Subject: Question about debugging dll
Posted by digital.raymond on Tue, 22 Jan 2008 23:23:06 GMT
View Forum Message <> Reply to Message

Hi,

First of all, bravo for this great piece of work !

I am very new to u++ and not really familiar with u++ concept, my question may seems dumb... but don't be too rude with me !

Here's my question:
I am creating a dll with one exported function 'Test'.
Is it possible to run it in debug mode (F5) by passing arguments to the debugger, something like rundll32.exe mydll,Test ?

Subject: Re: Question about debugging dll
Posted by digital.raymond on Wed, 23 Jan 2008 17:20:48 GMT
View Forum Message <> Reply to Message

No answer ...
Am i missing something or there are easier ways to debug dll ?
Tips/hints/links greatly appreciated !

Subject: Re: Question about debugging dll
Posted by bytefield on Wed, 23 Jan 2008 22:10:20 GMT
View Forum Message <> Reply to Message

You can debug a dll by make it a program. Put a main function in it , compile and run it as a program. If it work well then you can remove the unwanted main and compile it again as a dll and use it in your programs. You cannot (guess) run a dll(alone) in a debugger since there is no call to its functions(how you test it?).
However the Upp is made with statically linking in mind (see dll hell). Hope i was usefull(but don't know if it is the right answer).

Subject: Re: Question about debugging dll
Posted by digital.raymond on Wed, 23 Jan 2008 22:52:37 GMT
View Forum Message <> Reply to Message

bytefield wrote on Wed, 23 January 2008 23:10You can debug a dll by make it a program. Put a main function in it , compile and run it as a program. If it work well then you can remove the unwanted main and compile it again as a dll and use it in your programs.
Yes, i can do that, but it cost me too much effort and i'm lazy ...

bytefield wrote on Wed, 23 January 2008 23:10You cannot (guess) run a dll(alone) in a debugger since there is no call to its functions(how you test it?).
Totally agree with that, but look at my first post, i give an elegant (and easy) workaround:
use of rundll32.exe (from windows\system32) which accept two arguments, the name of a dll and an entry point (exported function).
In my case, i want to debug indirectly mydll.dll by using the following syntax:
rundll32.exe mydll.dll,Test
But i don't know how to pass it to the debugger !

bytefield wrote on Wed, 23 January 2008 23:10However the Upp is made with statically linking in mind (see dll hell). Hope i was usefull(but don't know if it is the right answer).
I don't have the choice, i am bound to DLL (and Hell !)

Anyways, your interest is greatly appreciated.

Subject: Re: Question about debugging dll
Posted by mr_ped on Thu, 24 Jan 2008 01:01:08 GMT
View Forum Message <> Reply to Message

If it is too much effort to do a helper main, then why do you even bother to debug at all?

Why are you bound to dll anyway? Maybe you can solve that dependency in other way, forget about dll, and work just on the code itself than happily.
(So far in my little U++ projects I never actually bothered what package resulted into dll and which got linked statically... it just works, and that's all I want to know about it, I want to code, not to babysit compilation tools and OS binary loader.)

If that DLL is already finished, and being called by some other process, you may use MS VisualStudio debugger to attach to that dll process. Not very easy, especially if it is compiled without debug info, I'm considering myself assembler wizard, but still I avoided anything like this for any costs, so I debugged binary code only several times in life, usually when I needed to bend some application from somebody else to do something differently.

If you are starting to code that DLL from scratch, why don't you use TDD? If the DLL interface is simple enough, you will probably not to debug it at all, if you have good enough suite of automated tests.

And one more thing. The debugger in TheIDE is very weak and incomplete. I'm not sure if I know any less powerful debugging tool. It's usually enough to detect place of problem, but impossible to do very basic (debugger) things like to modify memory/code, and watches are not persistent, etc. etc...
In Luzr's eyes it's complete, but I have seen in my life many different tools so I can compare. In recent years I don't need too much those things I named anymore, as I prefer not to debug my code at all, I rather understand what I write and let the rest of bugs to be discovered by tests, as debugging is very costly operation. But when I do, I want that tool to be total wizard so I can do some magic whenever I wish to, not just viewer over process execution to let me peek how it's going, but unable to affect anything.

Subject: Re: Question about debugging dll
Posted by mr_ped on Thu, 24 Jan 2008 01:07:24 GMT
View Forum Message <> Reply to Message

About running it trough rundll32.exe ...
I'm afraid the IDE is not enough flexible in the "Run" options to hack it like this.

You may try to use "Working directory" in Debug/Run options... to point directly to rundll32.exe with space on end, but I'm afraid TheIDE will not simply append the final .dll after compilation, so it will not work. It's up to you to try.

Still you will get very weak debugger, better get some external full debugger or MSVS and debug that dll in it. TheIDE is nice tool (especially for development), but far from the switz-army-knife enterprise tools (which sometimes forget they are supposed to be used for development in the first place).

Subject: Re: Question about debugging dll
Posted by digital.raymond on Thu, 24 Jan 2008 09:17:40 GMT
View Forum Message <> Reply to Message

Wooo... Not debugging my code... using TDD...
Maybe i'm too old for coding .

I thought that u++ debugger was as mature as the rest of u++ technologies... I show too much confidence in it !

Thanks mr_ped, even for the sad news

Subject: Re: Question about debugging dll
Posted by mr_ped on Thu, 24 Jan 2008 10:29:53 GMT
View Forum Message <> Reply to Message

digital.raymond wrote on Thu, 24 January 2008 10:17Wooo... Not debugging my code... using TDD...
Maybe i'm too old for coding .

That was my first thought too, like where's the fun of hacking in hexa bytes ...

But you really should try it some day, it feels somewhat like good old days.
http://en.wikipedia.org/wiki/Test-driven_development
"Programmers using pure TDD on new ("greenfield") projects report they only rarely feel the need to invoke a debugger."

BTW, I added UnitTest++ as U++ package to bazaar in SVN, so it's now very easy to start with TDD in U++.
(I should probably do some tutorial too? I don't know.)

Subject: Re: Question about debugging dll
Posted by digital.raymond on Thu, 24 Jan 2008 13:38:22 GMT
View Forum Message <> Reply to Message

When reading other posts about DLL, i feel like limits of U++ about DLL seems more ideological than technical. I always like C++ because it let you choose your rope for your own hanging (i don't know if my translation is really understandable)...

BTW, why did you choose UnitTest++ among all the others ?

Subject: Re: Question about debugging dll
Posted by mr_ped on Thu, 24 Jan 2008 14:38:40 GMT
View Forum Message <> Reply to Message

I can perfectly understand that rope, nice description of C++.

Yes, it's more ideological, after all the U++ source is available, so from technical point of view you may add DLL debug support there probably in hours if you know what you are doing.
IMHO the weak debugger is more of a problem here.

Also unless you absolutely must work with DLL, it's better to learn to work with U++ packages .. and whenever you feel your package is mature enough, you can somehow produce DLL from it with U++ anyway. (but I don't know what setup steps it would take)
But if your all application can be done in U++, you may forget about DLL at all.

Why UnitTest++. Because of it's simplicity, performance and portability. I often work on embedded platforms (some of them don't even have C++ compiler, only C one, don't even ask), and it was one of UnitTest++ goals to be as selfcontained and portable as possible.
Also I like the syntax of test writing, there's very little clutter and most of the lines in test file are part of test itself, so I don't lose too much time with syntax things when adding new tests. Usually all I need to type for new test is:

```
TEST( classname_testname ) {
  //here goes test itself
}
```

Sometimes I add things like SUITE(test_suite_name) { .. } or some #include in case of new file, and the most cumbersome thing I encountered was to force all tests to run also in release mode.

I mean.. it can not be any simpler? Can it?