TORONTO PERL MONGERS • APRIL 30, 2026

Parsing Perl Without Perl

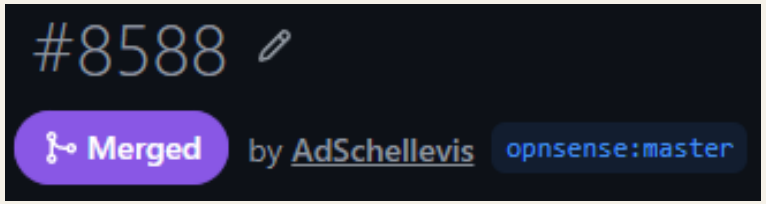
A Rust LSP for Perl 5

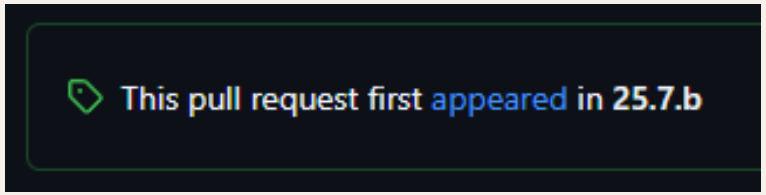
Built most of a compiler to stay out of the runtime.

Steven Zimmerman, CPA

@EffortlessSteven

unbound: fix invalid FQDN generation from DHCPv4 static map domains





WHY NOT JUST RUN PERL?

Modern LSPs need analysis at typing speed.

Perl's native answer is execution.

perl-lsp builds the front end. No runtime.

Perl parses Perl by running Perl.

Every other approach compromises somewhere.

TMTOWTDI

perl

```perl
$result = $x / $y / $z;       # division

@parts  = split /foo/, $bar;  # regex

$text   =~ s/\$(\w+)/$vars{$1}/ge;
```

Same character. Different token. **Parser state decides.**

BRACES ARE NOT ENOUGH

perl

```perl
my $href = { a => 1, b => 2 };

sort { $a <=> $b } @list;

if (1) { map { $_ * 2 } @arr }
```

Same braces. Different construct. **Context decides.**

HEREDOCS DEFER WORK

perl

```perl
print <<END or die;
Hello
END
```

The lexer sees the promise now.
The body arrives later.
The statement keeps going.

Tree-sitter failed.
PEG failed.

Not because they’re bad tools.

Because Perl needs parser-stateful lexing.

PEST LASTED ONE WEEK

tree-sitter-perl kept breaking my AST tooling.

I tried the obvious path: **Pest**.

A week later, easy Perl worked.

That was the trap.

GRAMMAR FIGHTS THE LANGUAGE

Backtracking exploded.

Parser-stateful lexing had nowhere clean to live.

Every workaround made it uglier and slower.

Perl is a state machine wearing a grammar's clothes.

THE PARSER CONTROLS THE LEXER

Parser expectation drives lexer mode.

ExpectTerm
ExpectOperator

Recursive descent parser.

STATE TRANSITION

```
Parser expects term
→ /…/ can be regex
Parser expects operator
→ / is division
```

LEXER AND PARSER COOPERATE

Input	Parser expects	Lexer returns
`$x`	term	**scalar**
`/`	operator	**division**
`split`	term	**builtin**
`/foo/`	pattern	**regex**

Same character. Different token. **The state machine decides.**

SUBLEXING STACK

```perl
my $s = "foo $bar->{key} baz";
```

```
outer string
    interpolated expression
        $bar->{key}
outer string
```

Push context. Tokenize inner stream.
Pop context. Resume.

SAME STACK UNDER LOAD

```perl
$text =~ s/\$(\w+)/$vars{$1}/ge;
```

Part	Context
`s///`	substitution
`\$(\w+)`	regex body
`$vars{$1}`	replacement expression
`/e`	**replacement is Perl code**

One mechanism. Several Perl contexts. **No execution.**

PERL-PARSER

Not just a parser.

A compiler front end.

600k lines of Rust • one person • one year.

Lexer	Symbol tables
Parser	Semantic analysis
AST	MRO-aware lookup
Scope resolution	BEGIN symbol-table modeling

Everything Perl does at parse time.

Nothing it does at runtime.

WHAT IT HANDLES NOW

Corpus	Clean parses
CPAN top 1000	8,931 / 9,372
Ubuntu system Perl	6,890 / 7,095
Project corpus	91 / 91

File-level clean parse: no parser-recorded errors.
Not full semantic fidelity.

THE EXECUTION BOUNDARY

What requires running Perl:

computed string `eval`
runtime glob aliasing
`AUTOLOAD` from runtime state
pathological source filters

The line is execution, not analysis.

SENIOR ATTENTION IS THE COST CEILING

Per verified PR:

LLM verification	$2.00	40 × $0.05
CI	$5.00	
Senior review	**$16.67**	5 min × $200/hr
Total	**$23.67**	

70% is the bottom row.
Spend more on the top two to shrink it.

At 1,000 PRs/day, every senior minute saved = $100k/month.

WHAT USERS GET

Single native binary.

VS Code • Open VSX • any LSP client.

definition · rename · completions
diagnostics · references · debug adapter

Everything Perl does at parse time.
Nothing it does at runtime.

Steven Zimmerman, CPA
@EffortlessSteven