

---

Subject: Re: C++ FQA

Posted by [cbpporter](#) on Wed, 07 Nov 2007 10:33:33 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

I've read a good deal of the FQA (and the FAQ too) and while it is mostly right, it is also strongly biased. It still makes a point though, and I think a lot of people agree that C++ is too complicated for it's and everybody else's sake. I would love for an alternative, something that is more simple, straight-forward, but still powerful.

Right now D could be the only alternative (C++0x is even more complicated and redundant). I used D for 2500+ line project (which in C++ would have been about 4000-5000 lines) with about 20 files and I think I have a pretty good idea of what it can do. It has some strong points:

1. True module support.

No more including hundreds of kilobytes if not megabytes of header files in each compilation unit. No more writing every declaration twice. Modules act like Java packages, can be fully qualified to avoid name clashes and can be combined to create a library with different access levels. And because each module is compiled only once, D is lightning fast. My entire project compiled from scratch almost instantly, and if I only modified the content of a function or other minor detail, truly instantly.

2. A lot of built in features, like variable length array and hash maps, which combined with some extra functions, can be used to create types like Vector or Map, but even more easy to use and without templates. `char[]` and about 10 extra functions could make String and StringBuffer obsolete.

3. Templates + mixins + static ifs are stronger than templates in C++ + preprocessor. I never used them, because you can do in D a lot more without templates than in C++.

4. Optional (but defaulted to true) garbage collection.

As much as you could dislike the idea of garbage collection, memory management in C++ is a nightmare, and from this point of view, even U++ which has a lot less such issues, is not able to give such pain free management.

But it also has some disadvantages. Mostly because it has been branched to D 1.0, which is stable, and 2.0 which is a moving target, is not compatible with 1.0, and is a little more complex. It introduces const correctness as in C++, but D can live a lot easier without such extra access methods to have to deal with. Also, the standard library is widely disliked, and a third party library has been created, which is pretty good (even though a little to C++ style), but cross linking between the libraries is impossible. Also operator overloading is just a little bit less powerful.

Still, it is quite a good programming language, having a lot of pain free features and performance comparable to C/C++. I would love to see such a great GUI library as U++ for D, and I'm sure it will continue to evolve.

---