

Re: Linked Lists debugger question

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.language/2008-09/msg00668.html>

- *From:* Robby <Robby@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Sun, 21 Sep 2008 20:11:01 -0700
-

Hello Giovanni, thanks for taking the time to reply to my C++ compiler orientation questions ! It is very nice of you.

"Note that Microsoft introduced some extensions to C++, developing C++/CLI. With C++/CLI extensions, you can use this RAD paradigma for building GUIs also with C++. "

Wow I didn't know this, I will have to keep an eye out for this!

So I guess that's that, Visual C++ is what I am sticking to.

What is the difference between the Visual C++ 2008 Express Edition free download version and the version that can be purchased. What I mean to ask is, with the Visual C++ 2008 Express Edition, am I missing out on some of the features or language capabilities such as stl, templates that would otherwise exist in the purchasable one. Also does the Visual C++ 2008 Express Edition free download have everything the VC++ 2006 did?

Na..... I don't think I will be doing any web based applications. And I think if I did, it would be very, very simple tasks like transferring a file from one PC to another via the internet and would probably end up subcontracting that out to someone who is much more familiar than I am with web designing.

Thanks!

—

Best regards
Robert

"Giovanni Dicanio" wrote:

"Robby" <Robby@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx> ha scritto nel messaggio news:F8E60C5B-B0E2-42FA-9CD1-C0136F586E28@xxxxxxxxxxxxxxxxxxxxxx

Re: Linked Lists debugger question

Hello fellows, thanks for your replies!

You are welcome.

I would like to ask you guys another question, Its been a long while I haven't used VC++ GUI and I would like get back into VC++ by progressively using its IDE to invoke small C programs that I would want to practice such as linked lists. Then slowly migrate into C++ so I can continue where I left off 2 years ago with Charles petzold's book.

However should I continue with the old version of VC++ or should I upgrade to some better and more recent version of VC++ like Visual studio perhaps.

I would suggest you to move to Visual C++ 2008 Express Edition. It is a free download:

<http://www.microsoft.com/express/vc/>

It has both a very good IDE and a very good C++ compiler. I don't know what you mean with "old version of VC++", but if you mean Visual C++ 6, then VC++2008 has a better and more standard compliant C++ compiler.

And so what's deal with C#, (I know I have already asked about this! but) is it still the language of choice for develloping PC based applications? and what about communications (like usb, rs232... is VC++ or C# better.

C# is a language that you could compare to Java: it is not native, it is "managed". C# code runs on a kind of virtual machine. Moreover C# has some features like garbage collector that is not present in C++ (but if you use modern C++ "tools" like STL containers like `std::vector`, and Boost/TR1 smart pointers like `shared_ptr`, you don't need a garbage collector in C# style). I think that lots of Windows PC successful applications are written in C++ (like Photoshop, Word, Excel, etc.). From what I heard and somehow seen, C# is widely used for web-based applications, like ASP.NET stuff. For communications with USB or RS232, I think that if you have proper libraries (in form of pure native C DLLs, or COM libraries, or .NET components) you can use whatever language, be it C++ or C# or VB.NET.

Re: Linked Lists debugger question

I *think* C# is close to VBA. You select a control, drag it and drop it onto a workspace, double click on it and you are prompted to start entering code

Yes, this is a RAD (Rapid Application Development) paradigm that I discovered with "classic" Visual Basic (i.e. Visual Basic 5 and 6). It is available also in C# and VB.NET if you use WinForms. You can download Visual C# 2008 Express Edition and experiment with WinForms. You can just drag some controls (like list-box, edit control, etc.) from a palette of controls and drop them into a form. Then you set some properties (like text or background colors) with the properties editor, and then you write some code to manage events (like button clicks).

Note that Microsoft introduced some extensions to C++, developing C++/CLI. With C++/CLI extensions, you can use this RAD paradigm for building GUIs also with C++.

but can someone tell me a little bit about the benefits of C# vs C++ native code.

The argument is very rich... In few words, I would say:

- 1) If you want top performances, use C++
- 2) If you have huge existing C++ code base, continue using C++ (it would make no sense to port to C#)
- 3) If you want portability to other platforms (e.g. Unix), you should use C++ instead of C# (there is a project called Mono that would like to port C# and .NET framework to Unix/Linux, but it seems to me not mature enough...)
- 4) If you want to build Windows system components like shell extensions (that are based on COM), you should use C++
- 5) If you want to write device drivers, you must use C++ (or better, I read that to write kernel drivers you need pure C – but I'm not expert in that fascinating field of device driver development).

I would suggest C# if you want to build web-applications using ASP.NET.

HTH,
Giovanni

Re: Linked Lists debugger question