Subject: Is U++ Core open for suggestions?
Posted by royalstream on Wed, 08 Feb 2006 20:08:46 GMT
View Forum Message <> Reply to Message

I have a question regarding the stage of development of U++.
Is it possible to make suggestions regarding core functionality such as the pick semantics?
The reason I ask is because U++ might be past that point.
I like the Pick Semantics but I think there is room for improvement in the way they are implemented. For example, in several situations is not clear wether deep copy or picking is taking place (unless you know the way the specific type works ).
I have several ideas (nothing on paper yet ) but I want to know beforehand if there's any chance for them to be taken seriously.

Thanks!

Steven

Subject: Re: Is U++ Core open for suggestions?
Posted by mirek on Wed, 08 Feb 2006 20:45:12 GMT
View Forum Message <> Reply to Message

Hi,

well, with 20MB of commercial codebase, doing huge changes to "pick" is hard

However, we are always open to discussion.

I agree that cleaner syntax sugar there would be nice. However, the trouble is that as long as pick semantics is supposed to be used for function return values, there does not seem to be any other option (we need to use regular copy constructors there to pass return value to target).

Even worse, we like compiler to generate default copy constructors / assignment operators (doing that manually is way too much trouble).

So in the end it goes down to "you need to know type's transfer semantics".

I was trying to find alternatives (especially I do not like that const cast ugliness) for years, but was unable to come up with anything better.

Mirek

Subject: Re: Is U++ Core open for suggestions?

Posted by royalstream on Thu, 09 Feb 2006 20:59:01 GMT
View Forum Message <> Reply to Message

I had several ideas, all of them more or less revolved around auxiliary types.
But just as you said, I always found things I didn't like about each one of them.
I came to think that even though there may be other ways of doing it; they all have shortcomings that defeat the whole purpose.
However, I am new to the project and it makes a very big difference to hear about a similar experience from you.

Thanks a lot for your prompt answer.

Steven

Subject: Re: Is U++ Core open for suggestions?
Posted by mirek on Thu, 09 Feb 2006 21:12:40 GMT
View Forum Message <> Reply to Message

BTW, there were some similar movements in C++ community, try to google for "MOJO Alexandrescu" or "Hinnant r-value reference".

First one is done using current C++, but has the exactly the sortcomings we both have encountered.

Second one is C++0x proposal, unfortunately it missed the composition issue... so very likely no help from there either

In the end, the best help comes from MSC compiler bug that actually allows binding temporaries to non-const references, so it is possible to define pick_ empty for MSC and detect at least some cases.

All of that said, maintaining pick semantics is much easier than it might seem. Yes, from time to time I do encounter "pick assert failed" (runtime enforcment of pick semantics in debug mode), but generally, effort is quite simple to get things work...

Mirek