

Sunday, December 13, 2009

### C++ Delayed Constructor

Messing recently with a multiple--dispatch solution in C++ / Boost / Loki environment I faced a number of challenges one of which was to implement a "delayed constructor" mechanism. Basically, this is a somehow known problem of binding values to constructors which isn't solvable by `boost::bind` as there is no way to get an address of the constructor, but for which a solution exists in `boost::lambda::bind`. Honestly, I didn't look into there in deep. Instead I went for a short [maybe elegant] solution, compilable by any compiler supporting variadic templates (spirit of c++0x is already within our codes):

```
template
class DelayedConstructor
{
public:
    template
    DelayedConstructor(Args ...args)
    {
        class ConstructHelper
        {
        public:
            static TObject* DoConstruct(Args ...args)
            {
                return new TObject(args...);
            }
        };
        m_funHelper = boost::bind(ConstructHelper::DoConstruct, args...);
    }

    TObject* operator>()()
    {
        return m_funHelper();
    }

protected:
    boost::function m_funHelper;
};
```

And here we use it:

```
class A
{
public:
    A(int a, std::string str) {}
};

//Binding values to the constructor
DelayedConstructor dc(21, "Good stuff!");

//Actually creating an object
A *pA = dc();
```

To me it looks very simple and rational. And keep in mind that it would work with any number of constructor arguments, which you don't have to worry about at all. Just Use it.

But what's left for a homework? Okay, let's teach it to accept binding of only specific args. As we are used to do with

regular functions.

P.S. But what is even more exciting is that as soon as c++0x lambdas come to play the solution of this problem would become trivial by using lambda factory function for desired class, which will eliminate usage of `boot::bind`.

Posted by Dmitry Shandyba at 05:25

Monday, January 7, 2008

### Same DLL loaded twice

If you ever felt interest into how Windows loader works you definitely know that it works just so same DLL can't be loaded twice, under any circumstances. OK, I can put a correction here, a DLL referred by same path. Though a colleague of mine, James Thomson, was lucky to prove the opposite. The question has already been risen in 4 different places in the Internet: MSDN Forums, Google Groups, WASM Phorum (in Russian), RSDN Forums (in Russian). Still I would like to point out attention to it here as well, as in none of that places we were able to get the explanation of what's going on with Windows loader. If you like you can have a look at sample application prepared by James here. In few words we first load a DLL by creating a COM object living in it using SxS (Side by Side) COM manifest for that purpose. And then we load it again, as statically linked DLL to an intermediate DLL that we load via LoadLibrary. Load order is explained here

Well, voila, you can already see 2 instances of the same binary image in memory. Just if you are curious and try changing steps of load procedure you'll be surprised to see that doing so causes only one DLL instance to reside in memory. That's all. Reproducible under WinXP SP2 and under Vista Business. Have not tried under any other OS though.

Posted by Dmitry Shandyba in Main at 18:06

Tuesday, July 17, 2007

**"What the heck?!" thought Bill trying to find a "Start" button on his iPhone**

Don't really know why everybody talks about iPhone so much?! :)

Posted by Dmitry Shandyba in Main at 22:26

Saturday, July 14, 2007

### **Google Analytics does not reveal IP addresses of site visitors**

Recently while playing with wonderful Google Analytics service I've been greatly dissapointed to know that thereÂ was no way to obtain statistical information about visitors IP addresses! This sounds like nonsense having in mind an awesome number of different other pieces of information about visitors capable of being analyzed. Really, the prehistory was also quite interesting, just look here:

[http://www.roirevolution.com/blog/2006/09/view\\_visitor\\_ip\\_address\\_in\\_google\\_analytics.html](http://www.roirevolution.com/blog/2006/09/view_visitor_ip_address_in_google_analytics.html). The found backdoor to workaround the limitation was closed quite fast :) If you haven't read all the comments there, here is what I wrote:"It seems to be a strange position of Google. Information about IP addresses hardly breaks any privacy rules as there is no way to access a web site and not to reveal the IP address. So any visitor by default agrees for his / her IP to be collected. If higher level of privacy is need, higher level tools should be used by users. That's how it is done, you all know that.This Analytics limitation just makes the product less handy as info about IPs is anyway stored in web-server's log files and can be easily accessed and analyzed by site owner."Â Another interesting thing is that there are people agreeing with Google in this strange Policy. Internet is a big sandbox with little still fundamental rules. One of the main ones is that privacy is really a hardly achievable thing. Just becuase it's not there by design.

Posted by Dmitry Shandyba in Main at 00:30