Subject: [XML] Assertion when GetAttrCount()
Posted by g00fy on Mon, 04 Dec 2006 07:29:56 GMT
View Forum Message <> Reply to Message

```
#include <CtrlLib/CtrlLib.h>
#include <ide/Common/Common.h>
#include <Core/Core.h>


void write_out( FileOut& out, const XmlNode& xml, int ident )
{
 switch( xml.GetType() )
 {
 case XML_TEXT:
  out << String(' ', ident) << "Text: " << xml.GetText() << "\n";
  break;
 case XML_TAG:
  out << String(' ', ident) << "Tag: " << xml.GetTag() << "\n";
  for ( int i = 0, total = xml.GetAttrCount(); i < total; i++ )
  {
   out << String( ' ', ident ) << "+> " << xml.AttrId(i) << ": " << xml.Attr(i) << "\n";
  }
  ident += 2;
  break;
 case XML_DOC:
  out << String(' ', ident) << "Doc:\n";
  break;
 default:
  out << String(' ', ident) << "**UNDEFINED TAG**\n";
  break;
 }

 for ( int i = 0, total = xml.GetCount(); i < total; i++ )
 {
  write_out(out, xml[i], ident + 2);
 }
}

GUI_APP_MAIN
{
 Package p;

 String filename("C:\\Upp\\uppsrc\\ide\\Common\\Common.upp");

 p.Load(filename);

 XmlNode xml;
 XmlNode makefile = xml.Add("makefile");

 XmlNode lib = makefile.Add("lib");
 lib.SetAttr("id", GetFileName(filename));

 FileOut out("c:\\tst.dat");
 write_out(out, xml, 2);
}
```

This code asserts on the call to "xml.GetAttrCount()". It thinks the vector holding those items have -1 items only...

Greetz

Subject: Re: [XML] Assertion when GetAttrCount()
Posted by mirek on Tue, 05 Dec 2006 12:12:03 GMT

View Forum Message <> Reply to Message

g00fy wrote on Mon, 04 December 2006 02:29#include <CtrlLib/CtrlLib.h>
#include <ide/Common/Common.h>
#include <Core/Core.h>

U++ headers are always arranged so that "higher" include "lower", means "#include <ide/Common/Common.h>" is enough here.

Quote:
This code asserts on the call to "xml.GetAttrCount()". It thinks the vector holding those items have -1 items only...

-1 in items is generally the sign of broken pick semantics. Indeed:

```
 XmlNode makefile = xml.Add("makefile");

 XmlNode lib = makefile.Add("lib");
```

Here you pick "makefile" node out of xml and "lib" out of makefile. This is basically "java style code"...

In U++ we are usually referencing objects created inside complex structures:

```
 XmlNode& makefile = xml.Add("makefile");
 XmlNode& lib = makefile.Add("lib");
```

These are basic paradigms of U++ - perhaps unusual for newcomers, but in the end this is a way how to keep things simple and deterministic (and avoid smart pointers and GC wishes...)

Mirek

Subject: Re: [XML] Assertion when GetAttrCount()
Posted by g00fy on Fri, 08 Dec 2006 00:50:38 GMT
View Forum Message <> Reply to Message

I like smartpointers! They keep my memory organised

Subject: Re: [XML] Assertion when GetAttrCount()
Posted by mirek on Fri, 08 Dec 2006 08:18:07 GMT
View Forum Message <> Reply to Message

g00fy wrote on Thu, 07 December 2006 19:50I like smartpointers! They keep my memory organised

Well, you will not like U++ then....

U++ has all resources very organised - but without smart pointers. Core U++ paradigms are:

- GC is bad idea
- Shared reference counted pointers is even worse idea
- Manually releasing resources is absolutely stupid idea

Mirek

Subject: Re: [XML] Assertion when GetAttrCount()
Posted by g00fy on Fri, 08 Dec 2006 09:02:41 GMT
View Forum Message <> Reply to Message

Could you underbuidl your statements please? I would like to learn a thing or two .

Or just point me to some website stating that?

Garbage collection is not really bad, it just makes your memory go 'boom'. But referenced shared pointer is generally a good idea imho.

Subject: Re: [XML] Assertion when GetAttrCount()
Posted by mirek on Fri, 08 Dec 2006 10:42:11 GMT
View Forum Message <> Reply to Message

g00fy wrote on Fri, 08 December 2006 04:02Could you underbuidl your statements please? I would like to learn a thing or two .

Or just point me to some website stating that?

No, that is *U++* paradigm. So the only website I can point you to is ours (see http://www.ultimatepp.org/www$uppweb$overview$en-us.html)

In fact, I think this is the main difference between U++ and the rest of the world.

The rest of the world insists that you absolutely do need at least of these three things in your code (I mean you need either GC or shared pointers or delete things manually).

U++ proves that you don't

Quote:
Garbage collection is not really bad, it just makes your memory go 'boom'. But referenced shared pointer is generally a good idea imho.

It is the nice *idea* (just like STL . But avoiding it is even better

BTW, do not get me wrong - U++ also uses reference counting here and there, but only as implementation detail - does not expose it at interface level. That is major difference.

Mirek
