Subject: Some Shocking Facts You Probably Didn't Want to Know about U++ Framework...
Posted by mirek on Mon, 09 Nov 2020 11:29:54 GMT
View Forum Message <> Reply to Message

New CodeProject article is now online:

https://www.codeproject.com/Articles/5280535/Some-Shocking-Facts-You-Probably-Didnt-Want-to-Kno

If you like it, do not forget to vote!

Mirek

Subject: Re: Some Shocking Facts You Probably Didn't Want to Know about U++ Framework...
Posted by Oblivion on Mon, 09 Nov 2020 15:16:34 GMT
View Forum Message <> Reply to Message

Quote: https://www.codeproject.com/Articles/5280535/Some-Shocking-Facts-You-Probably-Didnt-Want-to-Kno

As usual, well written and very informative!

If only I'd had read this 10 years ago...

Best regards,
Oblivion

Subject: Re: Some Shocking Facts You Probably Didn't Want to Know about U++ Framework...
Posted by mirek on Mon, 09 Nov 2020 15:33:30 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Mon, 09 November 2020 16:16Quote: https://www.codeproject.com/Articles/5280535/Some-Shocking-Facts-You-Probably-Didnt-Want-to-Kno

As usual, well written and very informative!

If only I'd had read this 10 years ago...

Best regards,
Oblivion

If only I'd had written this 10 years ago...

Mirek

Subject: Re: Some Shocking Facts You Probably Didn't Want to Know about U++ Framework...
Posted by Xemuth on Mon, 09 Nov 2020 16:24:38 GMT
View Forum Message <> Reply to Message

Hello Mirek,

Thanks for this article, I learned so many things (and of course gived it 5 stars !)

I have the feeling that the U++ vision is very modern. It makes me happy to know that I spent my firsts years in the development on this gem and I'm super excited to read there is still so much to learn about it.

Special mention to :
Quote:that going from Vector to Array is mostly the act of rewriting "Vector" to "Array" Oh god, that's so true and appreciable, I dont count the times I swaped from one to another !

About theIDE part : I think (from my humble beginner point of view) you summarized the most interesting features of TheIDE, don't bother anymore with external libs, get the package and it will work !

Subject: Re: Some Shocking Facts You Probably Didn't Want to Know about U++ Framework...
Posted by Tom1 on Mon, 09 Nov 2020 19:46:11 GMT
View Forum Message <> Reply to Message

Hi Mirek!

Thanks!! Great article!

Although I started learning U++ 2006, I'm still learning... : An Array can actually hold objects of inherited classes! This was interesting and very useful for my needs. However, I could not figure out how to make serialization/deserialization of this work correctly. Here's my failed test of that:

```
using namespace Upp;
class A{
public:
 int a;
 A(){
```

```
  a=1;
  Cout() << "A+\n";
 }

virtual ~A(){
  Cout() << "A-\n";
 }

virtual void Serialize(Stream &s){
  s % a;
 }

virtual void Print(){
  Cout() << "A: " << a << "\n";
 }
};

class B: public A{
public:
 int b;
 B(){
  b=2;
  Cout() << "B+\n";
 }

 ~B(){
  Cout() << "B-\n";
 }

virtual void Serialize(Stream &s){
  A::Serialize(s);
  s % b;
 }

virtual void Print(){
  Cout() << "B: " << a << ", " << b << "\n";
 }
};


CONSOLE_APP_MAIN{
 Array<A> v;

 v.Create<A>();
 v.Create<B>();
 v.Create<B>();
 v.Create<A>();
```

```
 Cout() << "Printing array...\n";
 for(int i=0;i<v.GetCount();i++) v[i].Print();

 Cout() << "Storing to file...\n";
 StoreToFile(v,GetHomeDirFile("Test.bin"));
 Cout() << "Clearing...\n";

 v.Clear();

 Cout() << "Loading from file...\n";
 LoadFromFile(v,GetHomeDirFile("Test.bin"));
 Cout() << "Loaded from file...\n";

 Cout() << "Printing array...\n";
 for(int i=0;i<v.GetCount();i++) v[i].Print();
}
```

What is the correct way to do this?

Thanks and best regards,

Tom

Subject: Re: Some Shocking Facts You Probably Didn't Want to Know about U++ Framework...
Posted by Didier on Mon, 09 Nov 2020 19:48:18 GMT
View Forum Message <> Reply to Message

Very Very good article.

It should be directly in U++ documentation as it contains several coding guidelines to work well with U++

Subject: Re: Some Shocking Facts You Probably Didn't Want to Know about U++ Framework...
Posted by koldo on Mon, 09 Nov 2020 20:10:18 GMT
View Forum Message <> Reply to Message

This has been an unconventional explanation but one that cannot leave indifferent anyone with extensive experience in programming and C++.
Really great job Mirek!

Subject: Re: Some Shocking Facts You Probably Didn't Want to Know about U++ Framework...
Posted by mirek on Mon, 09 Nov 2020 21:05:58 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Mon, 09 November 2020 20:46Hi Mirek!

Thanks!! Great article!

Although I started learning U++ 2006, I'm still learning... : An Array can actually hold objects of inherited classes! This was interesting and very useful for my needs. However, I could not figure out how to make serialization/deserialization of this work correctly. Here's my failed test of that:

```
using namespace Upp;
class A{
public:
 int a;
 A(){
  a=1;
  Cout() << "A+\n";
 }

virtual ~A(){
  Cout() << "A-\n";
 }

virtual void Serialize(Stream &s){
  s % a;
 }

virtual void Print(){
  Cout() << "A: " << a << "\n";
 }
};

class B: public A{
public:
 int b;
 B(){
  b=2;
  Cout() << "B+\n";
 }

 ~B(){
  Cout() << "B-\n";
 }

virtual void Serialize(Stream &s){
  A::Serialize(s);
```

```
  s % b;
 }

virtual void Print(){
  Cout() << "B: " << a << ", " << b << "\n";
 }
};


CONSOLE_APP_MAIN{
 Array<A> v;

 v.Create<A>();
 v.Create<B>();
 v.Create<B>();
 v.Create<A>();

 Cout() << "Printing array...\n";
 for(int i=0;i<v.GetCount();i++) v[i].Print();

 Cout() << "Storing to file...\n";
 StoreToFile(v,GetHomeDirFile("Test.bin"));
 Cout() << "Clearing...\n";

 v.Clear();

 Cout() << "Loading from file...\n";
 LoadFromFile(v,GetHomeDirFile("Test.bin"));
 Cout() << "Loaded from file...\n";

 Cout() << "Printing array...\n";
 for(int i=0;i<v.GetCount();i++) v[i].Print();
}
```

What is the correct way to do this?

Seriously, you cannot... Sorry that I might have created that impression.

I mean, you definitely can solve this, but not with out-of-box Array::Serialize. You need to provide your own serialization there and store the type on saving, load on loading.

In reality, this is not needed very often...

Mirek

Subject: Re: Some Shocking Facts You Probably Didn't Want to Know about U++ Framework...
Posted by Tom1 on Tue, 10 Nov 2020 09:03:50 GMT
View Forum Message <> Reply to Message

mirek wrote on Mon, 09 November 2020 23:05
Seriously, you cannot... Sorry that I might have created that impression.

I mean, you definitely can solve this, but not with out-of-box Array::Serialize. You need to provide your own serialization there and store the type on saving, load on loading.

In reality, this is not needed very often...

Mirek

Absolutely nothing to apologize.

Actually, I have already worked my way around this by serializing type ID and then object. However, I went the harder way by using Vector<BaseClass*> instead of Array<BaseClass> since, at the time, I did not know about the possibility to store inherited classes in an Array.

Thanks and best regards,

Tom

Subject: Re: Some Shocking Facts You Probably Didn't Want to Know about U++ Framework...
Posted by Klugier on Tue, 10 Nov 2020 14:24:18 GMT
View Forum Message <> Reply to Message

Hello,

I added the link to the article in articles section of Tutorial site. So, you can easily return to it if you forget about this announcment. I would like to thank Mirek for this outstanding job - from the PR point of view it is big thing. I hope I would join - right now I am working to make internal documentation better.

Mire do you think about simplifying title - for me it seems too long. What about:
Some Shocking Facts You Probably Didn't Want to Know about U++ Framework... ->
Shocking Facts You don't Know about U++ Framework...
or even
Shocking Facts about U++ Framework...

Simple and easier to remembered. In the title we should use the same principle as we use in context of our code - KISS.

Klugier

Subject: Re: Some Shocking Facts You Probably Didn't Want to Know about U++ Framework...
Posted by mirek on Tue, 10 Nov 2020 14:37:48 GMT
View Forum Message <> Reply to Message

Klugier wrote on Tue, 10 November 2020 15:24Hello,
Mire do you think about simplifying title - for me it seems too long. What about:

It is intentional. It could have been "Introduction to U++", but where is fun in that?

Mirek

Subject: Re: Some Shocking Facts You Probably Didn't Want to Know about U++ Framework...
Posted by wimpie on Tue, 10 Nov 2020 21:53:22 GMT
View Forum Message <> Reply to Message

Well this put a big smile on my face! And informative in the mean time too

And I learned something about myself.. I'm not a real programmer! (already knew that)

really nice write-up. Keep up the good work!

Regards,

Wimpie

p.s. still laughing about the title

Subject: Re: Some Shocking Facts You Probably Didn't Want to Know about U++ Framework...
Posted by jimlef on Fri, 13 Nov 2020 16:23:35 GMT
View Forum Message <> Reply to Message

Have just received this article in my inbox - and of course 5 stars all the way Congratulations on the great press!

Subject: Re: Some Shocking Facts You Probably Didn't Want to Know about U++ Framework...
Posted by mirek on Sun, 15 Nov 2020 08:41:03 GMT
View Forum Message <> Reply to Message

I have found

https://www.reddit.com/r/cpp

and tried to post a link there. If you are reddit active, you can consider upvoting...

Mirek

Subject: Re: Some Shocking Facts You Probably Didn't Want to Know about U++ Framework...
Posted by Klugier on Mon, 16 Nov 2020 23:30:09 GMT
View Forum Message <> Reply to Message

Hello Mirek,

I see that your comment on the reddit caused controversy We collected a lot of valuable data. From my point of view - the three main causes why they do not want to use our technology are:
- impossible to write in other IDE's than TheIDE.
- not easy to integrate with other build systems. If I want to add GUI part for existing code (front-end) with U++, this is not possible. Everything needs to be integrated with ecosystem. I know that many people like to work with CMake as a build system, so why not let them work with Upp
- custom allocator is not selling point - we should sell it that our GUI is super responsive thanks to that Anyway, we should think about it and what we would like to do with that.

In my opinion we should focus on U++ as library (only core libraries Console + GUI) and provide it as collection of .so and .dlls. Thanks to that our libraries can be integrated with other build system easily. In this case we should do not bundle allocator - it can works within library, but externally it should be hidden. So, all binaries linking to upp libraries should be in USEMALLOCK mode.

I am thinking about creating tool called udeploy that will compile selected uppsrc libraries and produce folder with shared libraries (.dlls) and headers. udeploy tool should copy headers from packages. In Upp not as a platform mode we should have includes like this:

#include <Upp/CtrlLib/CtrlLib.h>

With the change described above we should get out of the niche, however the investment in this case is huge. This will also allow Upp to be used in bigger projects.

Should, we do something with aggressive keyword in README.md? I see that there are some remarks about it.

Klugier

Subject: Re: Some Shocking Facts You Probably Didn't Want to Know about U++ Framework...
Posted by mirek on Mon, 16 Nov 2020 23:39:46 GMT
View Forum Message <> Reply to Message

Klugier wrote on Tue, 17 November 2020 00:30Hello Mirek,

I see that your comment on the reddit caused controversy We collected a lot of valuable data. From my point of view - the three main causes why they do not want to use our technology are:
- impossible to write in other IDE's than TheIDE.
- not easy to integrate with other build systems. If I want to add GUI part for existing code (front-end) with U++, this is not possible. Everything needs to be integrated with ecosystem. I know that many people like to work with CMake as a build system, so why not let them work with Upp
- custom allocator is not selling point - we should sell it that our GUI is super responsive thanks to that Anyway, we should think about it and what we would like to do with that.

In my opinion we should focus on U++ as library (only core libraries Console + GUI) and provide it as collection of .so and .dlls. Thanks to that our libraries can be integrated with other build system easily. In this case we should do not bundle allocator - it can works within library, but externally it should be hidden. So, all binaries linking to upp libraries should be in USEMALLOCK mode.

I am thinking about creating tool called udeploy that will compile selected uppsrc libraries and produce folder with shared libraries (.dlls) and headers. udeploy tool should copy headers from packages. In Upp not as a platform mode we should have includes like this:

#include <Upp/CtrlLib/CtrlLib.h>

With the change described above we should get out of the niche, however the investment in this case is huge. This will also allow Upp to be used in bigger projects.

Should, we do something with aggressive keyword in README.md? I see that there are some remarks about it.

Klugier

Absolutely, that was a flop. udeploy would be great. The article style was wrong idea, posting it on reddit even worse. There is always something to learn...

Mirek

Subject: Re: Some Shocking Facts You Probably Didn't Want to Know about U++ Framework...
Posted by mirek on Mon, 16 Nov 2020 23:54:16 GMT
View Forum Message <> Reply to Message

Klugier wrote on Tue, 17 November 2020 00:30Hello Mirek,

- custom allocator is not selling point - we should sell it that our GUI is super responsive thanks to that Anyway, we should think about it and what we would like to do with that.

In my opinion we should focus on U++ as library (only core libraries Console + GUI) and provide it as collection of .so and .dlls. Thanks to that our libraries can be integrated with other build system easily. In this case we should do not bundle allocator - it can works within library, but externally it should be hidden. So, all binaries linking to upp libraries should be in USEMALLOCK mode.

U++ allocator was not even mentioned. And in fact, it might be a great idependent piece of code.

As for allocator modes within U++, it is already ready because on MacOS you cannot overload global new/delete. So the library can already be configured in a mode that is using regular new/delete and U++ allocator for internals where possible.

Once again, if somebody would step in and provided regular ecosystem libraries, that would be excellent news. That OTOH does not mean I am going to abandon theide...

Mirek

Subject: Re: Some Shocking Facts You Probably Didn't Want to Know about U++ Framework...
Posted by Klugier on Tue, 17 Nov 2020 00:09:17 GMT
View Forum Message <> Reply to Message

Hello Mirek,

Thanks that you gave the green light for upp as a library. I think we could target it in the next year - this will be massive rework. The main problem is that for dll you need to explicitly said which symbols should be exported... I think we will try to participate in GSoC 2021 with only one idea. This seems to be top priority now. I plan to have short break from U++...

Before start thinkg of udeploy. I would see that we will create CommandLineArgumentParser class for U++ ecosystem (should be in uppsrc). This will be similar concept to QT Parser or cxxopts. We need to make sure that our connsole apps API looks great. We do not want to have similar problem to this one.

Backing to allocator - thanks for the clarification. I had to understand something wrongly from the discussion. I think it was UB

Klugier

Subject: Re: Some Shocking Facts You Probably Didn't Want to Know about U++ Framework...
Posted by mirek on Tue, 17 Nov 2020 09:10:40 GMT
View Forum Message <> Reply to Message

Klugier wrote on Tue, 17 November 2020 01:09Hello Mirek,

Thanks that you gave the green light for upp as a library.

That was always green light. I have even changed important aspects in the past, .icpp is no longer required. I just do not have energy (and mood) to do that myself.

Quote:
I think we could target it in the next year - this will be massive rework. The main problem is that for dll you need to explicitly said which symbols should be exported...

Produce static libraries then.... You can also simply export everything, there are methods for that. But I would rather do statics, in the past there was stigma associated with that, but I feel like that is now gone. In any case, it would be a good first step.

The real problem with this is how to destroy U++ modularity... E.g. how will you handle MySQL? SSL?

Quote:
Backing to allocator - thanks for the clarification. I had to understand something wrongly from the discussion. I think it was UB

No, Moveable is the UB. And probably not for much longer as this kind of UB is "established industry practice" now, Bloomber, Facebook and Electronics Arts template libraries are doing exactly the same thing and there are proposals to make it legal.

Mirek

Subject: Re: Some Shocking Facts You Probably Didn't Want to Know about U++ Framework...
Posted by mirek on Wed, 18 Nov 2020 11:22:44 GMT
View Forum Message <> Reply to Message

Klugier wrote on Tue, 17 November 2020 01:09Hello Mirek,
Thanks that you gave the green light for upp as a library. I think we could target it in the next year - this will be massive rework. The main problem is that for dll you need to explicitly said which symbols should be exported... I think we will try to participate in GSoC 2021 with only one idea. This seems to be top priority now. I plan to have short break from U++...

I was thinking about it and I believe the whole issue can be simpified immensely by adding a couple of flags to umk!

Simply put, upgrade ide/Builders to be able to produce a .lib file as the end product and we are halfway there, right?

Mirek

Subject: Re: Some Shocking Facts You Probably Didn't Want to Know about U++ Framework...
Posted by Oblivion on Wed, 18 Nov 2020 13:35:20 GMT
View Forum Message <> Reply to Message

Quote:I would see that we will create CommandLineArgumentParser class for U++ ecosystem (should be in uppsrc). This will be similar concept to QT Parser or cxxopts. We need to make sure that our connsole apps API looks great.

Nice one. I don't like to copy/paste code all the time. If nobody picks this up, I can give it a try after I finalize Turtle login improvements, and the Core/Markdown package.

Best regards,
Oblivion

Subject: Re: Some Shocking Facts You Probably Didn't Want to Know about U++ Framework...
Posted by mirek on Wed, 18 Nov 2020 14:47:18 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Wed, 18 November 2020 14:35Quote:I would see that we will create CommandLineArgumentParser class for U++ ecosystem (should be in uppsrc). This will be similar concept to QT Parser or cxxopts. We need to make sure that our connsole apps API looks great.

Nice one. I don't like to copy/paste code all the time. If nobody picks this up, I can give it a try after I finalize Turtle login improvements, and the Core/Markdown package.

Best regards,
Oblivion

I have started something little when ppl wanted to add more verbose interface to umk... Maybe I will dig it up.

Subject: Re: Some Shocking Facts You Probably Didn't Want to Know about U++ Framework...
Posted by mirek on Sun, 22 Nov 2020 16:49:13 GMT
View Forum Message <> Reply to Message

Klugier wrote on Tue, 17 November 2020 01:09
Thanks that you gave the green light for upp as a library.

https://www.ultimatepp.org/forums/index.php?t=msg&th=112 72&start=0&

Will probably be much simpler that I thought the whole time, but needs some work...

Subject: Re: Some Shocking Facts You Probably Didn't Want to Know about U++ Framework...
Posted by 281264 on Mon, 22 Mar 2021 19:51:53 GMT
View Forum Message <> Reply to Message

Thank you for the article. Crystal clear and, indeed, quite informative.

BR,

Javier
