Subject: _pick understanding
Posted by kohait00 on Tue, 27 Apr 2010 19:08:52 GMT
View Forum Message <> Reply to Message

hi there,

in
http://www.ultimatepp.org/srcdoc$Core$pick_$en-us.html

there is _pick explained as

#define _pick const

but when i look in code the _pick definition in MSC environment is empty.. why? setting it to const there as well compiles well..
is there any thing special to MSC to consider? just out of curiosity..

Defs.h:277

```
#ifdef COMPILER_MSC
#define pick_
#else
#define pick_ const
#endif
```

another question:

Topt.h:252

```
template <class T, class B = EmptyClass>
class DeepCopyOption : public B {
public:
 friend T& operator<<=(T& dest, const T& src)
 { if(&dest != &src) { (&dest)->T::~T(); ::new(&dest) T(src, 1); } return dest; }
 friend void DeepCopyConstruct(void *dest, const T& src)
 { ::new (dest) T(src, 0); }
 friend T *DeepCopyNew(const T& src)
 { return ::new T(src, 0); }
};
```

uses 1 for param in <<= operator while link above also states that the second int parameter is just for distinciton and ignored. is it evaluated anywhere? maybe to indicate a reuse??

IMHO _pick should be explained a bit better, i understand the problem leading to _pick solution, but especially the explanation

Quote:
C++ disallows binding temporaries to non-const references - and that is unfortunately just the thing we need to do here, as we need to change the source temporary returned from a function.

should be visualised by a (not permitted) code snippet as well.

---

## Subject: Re: _pick understanding
Posted by cbpporter on Wed, 28 Apr 2010 09:27:30 GMT
View Forum Message <> Reply to Message

I believe that the different defines are because in default mode, MSC is not 100% compliant with C++ standard and is a little more allowing than it should be. Basically, it allows you to not use const in some cases where const should be used. This dates back from the golden era of MFC AFAIK.

Please correct me if I'm wrong.

I would like to see const added to pick for MSC too on a pure ideological level, but I have no practical consideration that would make me change that.

---

## Subject: Re: _pick understanding
Posted by kohait00 on Wed, 28 Apr 2010 18:38:15 GMT
View Forum Message <> Reply to Message

as far as trying, it works  so far...

any ideas on the int value in special copy constructor, why sometimes 0 sometimes 1..?

---

## Subject: Re: _pick understanding
Posted by mirek on Thu, 29 Apr 2010 21:38:29 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Wed, 28 April 2010 05:27I believe that the different defines are because in default mode, MSC is not 100% compliant with C++ standard and is a little more allowing than it should be. Basically, it allows you to not use const in some cases where const should be used. This dates back from the golden era of MFC AFAIK.

Please correct me if I'm wrong.

You are right.

Quote:
I would like to see const added to pick for MSC too on a pure ideological level, but I have no

practical consideration that would make me change that.

There is a practical reason why we "exploit" this MSC bug:

Doing so, we are able to catch some pick semantics bugs at compile time. E.g.

```
void Foo(const Vector<int>& b) {
  Vector<int> a;
  a = b;
}
```

does not compile with MSC (which is correct).

(Frankly, I believe that M$ got it right here and standard has it wrong

Note: Thankfully M$ seems to use its own semantics in MFC in some places, therefore they perhaps have to keep this non-compliance to standard.

Mirek

---

Subject: Re: _pick understanding
Posted by mirek on Thu, 29 Apr 2010 21:40:08 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Wed, 28 April 2010 14:38as far as trying, it works  so far...

any ideas on the int value in special copy constructor, why sometimes 0 sometimes 1..?

The parameter is there only to distinguish the copy constructor. The value is ignored.

(Frankly, if I would do this again, I would probably used some sort of enum or other special type. But that would be only a minor difference).

Mirek

---

Subject: Re: _pick understanding
Posted by kohait00 on Fri, 30 Apr 2010 07:17:53 GMT
View Forum Message <> Reply to Message

thanks all, that claerified it a bit