Subject: U++ and Managed c++ /clr
Posted by conrad on Sat, 05 Nov 2011 18:50:33 GMT
View Forum Message <> Reply to Message

I'd like to give U++ a serious try but I need to integrate with a few .net libraries (On Windows). I am wondering if anyone has tried to combine /clr (managed c++) code with the U++ framework and TheIde?
Regards,
Conrad

Subject: Re: U++ and Managed c++ /clr
Posted by Sender Ghost on Sun, 06 Nov 2011 06:12:14 GMT
View Forum Message <> Reply to Message

Hello, Conrad.

I didn't try up to this moment, but this is definitely possible with knowledge of how TheIDE works.

- Open "Setup -> Build methods.." menu and duplicate existing MSC builder method.
- Remove "-GS-" from Optimize for speed and Optimize for size compiler options.
- Select Shared libs for Release and Debug mode defaults.

File Attachments
1) MSC9_build_method.png, downloaded 1107 times

Subject: Re: U++ and Managed c++ /clr
Posted by Sender Ghost on Sun, 06 Nov 2011 06:15:45 GMT
View Forum Message <> Reply to Message

Let's create simple "Core console project" application:

```
#include <Core/Core.h>
using namespace Upp;

#using <mscorlib.dll>
using namespace System;

CONSOLE_APP_MAIN
{
 Cout() << "Hello, from U++!\n";
 Console::WriteLine("Hello, from managed C++!");
}
```

Open "Project -> Package organizer.." for main package and insert following compiler option:

-clr

Open "Project -> Main package configuration.." menu and add following build flags:

EVC .USEMALLOC

EVC - is build flag (used for another C++ compiler), which adds compiler options to turn off exception handling, because -clr is not compatible with -EHsc compiler option.
USEMALLOC - disables U++ memory allocator to not conflict with .NET Framework memory allocation methods.

Also check "Build -> Output mode.." menu for the "Use shared libs" link mode.

Now, you can build combined U++ and Managed C++ package.

To note:
It also possible to extend C++ syntax higlighting for managed C++ extensions and add CLR specific compiler options, instead of EVC.
Here, I just showed possible way to use TheIDE with Managed C++ extensions. I don't think, this is recommended way.

File Attachments

1) ManagedCpp.zip, downloaded 321 times

Subject: Re: U++ and Managed c++ /clr
Posted by mirek on Sun, 06 Nov 2011 07:40:56 GMT
View Forum Message <> Reply to Message

Maybe it is time for 'official' investigation and support:

http://www.ultimatepp.org/redmine/issues/156

Subject: Re: U++ and Managed c++ /clr
Posted by Sender Ghost on Sun, 06 Nov 2011 12:04:36 GMT
View Forum Message <> Reply to Message

Here is screenshot for TheIDE, with applied patches.

It shows U++ package for GUI application with combined managed C++ and U++ source code.

File Attachments
1) ManagedGUI.png, downloaded 899 times

Subject: Re: U++ and Managed c++ /clr
Posted by conrad on Sun, 06 Nov 2011 16:12:37 GMT
View Forum Message <> Reply to Message

Many thanks for all the answers here.
I must say, I have rarely if ever experienced such support.
I will give this a try this week.
I have a fair amount of experience with mixed mode solutions running in VS 2009 and 2010.
The system I need to make work is one I call for the "firewall" solution. Let me explain.
I do not want the entire app to bee "seen" by the /clr compiler but only a wrapper "project" (a Package ?). The reason for this is that once the application grows, mixed mode compilation & linking becomes completely unmanageable. Footprint is 2.5 times fold and link times can be literally hours.
For this to work, a few .h header files will have to be compatible for both c++ and /clr compilation. With VS 2010 doing the compilation, std::string and std::wstring is for example a workable candidate. In a VS solution, these wrapper classes live in a unique project and only this project has /clr turned on. /clr only, not /pure, not /safe - as these create far too many issues. But /clr works, and I know for a fact that given a few wrapper classes where essentially plain old data and strings are converted from the .net world to the c++ world, mixed mode can have much to offer.

I will let you know how I am doing as soon as I know a bit more.
Again, thanks for the support here without which I may not have had the courage to give it a try

Conrad

Subject: Re: U++ and Managed c++ /clr
Posted by conrad on Sun, 06 Nov 2011 19:26:59 GMT
View Forum Message <> Reply to Message

conrad wrote on Sun, 06 November 2011 17:12
I will let you know how I am doing as soon as I know a bit more.
Conrad

I understand the concept of Packages a bit better now.
I made a "DotNetWrapper" package with -clr.
It looks like I need a way to pass on more compiler options to VS.
In particular the /MDd option must override the /MTd option.
(Adding -EHa seems to work, but adding -MDd doesn't)

I am using the HelloWorld example.

I don't think I need to change the default setting of the application. Only the wrapper package must be build with correct options.

I have read Miroslav's http://www.ultimatepp.org/redmine/issues/156 and looked at the diffs but that is going a bit too fast for me Besides, looks like I would have to recompile theIde first after pull in those changes.

Regards,
Conrad

Subject: Re: U++ and Managed c++ /clr
Posted by Sender Ghost on Sun, 06 Nov 2011 20:20:10 GMT
View Forum Message <> Reply to Message

With patches up to 4140 revision you still need specify -clr compiler option for particular package. The CLR build flag will use -EHac instead of -EHsc compiler option. Also, it will build like it uses shared libraries.
No need to remove -GS- compiler option from build method.

To build TheIDE (from uppsrc/ide package) you need latest svn sources or use automated build for latest revision.

Subject: Re: U++ and Managed c++ /clr
Posted by mirek on Sun, 06 Nov 2011 21:25:45 GMT
View Forum Message <> Reply to Message

Sender Ghost wrote on Sun, 06 November 2011 15:20
To build TheIDE (from uppsrc/ide package) you need latest svn sources or use automated build for latest revision.

Note: Automated builds are performed each night... (europe time)

Subject: Re: U++ and Managed c++ /clr
Posted by conrad on Mon, 07 Nov 2011 06:10:37 GMT
View Forum Message <> Reply to Message

I uninstalled upp and installed the latest 4140 build.
(Gave me a change to target a directory where I don't need to run in administration mode).
I am still having the same issue.
I have added the following compiler options to a new package:
-EHa, -MDd, -clr. (Order seems important). I get the following error when building the package:

----- DotNetWrappers (GUI MSC10 DEBUG DEBUG_FULL BLITZ WIN32 MSC)
cl : Command line warning D9025 : overriding '/EHs' with '/EHa'

cl : Command line warning D9025 : overriding '/MDd' with '/MTd'
cl : Command line error D8016 : '/clr' and '/MTd' command-line options are incompatible

Observe, my own -MDd override seems to be put back to /MTd by theIde.

Conrad

Subject: Re: U++ and Managed c++ /clr
Posted by Sender Ghost on Mon, 07 Nov 2011 06:20:58 GMT
View Forum Message <> Reply to Message

conrad wrote on Mon, 07 November 2011 07:10
I am still having the same issue.
I have added the following compiler options to a new package:
-EHa, -MDd, -clr.
You do not need the -EHa, -MDd compiler options. The CLR build flag will take care about this.

I attached example with GUI CLR build flags.

File Attachments
1) ManagedGUI.zip, downloaded 320 times

Subject: Re: U++ and Managed c++ /clr
Posted by conrad on Mon, 07 Nov 2011 07:07:38 GMT
View Forum Message <> Reply to Message

Ok. I now have the following (using the HelloWorld example main package).
Added new package DotNetWrappers to the HelloWorld package. Only compiler option set in DotNetWrappers package is -clr.
Added CLR option to HelloWorld's Main package configuration(s).
This is the one that confused me as I don't want to turn everything into CLI.
But it looks like this "global" CLR option only invokes the -clr compiler if the package has -clr defined.

I can now build without errors but I can't run because I have lost _main.
So, how do I get _main back in?
I am getting closer don't you think
Conrad

----- CtrlLib (CLR MSC10 DEBUG DEBUG_FULL BLITZ WIN32 MSC) (1 / 10)
----- DotNetWrappers (CLR MSC10 DEBUG DEBUG_FULL BLITZ WIN32 MSC) (2 / 10)
----- CtrlCore (CLR MSC10 DEBUG DEBUG_FULL BLITZ WIN32 MSC) (3 / 10)
----- Draw (CLR MSC10 DEBUG DEBUG_FULL BLITZ WIN32 MSC) (4 / 10)
----- plugin/bmp (CLR MSC10 DEBUG DEBUG_FULL BLITZ WIN32 MSC) (5 / 10)

----- RichText (CLR MSC10 DEBUG DEBUG_FULL BLITZ WIN32 MSC) (6 / 10)
----- Core (CLR MSC10 DEBUG DEBUG_FULL BLITZ WIN32 MSC) (7 / 10)
----- plugin/z (CLR MSC10 DEBUG DEBUG_FULL BLITZ WIN32 MSC) (8 / 10)
----- plugin/png (CLR MSC10 DEBUG DEBUG_FULL BLITZ WIN32 MSC) (9 / 10)
----- HelloWorld (CLR MAIN MSC10 DEBUG DEBUG_FULL BLITZ WIN32 MSC) (10 / 10)
Linking...
MSVCRTD.lib(crtexe.obj) : error LNK2019: unresolved external symbol _main referenced in function ___tmainCRTStartup
E:\upp\out\examples\MSC10.Clr.Debug.Debug_Full\HelloWorld.ex e : fatal error LNK1120: 1 unresolved externals

---

Subject: Re: U++ and Managed c++ /clr
Posted by Sender Ghost on Mon, 07 Nov 2011 08:25:40 GMT
View Forum Message <> Reply to Message

conrad wrote on Mon, 07 November 2011 08:07
I am getting closer don't you think :)

I think, you need to look packages from tutorial, reference, examples, bazaar assemblies to get familiar with them and reading documentation by parallel. Especially, how packages, assemblies and nests organized.

I prepared example with two packages. The library package with .Net Framework functions/classes and example package, which uses library calculation results.

File Attachments
1) ManagedExample.zip, downloaded 284 times

Subject: Re: U++ and Managed c++ /clr
Posted by conrad on Mon, 07 Nov 2011 09:05:18 GMT
View Forum Message <> Reply to Message

I seemed to have found a reason for my trouble:
In the Main package configuration, I had added a separate "row" for CLR. I removed that row and changed the GUI row to "GUI CLR".
Now I can run the HelloWorld app
Funny thing is, my anti-virus blocked it as a "Potential Malicious application". Ehm.. Oh well. Live dangerously
Any way, thanks for all the help here. I seem to have something to work with now.
Conrad
