Ruby master - Misc #16172

C++ situation?

09/19/2019 01:51 AM - shyouhei (Shyouhei Urabe)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td></td>
</tr>
</tbody>
</table>

**Description**

(This issue is MRI specific)

Few weeks ago I pushed a minimalist C++ compatibility layer to the master branch. Then Nobu pushed his MakeMakefile::CXX implementation. It seems our C++ support got suddenly improved. What is our current stance? Are the commits just episodic events that was never meant to change something? Or is it okay to say we think people can write their extension libraries in C++? Do we plan to have some CI to test C++ extensions then?

**History**

1 - 09/20/2019 02:25 AM - shyouhei (Shyouhei Urabe)

- Status changed from Open to Closed

We discussed this topic at the developer meeting held yesterday.

The attendees' mood was that we would like to say using C++ for extension libraries is just okay. However because C and C++ have diverged for ages, it is not trivial to write a header file that is valid for both C and C++. We agreed to improve tests around this area; C++-written extension libraries would be added to our test suite.

2 - 09/20/2019 07:27 AM - shevegen (Robert A. Heiler)

I wanted to briefly add that I think C++ support or have it as a "first class citizen" could be great - even though competent programmers such as matz or linus are not necessarily the biggest fan of C++. :-)

I like some parts of C++, simple things such as cout << "hello world" (almost like in ruby, at the least the <<), and there have been other ruby users who wrote wrapper code for C++ before - hammac here, for example, with wxwidgets bindings, but also the old kde/qt bindings for ruby, which unfortunately have not been that active ever since the kde4 to kde5 transition (I can not say why for certain; I don't think it is directly related to ruby per se, but to KDE moving quite rapidly and people getting frustrated or just lacking time to add code when a target moves.)

Anyway - I think there may always be interested folks for C++ stuff, so for these users/developers, I think C++ support as a "first-class citizen" would be great.

This reasoning is a bit similar to e. g. integrating bundler and gemifying the stdlib; the net benefit to individual ruby users, but also to the whole ecosystem, may easily warrant the transition (and of course the effort, time etc... spent to do so, too). Anyway that is just my 2 cc. :)