

---

Subject: [Tutorial] Using the Visual Studio IDE with U++

Posted by [crydev](#) on Wed, 17 Oct 2012 09:25:09 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hi there, I have been using U++ now for a few weeks. I like the library and the way of programming. However, there is one thing that I find very annoying. The packaged IDE "TheIDE". It lacks debugging features in my opinion and does not sufficiently supports me in my work with intellisense and auto-completion.

After two hours I managed to get my whole U++ application including libraries into Visual Studio 2012. It compiles, runs and now debugging is much easier to do. This topic is meant to explain to the community what I did, in case of any others having the same opinion as I have or just want to use the Visual Studio IDE anyway.

I use Visual Studio 2012, but this will probably work with 2010 and 2008 too. I'm not sure about 2005.

1. First you have to open TheIDE and compile every U++ package that you want to use in Debug mode. Make sure you have TheIDE configured with the compiler of the Visual Studio version you want to use!
2. Create a new folder somewhere where you can find it and place every output .lib file in there. With these files I mean Core.lib, CtrlLib.lib, etc.
3. Open up Visual Studio and create a new C++ Win32 project. Make it an empty project.
4. Open up the properties of the project by right-clicking on the project and clicking Properties.
5. Go to C/C++ and add the <upp directory>\uppsrc\ to the list of Additional Include Directories
6. Go to Code Generation and set Runtime Library to Multi-Threaded Debug (/MTd). Do not set it to MDD! This will generate errors.
7. Go to Linker and click General. Click Additional Library Directories to open up the list. Add the folder where you placed the compiled U++ lib files from step 2 to the list and click OK.
8. Go to Linker and click Input. Click Additional Dependencies to open up the list. Add the following libraries to the list:
  - Winmm.lib
  - Mpr.lib
  - CtrlLib.lib
  - Core.lib
  - CtrlCore.lib
  - bmp.lib
  - png.lib
  - z.lib
  - Draw.lib
  - RichText.lib

These libraries ofcourse are the ones my application needed. You also need to link a few Win32 libraries because the U++ libs need these.

9. Rename every .icpp file that does not compile in your project to .cpp.

10. If you are using .iml image files you need to change the include from:

```
#include <Folder/imlfile.iml>
```

to:

```
#include "imlfile.iml"
```

Try and compile your project. It should work as it did for me. There is one problem that I got by. However, if you want to use Visual Studio only for debugging this should not be a big problem. When you run your program from Visual Studio the GUI is not drawn properly. Buttons are black, borders are missing and arrayctrl headers are not drawn properly. If there are any other problems please report them.

---

---

Subject: Re: [Tutorial] Using the Visual Studio IDE with U++

Posted by [mirek](#) on Wed, 17 Oct 2012 14:16:44 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

It is fine, except it does not solve .icpp problem.

Things like loading image in any format will not work (because image plugins will not get registered).

Mirek

---

---

Subject: Re: [Tutorial] Using the Visual Studio IDE with U++

Posted by [kohait00](#) on Thu, 18 Oct 2012 07:29:48 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

would be great to offer means on how to use upp in eclipse as well.. i think Upp would be percieved as even more versatile if that were the case.

something like

"yeah, u have theide to use upp with, but if u want to use tool X for developping, this is how to do it best way, but be aware of drawback A, B and C"

i think proper package and assemblies management would be an additional case to consider..

i'm recently working with eclipse in work and kinda become used to it, would be a nice option to have all in one place (still enjoying development in the ide though).

---

Subject: Re: [Tutorial] Using the Visual Studio IDE with U++  
Posted by [crydev](#) on Thu, 18 Oct 2012 08:58:55 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Quote:It is fine, except it does not solve .icpp problem.

Things like loading image in any format will not work (because image plugins will not get registered).

Sorry I forgot my solution to these problems. As for the .icpp files: I renamed them to .cpp and they compiled.

The image .iml files need a different include. You need to put "#include "<header>.iml" instead of "#include <folder/imlfile.iml>"

I also forgot to post a common problem with this solution which I think has to do with one of the issues you mention. When you run your program the GUI is not proper. The border are missing, buttons are black on some places and arrayctrl headers are not drawn properly.

I will add these to the OP.

---

Subject: Re: [Tutorial] Using the Visual Studio IDE with U++  
Posted by [mirek](#) on Thu, 18 Oct 2012 09:54:47 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

crydev wrote on Thu, 18 October 2012 04:58Quote:It is fine, except it does not solve .icpp problem.

Things like loading image in any format will not work (because image plugins will not get registered).

Sorry I forgot my solution to these problems. As for the .icpp files: I renamed them to .cpp and they compiled.

But that does not solve the problem.

Please test this app:

[http://www.ultimatepp.org/examples\\$ImageView\\$en-us.html](http://www.ultimatepp.org/examples$ImageView$en-us.html)

Without correct .icpp handling, it would not work, because

StreamRaster::OpenAny

will return nothing, as registration in .icpp files is not performed.

Quote:

I also forgot to post a common problem with this solution which I think has to do with one of the issues you mention. When you run your program the GUI is not proper. The border are missing, buttons are black on some places and arrayctrl headers are not drawn properly.

Yes, very likely.

The definition of .icpp files is that all global constructors in these are performed on startup. If you put those into .lib, linker will not link these files into .exe (as there are no references to the code) and thus you e.g. will not have png plugin available for StreamRaster::OpenAny.

Mirek

---

Subject: Re: [Tutorial] Using the Visual Studio IDE with U++

Posted by [mirek](#) on Thu, 18 Oct 2012 09:55:56 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

kohait00 wrote on Thu, 18 October 2012 03:29 would be great to offer means on how to use upp in eclipse as well.. i think Upp would be percieved as even more versatile if that were the case.

something like

"yeah, u have theide to use upp with, but if u want to use tool X for developping, this is how to do it best way, but be aware of drawback A, B and C"

i think proper package and assemblies management would be an additional case to consider..

i'm recently working with eclipse in work and kinda become used to it, would be a nice option to have all in one place (still enjoying development in theide though).

Would be great. Perhaps it would be possible to make Eclipse use .upp files and umk as plugin?

Mirek

---

Subject: Re: [Tutorial] Using the Visual Studio IDE with U++

Posted by [kohait00](#) on Thu, 18 Oct 2012 10:45:09 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

i'll try it soon..

a good point to become familiar with eclipse  
and..android integration is waiting too..

---

---

Subject: Re: [Tutorial] Using the Visual Studio IDE with U++

Posted by [koldo](#) on Thu, 18 Oct 2012 18:41:11 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hello crydev

I think yours has been a good achievement going to the right track.

It would be great that full U++ could be debugged and run from other IDEs like Eclipse and Visual Studio.

It seems that the last problem are .icpp files. Perhaps with little effort they could be handled properly from other IDEs. If not perhaps we could consider to find a way to avoid them.

---

---

Subject: Re: [Tutorial] Using the Visual Studio IDE with U++

Posted by [mirek](#) on Fri, 19 Oct 2012 07:34:16 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

koldo wrote on Thu, 18 October 2012 14:41Hello crydev

I think yours has been a good achievement going to the right track.

It would be great that full U++ could be debugged and run from other IDEs like Eclipse and Visual Studio.

It is actually possible to use Visual Studio as debugger invoked from theide (that what that "External debugger" in build methods is useful for).

Quote:

It seems that the last problem are .icpp files. Perhaps with little effort they could be handled properly from other IDEs. If not perhaps we could consider to find a way to avoid them.

We tried in the past: you can notice there are "init" files in most directories (generated by theide), in theory all you need is to include "init" files of packages that you are directly using in one .cpp file (like `#include <CtrlLib/init>`), which in turn will include "init" of packages they are using. If package contains ".icpp", such file gets included by its "init".

Check plugin/png/init

Anyway, this will not resolve some other issues (like .brc files), so theide cannot be compiled this way. But perhaps it is OK for most of development.

What seems even more potentially promising is using umk and providing some sort of Package organizer plugins (for Eclipse or any other similar ide)...

(That said, theide is still my preferred hammer, so hopefully somebody else will do it. I have quite low motivation here...)

Mirek

---

---

Subject: Re: [Tutorial] Using the Visual Studio IDE with U++

Posted by [kohait00](#) on Fri, 19 Oct 2012 08:54:15 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

ofcourse it just makes sense to use ide to dev upp.

to include upp packages in own existing eclipse development would show the versatility of upp in a great way..

- \* use is as a self contained dependency free framework, building and debuggin in own the ide space

- \* extend your capabilities by binding your existing (VS or eclipse or wahtever) projects with upp packages.. (providing only very small glue logic / wrapper classes if at all).

- \* develop entirely in another IDE, but benefit from everything upp offers as source base.

since eclipse is my current dvelopment environment at work i will be persuing it slowly but steadily..

EDIT: maybe i can use doliks universal makefile generation for the purpose of eclipse building.. as a prebuild step or sth..

---

---

Subject: Re: [Tutorial] Using the Visual Studio IDE with U++

Posted by [dolik.rce](#) on Fri, 19 Oct 2012 10:20:51 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

kohait00 wrote on Fri, 19 October 2012 10:54EDIT: maybe i can use doliks universal makefile generation for the purpose of eclipse building.. as a prebuild step or sth..

Note that the Makefile is now generated by UppBuilder package, which allows to easily write specialized tools based on the same parser of U++ files. So if you need some special handling for

eclipse plugin or something, it can probably be done rather easily

Honza

---

---

Subject: Re: [Tutorial] Using the Visual Studio IDE with U++

Posted by [lectus](#) on Sat, 15 Dec 2012 18:11:02 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Honestly I prefer to use Thelde.

Having only one IDE running on all OSes U++ supports is great. And Thelde has all features I need.

Yes, it's a good effort to compile U++ on Visual Studio, but I'd rather open a package in Thelde and it builds without any path fiddling because everyone is using Thelde too.

AFAIK that's one of the problems with wxWidgets. It has too many IDEs and GUI editors but no official one.

---

---

Subject: Re: [Tutorial] Using the Visual Studio IDE with U++

Posted by [crydev](#) on Sun, 29 Sep 2013 21:06:16 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

I found the solution to the icpp problem in Visual Studio. When compiling in ThelIDE, the .icpp files are passed to the compiler seperately using /Tp and linked as .obj files together with the library files.

First of all: Forget the renaming of RichImage.icpp to RichImage.cpp. The linker error you get will be solved if you follow the steps below.

To link the object files properly in Visual Studio, set ThelIDE to be verbose, copy the link command in notepad and filter out all the .obj files that are not included in your own package. Copy all of these strings to 'Additional command lines options' in Visual Studio. If all went well it should work now.

---

---

Subject: Re: [Tutorial] Using the Visual Studio IDE with U++

Posted by [dolik.rce](#) on Mon, 30 Sep 2013 04:17:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

crydev wrote on Sun, 29 September 2013 23:06 To link the object files properly in Visual Studio, set ThelIDE to be verbose, copy the link command in notepad and filter out all the .obj files that are not included in your own package. Copy all of these strings to 'Additional command lines options' in Visual Studio. If all went well it should work now.

Could this be automated? It seems like annoyance to manually set this up for each project,

especially if it requires running TheIDE...

I'm not using Visual Studio (not even windows), but I support all attempts to make U++ work without TheIDE. To be able to use it with only a makefile, I created minimalistic .upp parser. It is written in such way that it allows adding multiple backends for various purposes, so it could perhaps be used also to automate the build process in VS. You can find the source in the svn repository. It is partially documented, but if you have trouble understanding something, I can help. I'd try to do this myself, but I just don't have the platform to test it on. So you'll have to figure out how to make it work with VS, and I can help you generate the proper command lines

Best regards,  
Honza

---

---

Subject: Re: [Tutorial] Using the Visual Studio IDE with U++  
Posted by [Dsonophorus](#) on Tue, 29 Oct 2013 17:37:08 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

I'm just getting started with UPP. Here is another quick and dirty integration of UPP with visual studio. I have included the sln as an example. I use it as a template project. As with the above method, it must be linked with winmm.lib and mpr.lib from windows.

beyond that, I just copied a version of uppsrc into the directory of this project, but it can reside anywhere as long as you set your include directory to it in the project configuration.

I included all the .cpp files into a single file called core.cpp. The .icpp files I renamed as .cpp The .c file were less encapsulated so I added the .c files from the plugin/bmp, plugin/png and plugin/z directories manually.

As was noted the buttons and some borders draw black - even in the release version, so that still needs some solution.

I really like the work done on UPP. It is very clean and tight in many important ways. The "code locking" into theide is questionable so perhaps something like this can help.

### File Attachments

1) [UPP\\_VS\\_SLN.zip](#), downloaded 397 times

---

---

Subject: Re: [Tutorial] Using the Visual Studio IDE with U++  
Posted by [mirek](#) on Wed, 30 Oct 2013 06:56:09 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Wrt .icpp issue:

Has any of you have tried 'init' files yet? Those are supposed to solve the issue, these automatically generated (by theide) files reside in each package directory and include all .icpp



files of package and all 'init' files of used packages (with include guards).

So instead of other .icpp tricks, it should be possible to include all init files of used packages into VS++ file with main....

Mirek

---

---

Subject: Re: [Tutorial] Using the Visual Studio IDE with U++  
Posted by [Dsonophorus](#) on Wed, 30 Oct 2013 20:49:30 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

O yeah, that is cleaner than what I just did. I will probably go that route so the library isn't modified.

I redefined the MK\_\_s token to use a macro counter instead of the apparently malfunctioning \_\_LINE\_\_. I then systematically (through the entire UPP library) incremented the counter everywhere MK\_\_s was implicitly referenced - so mostly a global replace of the INITBLOCK, EXITBLOCK symbols.

In some ways it is cleaner to add a line instead of a file. If you want to get rid of the init files in a future version, feel free to rip any of my macro code you can use for that. I have more macro code when I was investigating automated class generation and type lists for templates.

---

### File Attachments

1) [MacroUtilities.zip](#), downloaded 414 times

---

---

Subject: Re: [Tutorial] Using the Visual Studio IDE with U++  
Posted by [mirek](#) on Thu, 31 Oct 2013 06:36:26 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Dsonophorus wrote on Wed, 30 October 2013 16:49O yeah, that is cleaner than what I just did. I will probably go that route so the library isn't modified.

I redefined the MK\_\_s token to use a macro counter instead of the apparently malfunctioning \_\_LINE\_\_.

Weird, I would say, w.r.t. 'init', BLITZ\_INDEX\_\_ defines (already in init) are supposed to solve this issue....

(It is called 'BLITZ\_INDEX\_\_' because the original purpose is to make BLITZ work, but the required functionality is the same - alter MK\_\_s for various files).

On related note, I will soon alter theide so that it can serve as file editor (if filename given on commandline). That way, it would be possible to use theide from VS (or Eclipse or anything) as external editor to edit .lay and .iml files...

Mirek

---