Subject: Re: Strange behavior of Point in watches
Posted by mirek on Wed, 02 Sep 2009 15:59:14 GMT
View Forum Message <> Reply to Message

dolik.rce wrote on Tue, 01 September 2009 22:47Helo Mirek!

Sorry it took me so long, but I was busy at work.

I did a little investigation and finally found exact piece of code that causes problems in gdb. I'm not sure what is the real reason, but all the troubles are caused by following constructors: Point_(const Point_<int>& pt) : x((T)pt.x), y((T)pt.y) {}
Point_(const Point_<short>& pt) : x((T)pt.x), y((T)pt.y) {}
Point_(const Point_<double>& pt) : x((T)pt.x), y((T)pt.y) {}
Point_(const Point_<int64>& pt) : x((T)pt.x), y((T)pt.y) {}
If you replace those four lines by template <class U>
Point_(const Point_<U>& pt) : x((T)pt.x), y((T)pt.y) {}
watches show correct values. Do you think that this workaround is safe? To my best knowledge it should produce absolutely same results as original code and nobody should try passing there any non-scalar type, so it should not cause any harm... but I'm no expert when it comes to templates

Also, during the investigation I found that same problems apply to Size_ and Rect_. Since their anatomy is very similar, it should be possible to apply the same workaround.

Honza

I do not know... I really hate such workarounds for debugger problems...

Mirek