expltools

development tools for expl3 programmers

```
\platy_examine_evidence:V
\c_false_alibi_bool
```

vít starý novotný (CSTUG)

tug 2025, trivandrum

july 19, 2025

introduction static analysis

idea automatically determine if a tex program is correct

problem *rice's theorem* semantic properties of programs are undecidable

conclusion

can’t be done

thank you for your attention?

introduction static analysis *ii*

idea automatically determine if a tex program is correct

problem 1 *rice's theorem* semantic properties of programs are undecidable
solutions *static analysis* annotate semantics in program syntax
dynamic analysis execute program to determine semantics

introduction expl3 *i*

idea automatically determine if a tex program is correct

problem 1 *rice's theorem* semantic properties of programs are undecidable
solution *static analysis* annotate semantics in program syntax

problem 2 tex's syntax is minimal and dynamic

```
\def\foo{...}  % a function, variable, or constant? what type is it? 🤔
\catcode`\<=0 <catcode`\ =0<catcode`\>=2<catcode`\==1 <def foo=...>  % 🤯
```

introduction expl3 *ii*

idea automatically determine if a tex program is correct

problem 1 *rice's theorem* semantic properties of programs are undecidable
solution *static analysis* annotate semantics in program syntax

problem 2 tex's syntax is minimal and dynamic

```
\def\foo{...}  % a function, variable, or constant? what type is it? 🤔
\catcode`\<=0 <catcode`\ =0<catcode`\>=2<catcode`\==1 <def foo=...>  % 🤯
```

solution *expl3*[1, 2] a restricted and richly annotated subset of tex, >10% of all tex code[3]

```
\cs_new:Nn \foo:nnnn { ... }  % a public function with four arguments
\int_set:Nn \l_foo_int { ... }  % a local variable of type int
```

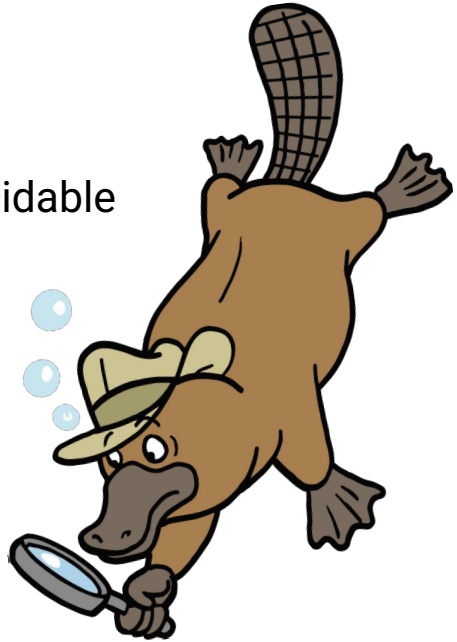
1 texdev.net/2024/11/11/the-mythical-latex3 2 ctan.org/pkg/l3kernel 3 witiko.github.io/Expl3-Linter-5

introduction explcheck

idea automatically determine if a tex program is correct

problem 1 *rice's theorem* semantic properties of programs are undecidable
solution *static analysis* annotate semantics in program syntax

problem 2 tex's syntax is minimal and dynamic
solution *expl3* a restricted and richly annotated subset of tex
explcheck a static analyzer (aka *linter*) of expl3 programs[4, 5]
platy explcheck mascot, an underwater detective, cc-by[6]
idea: paulo cereda, artist: greg (quickcartoons.com)



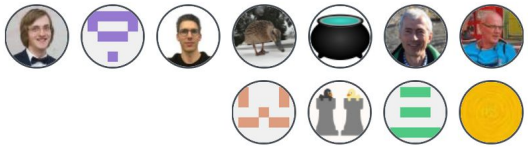
4 witiko.github.io/tag/expl3 (april) 5 tug.org/tc/devfund/documents/2024-09-expltools.pdf (september)
6 github.com/witiko/expltools/tree/main/artwork

introduction expltools

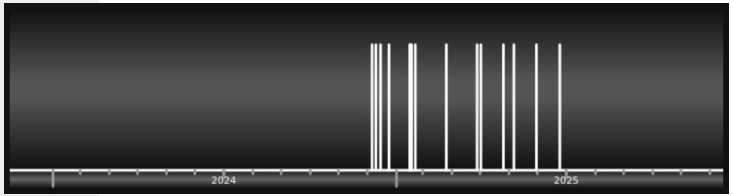
this bundle introduces `explcheck`, a static analysis tool for developers working with `expl3` code
currently in its initial release, `explcheck` aims to help developers identify issues and improve code quality
in the future, this bundle may expand to include additional development tools for `expl3`

sources	`/support/expltools`
documentation	readme.md list of warnings and errors project proposal
support	https://github.com/witiko/expltools/discussions
bug tracker	https://github.com/witiko/expltools/issues
announcements	https://github.com/witiko/expltools/releases
repository	https://github.com/witiko/expltools
version	2025-06-24
licenses	gnu general public license, version 2 or newer the latex project public license 1.3c
copyright	2024–2025 vít starý novotný

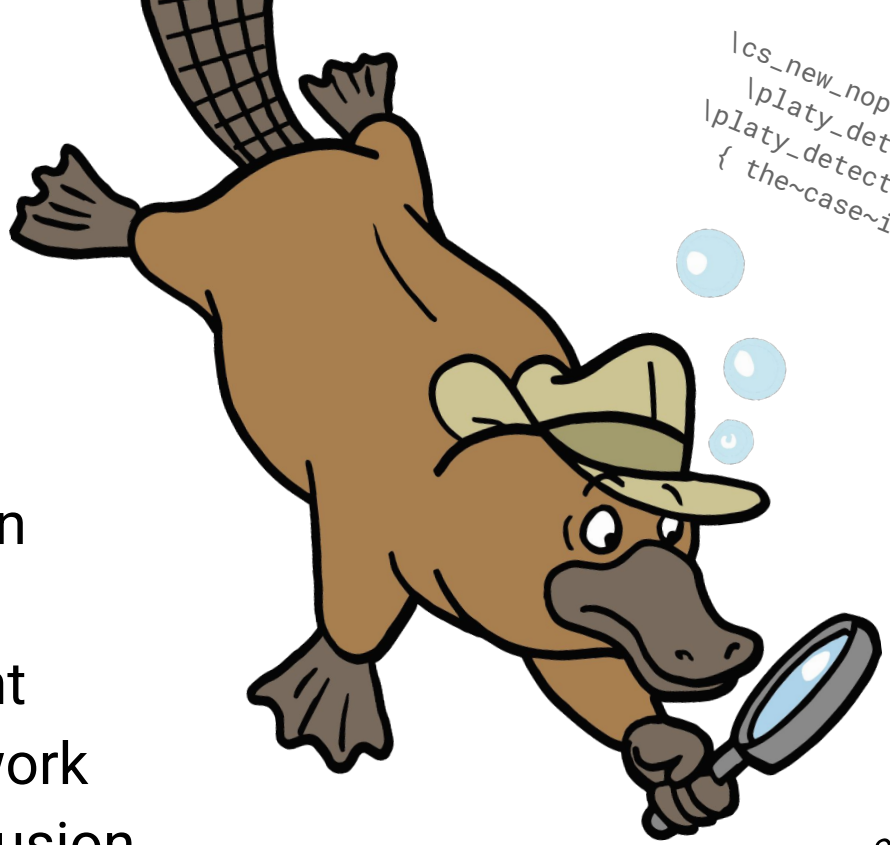
contributors 11



announcements



contents

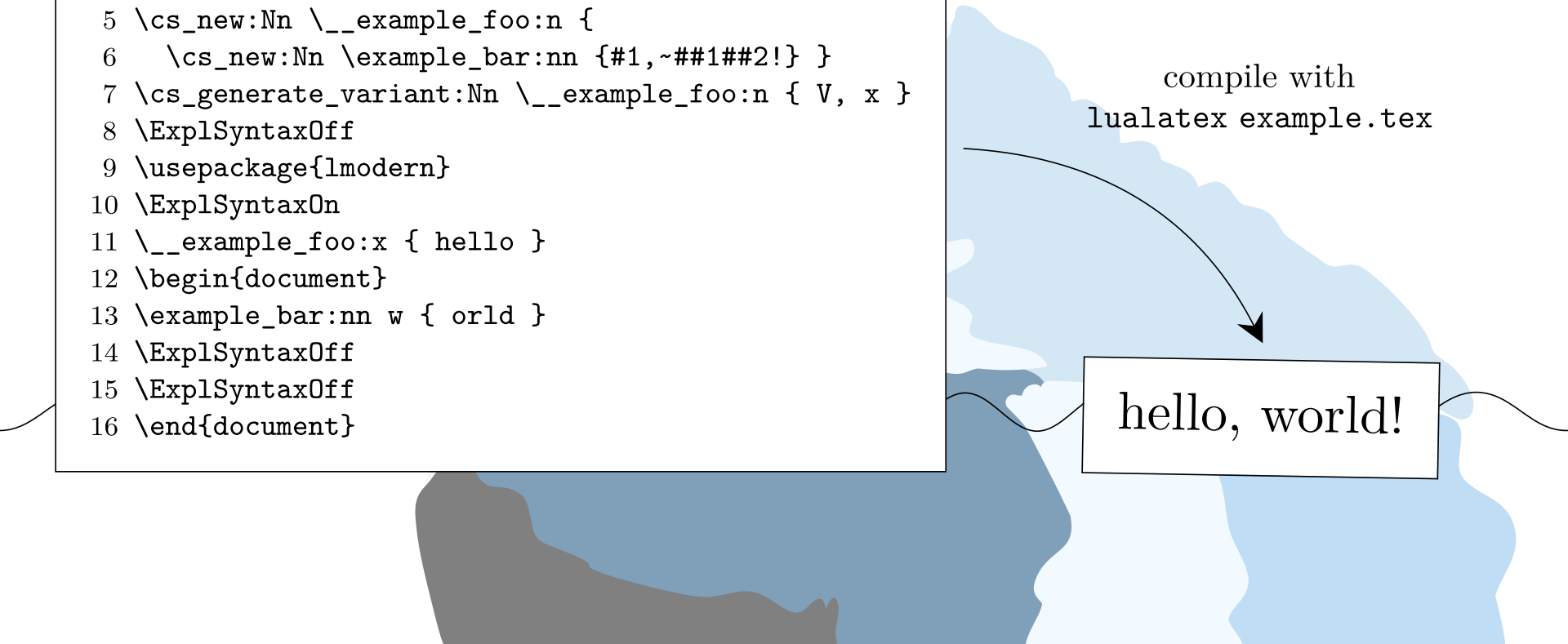


```
\cs_new_nopar:Nn
  \platy_detective_report:n { #1 }
\platy_detective_report:n
{ the~case~is~closed. \par }
```

```
\cs_generate_variant:Nn
  \platy_mission_done:
  { no }
```

```
\documentclass{standalone}
\ExplSyntaxOn
\iow_new:N \l_example_iow
\iow_open:Nn \l_example_iow { example.idx }
\cs_new:Nn \__example_foo:n {
  \cs_new:Nn \example_bar:nn {#1,~##1##2!} }
\cs_generate_variant:Nn \__example_foo:n { V, x }
\ExplSyntaxOff
\usepackage{lmodern}
\ExplSyntaxOn
\__example_foo:x { hello }
\begin{document}
\example_bar:nn w { orld }
\ExplSyntaxOff
\ExplSyntaxOff
\end{document}
```

compile with
`lualatex example.tex`



```
16 \end{document}
```

analyze with
`explcheck example.tex`

```
 3 \iow_new:N \l_example_iow
 4 \iow_open:Nn \l_example_iow { example.idx }
 5 \cs_new:Nn \__example_foo:n {
 6   \cs_new:Nn \example_bar:nn {#1,~##1##2!} }
 7 \cs_generate_variant:Nn \__example_foo:n { V, x }
```

```
11 \__example_foo:x { hello }
12 \begin{document}
13 \example_bar:nn w { orld }
```

15 \ExplSyntaxOff

(a) preprocessing identifies two expl3 parts and warns about an unnecessary command `\ExplSyntaxOff` on line 15

hello, world!

iow_new:N l_example_iow
iow_open:Nn l_example_iow {$_1$ e$_{11}$ x$_{11}$ a$_{11}$ m$_{11}$ p$_{11}$ l$_{11}$ e$_{11}$.$_{12}$ i$_{11}$ d$_{11}$ x$_{11}$ }$_2$ cs_new:Nn __example_foo:n {$_1$ cs_new:Nn example_bar:nn {$_1$ #$_6$ 1$_{12}$,$_{12}$ ␣$_{10}$ #$_6$ #$_6$ 1$_{12}$ #$_6$ #$_6$ 2$_{12}$!$_{12}$ }$_2$ }$_2$ cs_generate_variant:Nn __example_foo:n {$_1$ V$_{11}$,$_{12}$ x$_{11}$ }$_2$

__example_foo:x {$_1$ h$_{11}$ e$_{11}$ l$_{11}$ l$_{11}$ o$_{11}$ }$_2$ begin {$_1$ d$_{11}$ o$_{11}$ c$_{11}$ u$_{11}$ m$_{11}$ e$_{11}$ n$_{11}$ t$_{11}$ }$_2$ example_bar:nn w$_{11}$ {$_1$ o$_{11}$ r$_{11}$ l$_{11}$ d$_{11}$ }$_2$

(b) lexical analysis identifies 68 tokens and warns about missing whitespace around braces ({, }) on lines 6 and 12

c1 call `\iow_new:N` with the argument `\l_example_iow`

c2 call `\iow_open:Nn` with arguments `\l_example_iow` and `{ example.idx }`

c3 call `\cs_new:Nn` with arguments `\__example_foo:n` and `{ \cs_new:Nn ... }`

c4 call `\cs_generate_variant:Nn` with `\__example_foo:n` and `{ V, x }`

c5 call `\__example_foo:x` with `{ hello }`

c6 call `\example_bar:nn` with `w` and `{ orld }`

(c) syntactic analysis identifies six expl3 calls and warns about missing braces around the `n`-type argument `w` on line 13

s1 reserve a stream `\l_example_iow`

s2 open a file `example.idx` for writing

s3 define a function `\__example_foo:n`

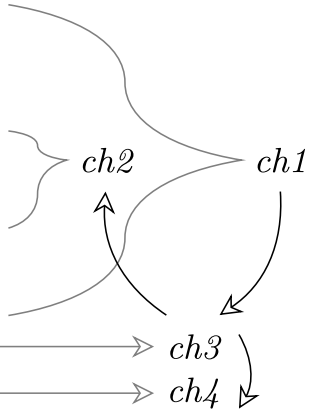
s4 define `\__example_bar:nn` with text ⟨*argument*⟩`,␣`⟨*arguments*⟩`!`

s5 generate variant `\__example_foo:V`

s6 generate variant `\__example_foo:x`

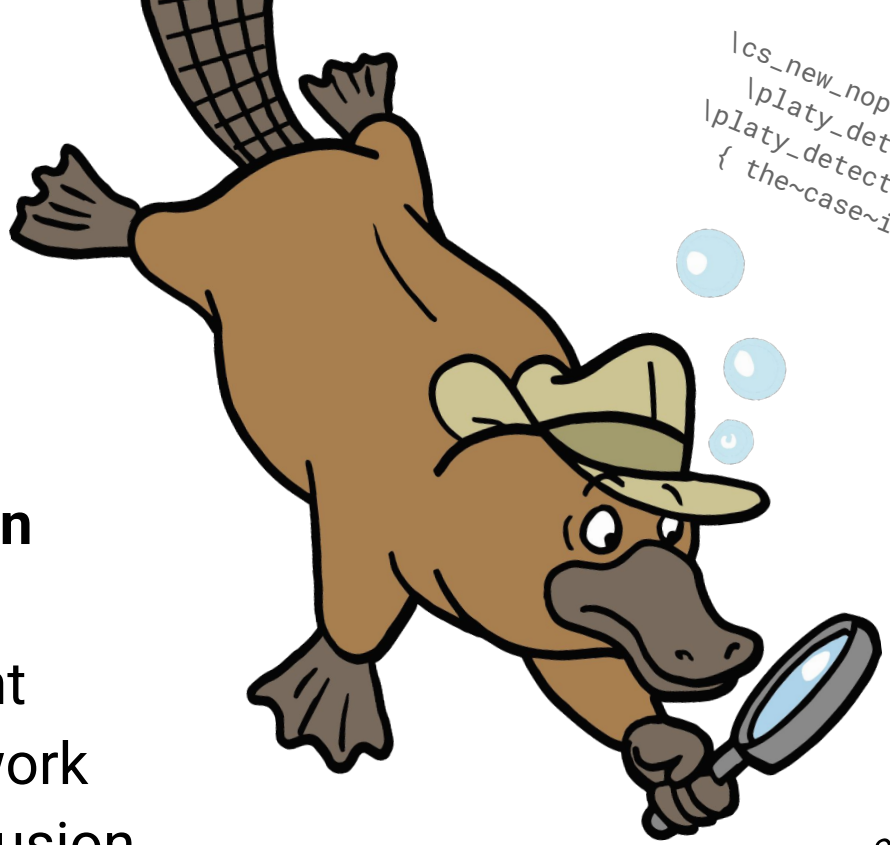
s7 call `\__example_foo:x`

s8 call `\__example_bar:nn`



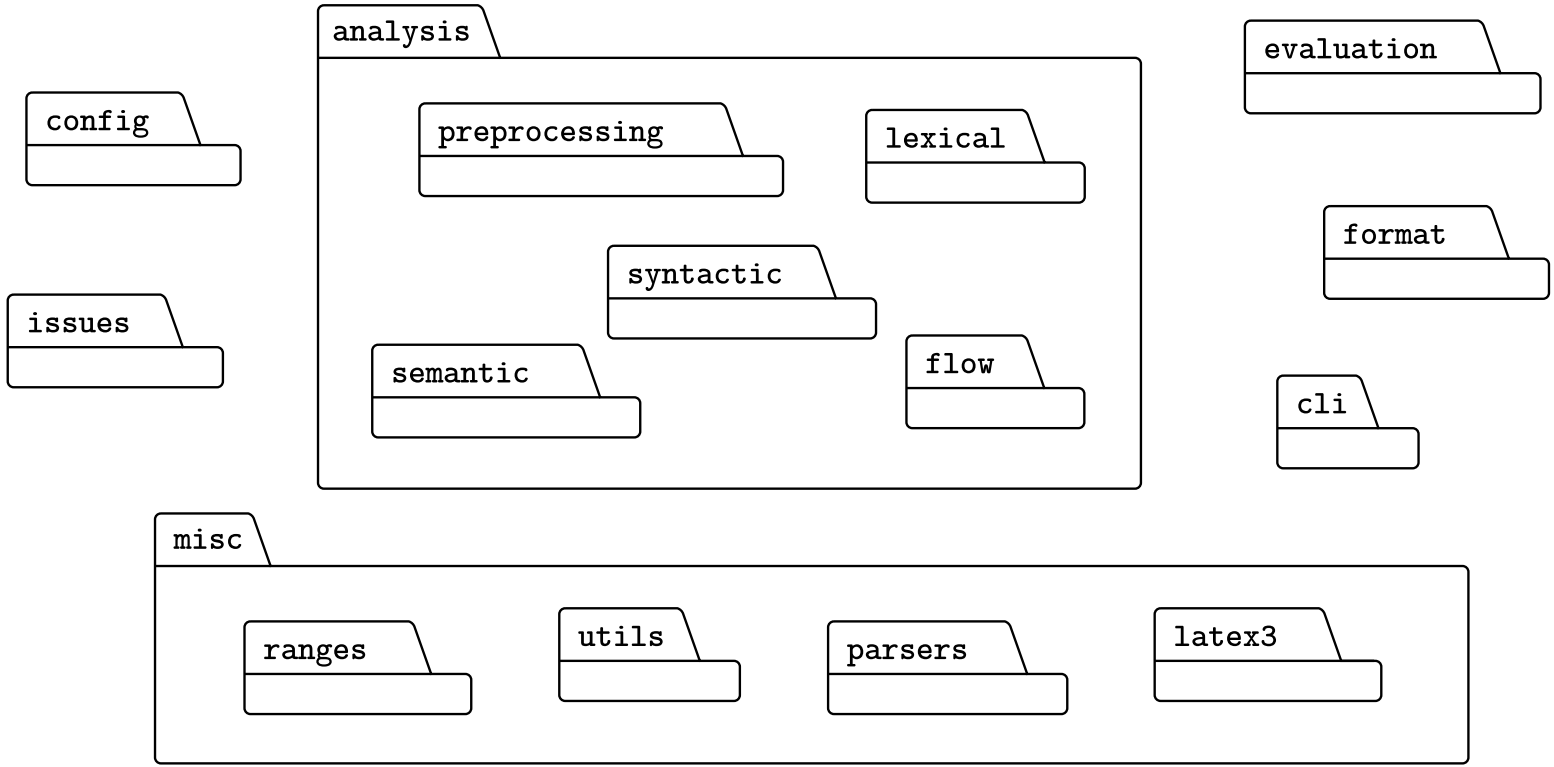
(d) semantic analysis identifies eight statements and warns about an unused variant from line 7; flow analysis identifies four chunks and warns about an unclosed stream from line 4

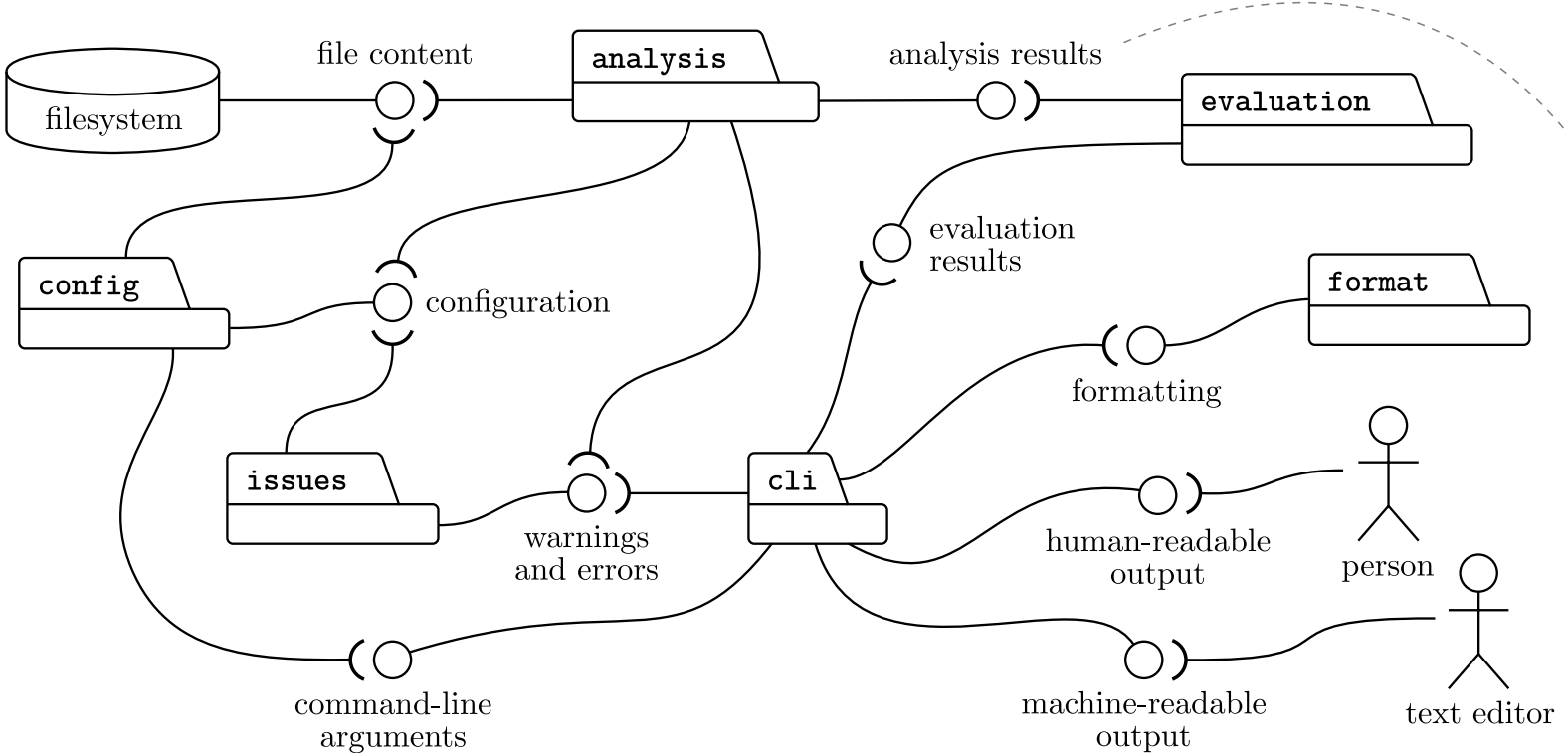
contents

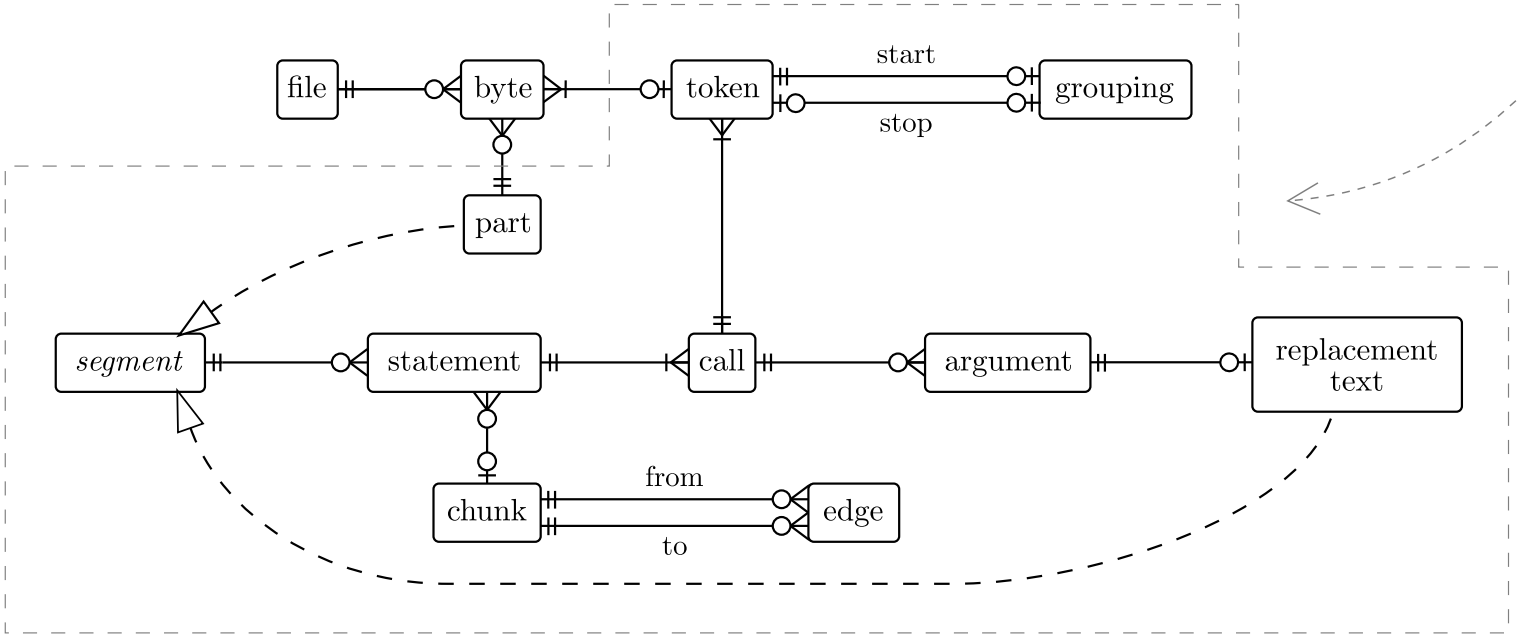


```
\cs_new_nopar:Nn
  \platy_detective_report:n { #1 }
\platy_detective_report:n
{ the~case~is~closed. \par }
```

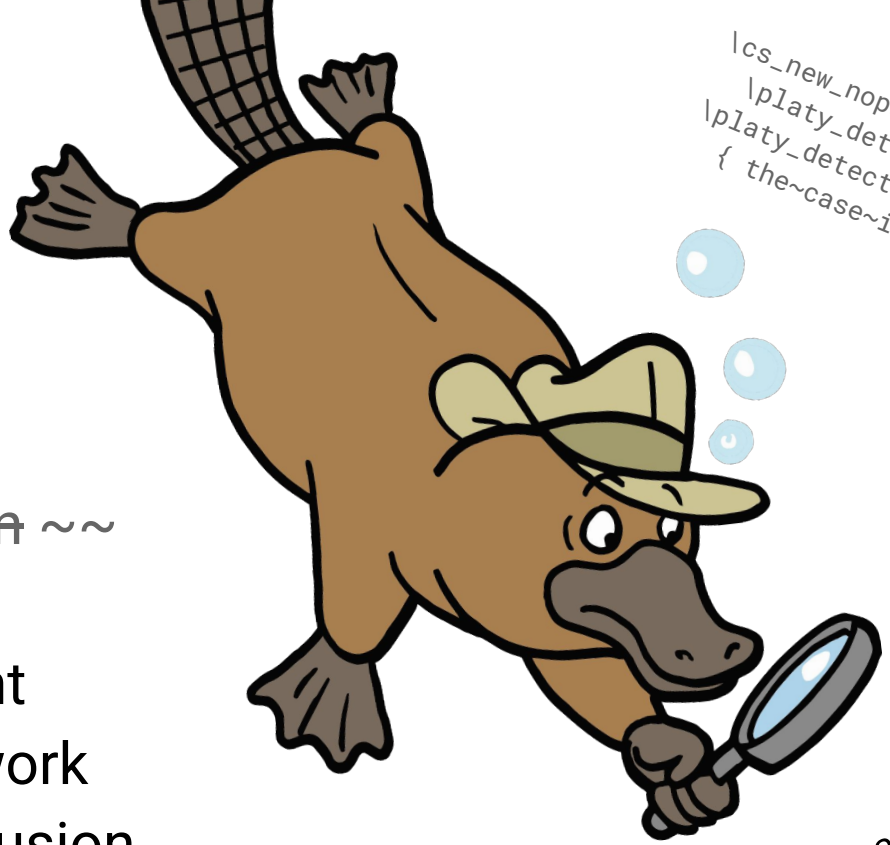
```
\cs_generate_variant:Nn
  \platy_mission_done:
  { no }
```


analysis
preprocessing
lexical
syntactic
semantic
flow
config
issues
evaluation
format
cli
misc
ranges
utils
parsers
latex3


filesystem
file content
analysis
analysis results
evaluation
config
configuration
evaluation
results
format
formatting
issues
warnings
and errors
cli
human-readable
output
person
command-line
arguments
machine-readable
output
text editor


file
byte
token
start
stop
grouping
part
segment
statement
call
argument
replacement
text
chunk
from
to
edge

contents



```
\cs_new_nopar:Nn
  \platy_detective_report:n { #1 }
\platy_detective_report:n
{ the~case~is~closed. \par }
```

```
\cs_generate_variant:Nn
  \platy_mission_done:
  { no }
```

expl3 issues

10 entries per page

search:

filename	# errors	# warnings
texmf-dist/tex/latex/linguistix/linguistix-ipa.sty	6	0
texmf-dist/tex/latex/pgf-go/pgf-go-stones.sty	6	2
texmf-dist/tex/latex/pgf-go/pgf-go-profiles.sty	4	0
texmf-dist/tex/latex/cooking-units/cooking-units.sty	3	43
texmf-dist/tex/latex/pgf-go/pgf-go-marks.sty	3	1
texmf-dist/tex/latex/robust-externalize/robust-externalize.sty	3	13
texmf-dist/tex/latex/linguistix/linguistix-logos.sty	2	0
texmf-dist/tex/latex/fontawesome6/fontawesome6.sty	2	1
texmf-dist/tex/latex/nwejm/nwejmart.cls	2	40
texmf-dist/tex/latex/nwejm/nwejm.cls	2	53

showing 1 to 10 of 525 entries (filtered from 1,041 total entries)

« ‹ 1 2 3 4 5 … 53 › »

(almost) all expl3 functions

vít starý novotný

july 8, 2025

abstract

this document uses the expltools bundle to generate a list of (almost) all expl3 functions defined in current tex live

contents

function list

function \aac_symbolEntry:nn

function \aac_symbolEntry:nn is defined in these files:

1. /usr/local/texlive/2025/texmf-dist/tex/latex/musicography/musicography.sty

function \aac_symbolLoop:nn

function \aac_symbolLoop:nn is defined in these files:

1. /usr/local/texlive/2025/texmf-dist/tex/latex/musicography/musicography.sty

function __acceptor_data_add:nnn

function __acceptor_data_add:nnn is defined in these files:

1. /usr/local/texlive/2025/texmf-dist/tex/latex/mitthesis/mitthesis.cls

function __acro_absolute_page:w

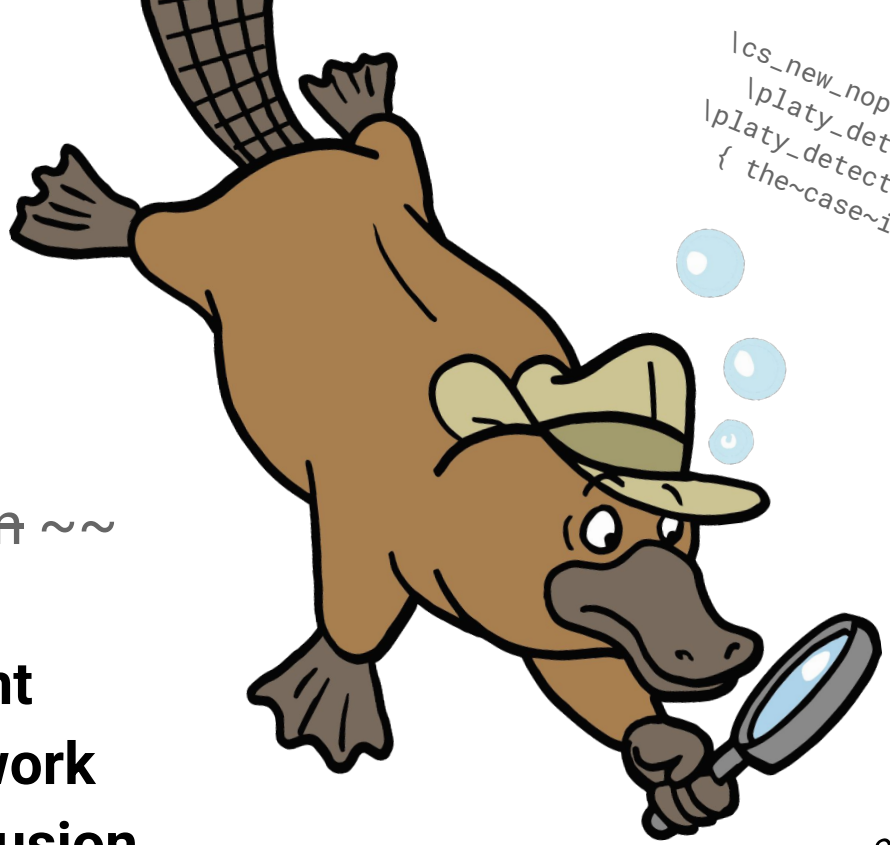
function __acro_absolute_page:w is defined in these files:

1. /usr/local/texlive/2025/texmf-dist/tex/latex/acro/acro.sty

function index

file index

contents



```
\cs_new_nopar:Nn
  \platy_detective_report:n { #1 }
\platy_detective_report:n
{ the~case~is~closed. \par }
```

```
\cs_generate_variant:Nn
  \platy_mission_done:
  { no }
```

5 experiment

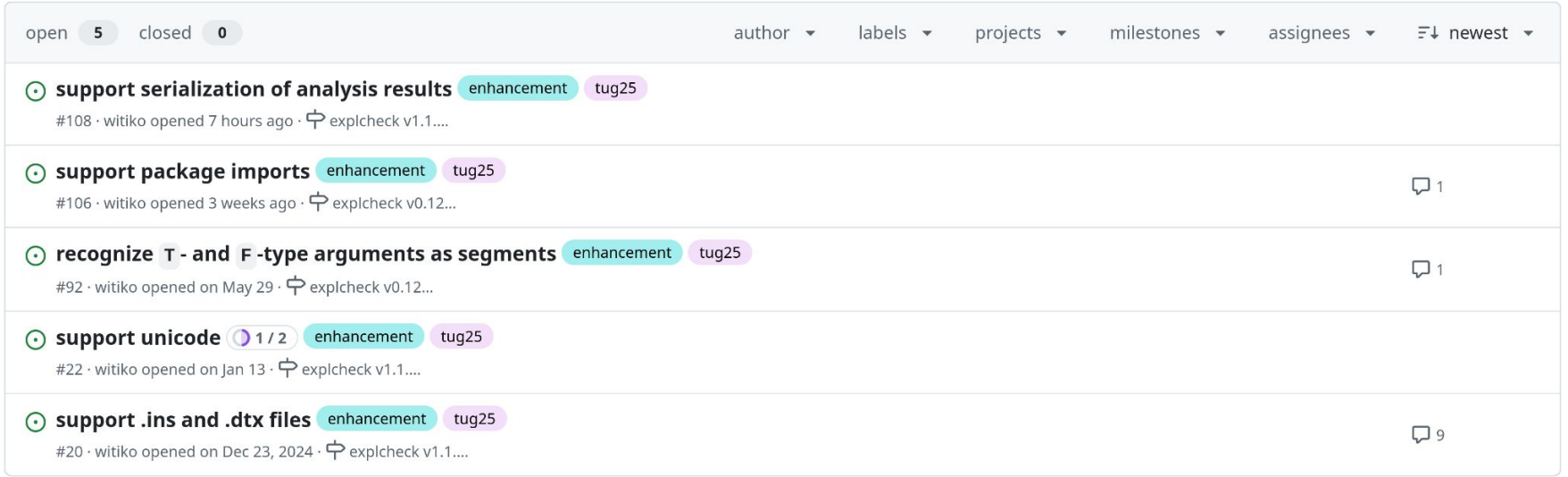
```
$ docker run --rm -it texlive/texlive@sha256:9217582309e77b37da602fdfb2bcefcbd067c30b209cff5efc86a76371a43c65
# find /usr/local/texlive/ -type f -iregex '.*\.\(tex\|cls\|opt\|sty\)$' > files.txt
# explcheck --verbose -- $(cat files.txt) | tr '[:upper:]' '[:lower:]' | sed -n '/aggregate statistics:/,$p'
```

aggregate statistics:

- 210,299,484 total bytes in 11,203 files
- 25,719,897 expl3 bytes (12% of total bytes) containing 7,177,900 tokens and 573,209 groupings
- 54,257 top-level other statements spanning 4,020,983 tokens (56% of tokens, ~7% of total bytes)
- 23,758 top-level function definitions spanning 1,436,307 tokens (20% of tokens, ~2% of total bytes)
    + 40,448 nested other statements spanning 1,073,600 tokens, some in nested function definitions
    + 11,514 nested blocks of other complex tokens spanning 168,743 tokens, some in nested function definitions
    + 2,525 nested function definitions with a maximum nesting depth of two, spanning 31,098 tokens
    + 892 nested blocks of other simple tokens spanning 7,565 tokens, some in the nested function definitions
    + four nested function variant definitions spanning 26 tokens
- 16,775 top-level blocks of other complex tokens spanning 741,463 tokens (10% of tokens, ~1% of total bytes)
- 5,166 top-level function variant definitions spanning 28,018 tokens (<1% of tokens, ~<1% of total bytes)
- 261 top-level blocks of other simple tokens spanning 3,440 tokens (<1% of tokens, ~<1% of total bytes)

6 future work

finished 3 out of 5 (60%) processing steps and 29 out of 75 (~40%) warnings and errors[10]



10 mirrors.ctan.org/support/expltools/doc/warnings-and-errors.pdf 11 github.com/witiko/expltools/issues?q=label:tug25

7 conclusion

correctness of tex programs is difficult to check
expl3 is rising in popularity, programs are much easier to analyze than with tex

explcheck is an expl3 linter, developed as part of the tex development fund
expltools bundles explcheck and (soon) other tools for expl3 developers

current version implements ~50% of the planned functionality
v1.0.0 eta: sometime between november 2025 and september 2026

thank you for
your attention