƮʈȒȓIiӾӿꝐꝑḲḳⅡⅱTτꟼꟽꝎꝏRr }
\tl_const:cn { c__tl_lower_3_1_tl } { ƯưΣσȈȉꝖꝗΫϋSs }
\tl_const:cn { c__tl_lower_3_2_tl } { ŌōȔȕTτJjѪѫĂăꝘꝙḴḵƂƃYγꙀꙁꝒꝓTt }
\tl_const:cn { c__tl_lower_3_3_tl } { ƱʊYυЉљѢѣUu }
\tl_const:cn { c__tl_lower_3_4_tl } { ŎŏƲʋȖȗꝊꝋЊњǮǯÄäꝘꝙḶḷꝠꝡѲѳꙐꙑꝒꝓVv }
\tl_const:cn { c__tl_lower_3_5_tl } { YyXχĦħꝞꝟWw }
```

Subsection 2

Case changing for typesetting

Expandable case changing

Currently ONLY catering for plain Unicode text
(i.e., more work is needed.)

```
\tl_set:Nx \g_my_title_tl
  { \tl_upper_case:n {Some~ normal~ text} }
\g_my_title_tl
```

→ SOME NORMAL TEXT

Expandable case changing

Braces ‘hide’ content:

```
\tl_set:Nx \g_my_title_tl
  { \tl_upper_case:n {Some~ {normal}~ text} }
\g_my_title_tl
```

→ SOME normal TEXT

Multilingual in XeLaTeX/ LuaLaTeX

`\tl_upper_case:n { åéîøдα }` → ÅÉÎØДA
`\tl_lower_case:n { ΎΩƐ }` → ώωε

Language support:

`\tl_upper_case:n { Ragıp Hulûsi Özdem }`

→ RAGIP HULÛSIÖZDEM

`\tl_upper_case:nn {tr} { Ragıp Hulûsi Özdem }`

→ RAGIP HULÛSİ ÖZDEM

Mixed case

Towards automatic sentence formatting

Note this is *not* intended to iterate over words in a sentence.

Only the *very first* 'character' (besides exceptions such as quotes) is uppercased.

- `\tl_mixed_case:n {frank}` →
- `\tl_mixed_case:n {``frank''}` →
- `\tl_mixed_case:nn {ne} {ijsje}` →
- `\tl_mixed_case:n {THIS IS AN UPPERCASE TITLE}` →

Mixed case

Towards automatic sentence formatting

Note this is *not* intended to iterate over words in a sentence.

Only the *very first* 'character' (besides exceptions such as quotes) is uppercased.

- `\tl_mixed_case:n {frank}` → Frank
- `\tl_mixed_case:n {``frank''}` →
- `\tl_mixed_case:nn {ne} {ijsje}` →
- `\tl_mixed_case:n {THIS IS AN UPPERCASE TITLE}` →

Mixed case

Towards automatic sentence formatting

Note this is *not* intended to iterate over words in a sentence.

Only the *very first* 'character' (besides exceptions such as quotes) is uppercased.

- `\tl_mixed_case:n {frank}` → Frank
- `\tl_mixed_case:n {``frank''}` → “Frank”
- `\tl_mixed_case:nn {ne} {ijsje}` →
- `\tl_mixed_case:n {THIS IS AN UPPERCASE TITLE}` →

Mixed case

Towards automatic sentence formatting

Note this is *not* intended to iterate over words in a sentence.

Only the *very first* 'character' (besides exceptions such as quotes) is uppercased.

- `\tl_mixed_case:n {frank}` → Frank
- `\tl_mixed_case:n {``frank''}` → “Frank”
- `\tl_mixed_case:nn {ne} {ijsje}` → IJsje
- `\tl_mixed_case:n {THIS IS AN UPPERCASE TITLE}` →

Mixed case

Towards automatic sentence formatting

Note this is *not* intended to iterate over words in a sentence.

Only the *very first* 'character' (besides exceptions such as quotes) is uppercased.

- `\tl_mixed_case:n {frank}` → Frank
- `\tl_mixed_case:n {``frank''}` → “Frank”
- `\tl_mixed_case:nn {ne} {ijsje}` → IJsje
- `\tl_mixed_case:n {THIS IS AN UPPERCASE TITLE}` → This is an uppercase title

Extending mixed case to title case

Not a ‘token list’ function.

- `THIS IS AN UPPERCASE TITLE`
  → This is an Uppercase Title

Lots of edge cases! Style guides differ:

- Variable exception list:
  a an and as at but by en for if in of on or the to v via vs
- Modern words like ‘iPhone’ and ‘eyeTV’
- Always capitalise first and last words regardless of other rules

Anyway, not impossible, but part of some future ‘text processing’ module.

Subsection 3

Using weird tokens

TEX programming tricks

```
\begingroup
\lccode`\~=`\_
\lowercase{
  \endgroup
  \def~{\sb}
}
\mathcode`\_="8000\relax
\catcode`\_=12\relax
x_2 \quad $x_2$
```

→ x_2 x_2

TEX programming tricks

```
\begingroup
  \catcode`P=12
  \catcode`T=12
  \lowercase{
    \def\x{\def\rem@pt##1.##2PT{##1\ifnum##2>\z@.##2\fi}}
  }
  \expandafter\endgroup\x
\def\strip@pt{\expandafter\rem@pt\the}
```

Anything better with expl3?

- Potential wrapper around `\lowercase`.
- Not entirely decided upon yet.

```
\char_set_catcode_active:N \*
\tl_transform:nn
  { \char_transform:NN \* \_ }
  { \cs_set:Npn * { \sb } }
```

Of course, for something like this we also have candidate function `\char_set_active:Npn`.

Anything better with expl3?

```
\tl_transform:nn
  {
    \char_set_catcode_other:N \P
    \char_set_catcode_other:N \T
    \char_transform:NN \P \p
    \char_transform:NN \T \t
  }
  {
    \cs_set:Npn \__dim_to_decimal:w ##1.##2 PT
      { ##1 \int_compare:nT { ##2 > 0 } { .##2 } }
  }
```

`\__dim_to_decimal:w` used to define `\dim_to_decimal:n`.