

autoscale: true

6 July 2017

- ▶ Experimental support for MSVC toolchain
- ▶ Debugging not supported yet

<https://www.surveymonkey.co.uk/r/TLLYQJW>

- ▶ [Blog post](#)
- ▶ [Packt](#)
- ▶ [Amazon](#)

MinGW by STL

by Stephan T. Lavavej of Microsoft's STL fame

"Steh-fin Lah-wah-wade"

Nuwen.org



- ▶ RHEL7 and earlier: GCC uses old C++98 copy-on-write strings (no SSO)
- ▶ even in C++11 and C++14 mode (non-compliant behaviour)
- ▶ RHEL8 will have C++11 SSO strings

Release Notes

- ▶ bugfix release – fixes regressions in GCC 6.3

Generics in Go?

Twitter

The screenshot shows a Twitter thread. At the top is a tweet from 'Uncaffeinated' (medium.com) titled 'Parallelizing Enjarify in Go and Rust', submitted 3 months ago. It has 138 retweets and 79 comments. Below this, a yellow box states 'you are viewing a single comment's thread. view the rest of the comments →'. The thread continues with a comment from 'pftbest' (7 points, 3 months ago) asking for an explanation of Go syntax, showing a code snippet: `type ImmutableTreeList<ElementT> struct {`. This is followed by a reply from 'Uncaffeinated' (12 points, 3 months ago) stating that Go doesn't have generics and that the code is a 'template' file where angle brackets are replaced with Canadian Aboriginal Syllabics characters. The reply also includes a link to a post about 'The Canadian Aboriginal Syllabics block'.

Parallelizing Enjarify in Go and Rust (medium.com)
submitted 3 months ago by Uncaffeinated
138 79 comments share save hide give gold report

sorted by: best ▼

you are viewing a single comment's thread.
view the rest of the comments →

[–] pftbest 7 points 3 months ago

can you please explain this go syntax to me?

```
type ImmutableTreeList<ElementT> struct {
```

I thought go doesn't have generics.

permalink embed save report give gold reply

[–] Uncaffeinated 12 points 3 months ago

It doesn't. That's just a "template" file, which I use search and replace in order to generate the three monomorphized go files.

If you look closely, those aren't angle brackets, they're characters from the Canadian Aboriginal Syllabics block, which are allowed in Go identifiers. From Go's perspective, that's just one long identifier.

Figure 2: Inline

C++London: The Badlands of C++ (game development)

Valentin Galea

- ▶ MSVC is prevalent
- ▶ Huge amounts of legacy code using C++98
- ▶ Mostly proprietary code, sometimes really ugly
- ▶ Process- and deadline-driven, no clean-up exercises (“ship it!”)
- ▶ Warnings are just fine (ignored)
- ▶ Low-latency specifics
 - ▶ no exceptions
 - ▶ minimize allocations
 - ▶ special container versions (EASTL)
- ▶ garbage collection for dynamic objects (Lua)
- ▶ [Splash Damage Tech Blog](#)

Jackie Kay

- ▶ a shorter version of the talk at C++Now 2017
- ▶ serialization
- ▶ program options
- ▶ addition: metaclasses

Declaration style

Style 1

```
1 | const char *foo = "Foo"; // * and & bind to the right
```

Style 2

```
1 | const char* foo = "Foo"; // the type is 'pointer to const char'
```

Style 3

```
1 | char const *foo = "Foo"; // declaration reads from right to left
```

MS C++ 7 on eBay

\$300

- ▶ original box, still sealed!
- ▶ MFC, no templates
- ▶ Windows 3.1 SDK
- ▶ overlays for extended memory/swapping
- ▶ OLE, DDE
- ▶ CodeView debugger
- ▶ over 5000 pages of manuals!



Abhisekh Gupta teaches C++

- ▶ uses Turbo C++ within DosBox within Windows 10
- ▶ why?
- ▶ what (if anything) should we tell him?