C++0x implementation guesstimates?

Asked 15 years, 11 months ago   Modified 14 years, 9 months ago

Viewed 1k times

5

The C++0x standard is on its way to being complete. Until now, I've dabbled in C++, but avoided learning it thoroughly because it seems like it's missing a lot of modern features that I've been spoiled by in other languages. However, I'd be very interested in C++0x, which addresses a lot of my complaints. Any guesstimates, after the standard is ratified, as to how long it will take for major compiler vendors to provide reasonably complete, production-quality implementations? Will it happen soon enough to reverse the decline in C++'s popularity, or is it too little, too late? Do you believe that C++0x will become "the C++" within a few years, or do you believe that most people will stick to the earlier standard in practice and C++0x will be somewhat of a bastard stepchild, kind of like C99?

`c++`   `programming-languages`   `language-design`   `legacy`

Share

Improve this question

Follow

asked Jan 24, 2009 at 6:20

dsimcha
68.6k ● 55 ● 219 ● 340

5 Answers

16

I see no reason why C++0x shouldn't be adopted. The C++ community is much more forward-looking than C. C was always meant to be a "portable assembler language", so people who use that aren't really super interested in fancy new features. C++ spans much wider, and I've yet to hear of a C++ programmer who *wasn't* looking forward to 0x. (It's also my impression that the C++ community is much "stricter", and really don't want to move outside the standard into undefined behavior, which implies you choose either C++03 or C++0X rather than a half-implemented hybrid. C programmers tend to be much more relaxed about that, and seem happy to use C89 with just a couple of C99 features and headers mixed in)

However, it'll take a few years before Microsoft catches up, at least. Visual Studio 2010 will support a small handful of C++0x features (lambdas, decltype and a couple of others), but the vast majority will *not* be supported. We'll have to wait for VS2012 or whatever the next version ends up being, to have somewhat complete support.
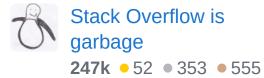
With GCC/G++, the situation is a lot better, since most of the standard has been implemented there already (the standard committee doesn't like adopting features that haven't been implemented and tested in a real compiler, and a GCC fork is often used for that)

But it'll probably still take some time to get that stable and production-ready.

About C++'s "decline in popularity", I don't really see it. I don't think C++ has declined significantly in popularity for the last years. RAD developers have already jumped ship, of course, to .NET, Python or other languages or platforms. But where C++ is used today, there aren't many viable alternatives, and no reason why it should decline in popularity.

Share Improve this answer

Follow

edited Jan 24, 2009 at 7:38

answered Jan 24, 2009 at 7:32

Stack Overflow is garbage

247k ● 52 ● 353 ● 555

▲

3

▼

I don't know about other vendors, but from what I've seen, Microsoft plans to include four C++0x language features in Visual C++ 2010:

1. rvalue references

2. auto

3. lambdas

4. static assert

Although this is a small set of C++0x features, they are important ones. Some will enable programmers to write much more compact (auto, lambdas) and error free code. Some (like rvalue references) enable libraries to be more efficient. Microsoft likes lambdas as an [enabler for parallel computing](#).
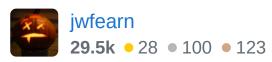
IMHO: `auto` alone will make it so much easier to use templates that more programmers will do so. And hopefully this will increase demand for more C++0x features from Microsoft and all vendors.

Microsoft will also be updating their C++ Standard Library implementation but I don't know the details. I believe they are modifying some container classes to take advantage of rvalue reference move semantics. And I believe they including more of TR1 .

Share Improve this answer   edited Mar 28, 2010 at 19:38

Follow

answered Jan 24, 2009 at 16:11

jwfearn
29.5k ● 28 ● 100 ● 123

---

1   You'd expect them to also include the TR1 stuff they ship today. – MSalters Jan 26, 2009 at 8:48

Of course that's true, my post is about language features. I will edit to make this more clear. – jwfearn Jan 29, 2009 at 20:17

Your edit is correct: the TR1 library additions have been added to the std namespace and the containers have been updated to support move semantics. – James McNellis Mar 28, 2010 at 19:45

2

Some implementations are already on their way to C++0x: (gcc). My intuition says that C++0x support will be available in major compilers fairly quickly; however a large body of legacy code still exists that must be maintained.

Share Improve this answer

Follow

answered Jan 24, 2009 at 6:33

thekidder
3,574 ● 1 ● 34 ● 37

2

Newer versions of gcc already support many of the C++0x features: http://gcc.gnu.org/projects/cxx0x.html

Share Improve this answer

Follow

answered Jan 24, 2009 at 6:35

grepsedawk
6,059 ● 5 ● 26 ● 22

0

Microsoft will be including C++0x support in Visual Studio 2010 later this year (a community technology preview is already available).
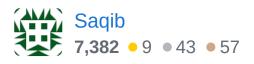
I don't think it'll become "the C++" any time soon, but rather certain people will choose to add parts of the new syntax where it makes sense in their code.

I don't do much C++ these days, but people I talk to either love it, or feel that the beauty of C++ is in the control they get, and that if they wanted a language with all those extra features they'd use C#/Java.

Share Improve this answer

Follow

answered Jan 24, 2009 at 6:38

Saqib
7,382 ● 9 ● 43 ● 57

No, Microsoft will be including a couple of 0x features. At a rough estimate, we're talking about 10% of the new c++0x stuff. – Stack Overflow is garbage Jan 24, 2009 at 7:26

... sure it may be only 10% of the feature list jalf, but it's ~70% of the "value" for the average C++ programmer. – j_random_hacker Jan 25, 2009 at 11:45