Subject: Defs.h minmax() change/fix
Posted by kohait00 on Wed, 23 Dec 2009 18:31:19 GMT
View Forum Message <> Reply to Message

hi there

in Defs.h:214

why not using

```
template <class T>
inline const T& minmax(const T& x, const T& _min, const T& _max) { return min(max(x, _min), _max); }
```

instead of

```
template <class T>
inline T minmax(T x, T _min, T _max) { return min(max(x, _min), _max); }
```

which is sort of logically semantics
especially when you consider the other related templates:

```
template <class T> inline const T& min(const T& a, const T& b) { return a < b ? a : b; }
template <class T> inline const T& max(const T& a, const T& b) { return a > b ? a : b; }
```

Subject: Re: Defs.h minmax() change/fix
Posted by mirek on Thu, 24 Dec 2009 09:58:10 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Wed, 23 December 2009 13:31hi there

in Defs.h:214

why not using

```
template <class T>
inline const T& minmax(const T& x, const T& _min, const T& _max) { return min(max(x, _min), _max); }
```

instead of

```
template <class T>
inline T minmax(T x, T _min, T _max) { return min(max(x, _min), _max); }
```

which is sort of logically semantics
especially when you consider the other related templates:

```
template <class T> inline const T& min(const T& a, const T& b) { return a < b ? a : b; }
template <class T> inline const T& max(const T& a, const T& b) { return a > b ? a : b; }
```

What if min / max for T is defined returning temporary?

Foo min(const Foo& a, const Foo& b);

then you would be returning reference to temporary....

Well, maybe this is not really strong argument as min/max are rarely defined directly - OTOH compilers are really goot at optimizing, so practical differences in code generated are, I believe, unlikely.

That said, proposed change would be nice if type does not have copy.

I am sort of undecided, but current minmax worked fine for 10 years, I guess there is not a strong incentive to change now.

Mirek

---

Subject: Re: Defs.h minmax() change/fix
Posted by kohait00 on Wed, 17 Feb 2010 09:24:51 GMT
View Forum Message <> Reply to Message

yiy are right, the compiler warns
warning C4172: returning address of local variable or temporary

using a

```
s = min(max(a, b), c);

//instead of

s = minmax(a, b, c);
```

does not produce this warning. i still dont get why.

maybe to have a

#define minmax(x, _min, max) min(max(x, _min), _max)

is a choice.

or at least

template <class T>
inline T minmax(const T& x, const T& _min, const T& _max) { return min(max(x, _min), _max); }

reducing the need of object copy by 60%  (roughly)

dont know it this is that much of speed saving, but using refs instead of temp copies is better anyway, isnt it?

---

Subject: Re: Defs.h minmax() change/fix
Posted by mirek on Fri, 19 Feb 2010 15:52:39 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Wed, 17 February 2010 04:24yiy are right, the compiler warns
 warning C4172: returning address of local variable or temporary

using a

s = min(max(a, b), c);

//instead of

s = minmax(a, b, c);

does not produce this warning. i still dont get why.

maybe to have a

#define minmax(x, _min, max) min(max(x, _min), _max)

is a choice.

or at least

template <class T>

inline T minmax(const T& x, const T& _min, const T& _max) { return min(max(x, _min), _max); }

reducing the need of object copy by 60%  (roughly)

dont know it this is that much of speed saving, but using refs instead of temp copies is better anyway, isnt it?

In this case, I believe there will be no difference in code generated in most cases. It is inline after all. And in fact, it is used on fundamental types 90% of time.

Mirek

---