Protocol Compiler - Feature #21785

Add Rust as a supported language

01/29/2019 11:28 AM - Richard Neswold

Status:	Feedback	**Start date:**	01/29/2019
Priority:	Normal	**Due date:**	
Assignee:	Richard Neswold	**% Done:**	0%
Category:	Rust generator	**Estimated time:**	0.00 hour
Target version:	Protocol Compiler v1.2		

Description

There is some interest to have Rust be a supported target language of the protocol compiler. This issue will track the efforts adding it.

History

#1 - 01/29/2019 11:30 AM - Richard Neswold

I've added some notes about mapping to Rust: Rust Generator.

#2 - 02/11/2019 11:06 AM - Richard Neswold

- Target version set to Protocol Compiler v1.2

- Status changed from New to Closed

Added: 66acdb6a

This is a preliminary implementation; I fully expect further changes to the API as Rust users find deficiencies in the code. I encourage Rust programmers to look at the generated code and recommend changes to the API which improve correctness and to offer optimization suggestions.

#3 - 02/18/2019 10:43 AM - Richard Neswold

- Status changed from Closed to Feedback

Re-opening for feedback.

After some more reading and research into Rust, it seems the crate that provides the official, asynchronous API is tokio. If/when we develop an ACNET crate, it'll probably be based on this API. The tokio crate has lots of useful, asynchronous constructs including traits for encoding and decoding protocols! I think it may be a good idea to retool the protocol compiler's Rust support to use these traits. My only hesitation is doing this would make Rust the first language dependent on a third-party library[1]. Installing third-party crates is insanely easy (at least on my Mac), so it doesn't seem like this is too much of a hardship. Plus, there's talk the tokio implementation might become the async/await core for Rust which means, in the future[2], we won't be using a third-party crate.

Any thoughts, Brian?

---

[1] We do have a GWT generator, but it's a specialization of the Java generator. The generic Java generator doesn't have any third-party dependencies.

[2] But I don't wait to await on that Future. Ha! A Rust joke!

#4 - 04/01/2019 03:08 PM - Richard Neswold

- Tracker changed from Support to Feature