Re: Is There Any Reason to Even Use VC++ Anymore?

Source:
http://www.tech−archive.net/Archive/DotNet/microsoft.public.dotnet.languages.vc/2005−12/msg00213.html

- *From*: "Peter Oliphant" <poliphant@xxxxxxxxxxxxxxx>
- *Date*: Wed, 7 Dec 2005 15:19:14 −0800

> Contracts between the calling function and the destination functions
> work quite nicely for me.

I've used this method to, and it's pretty good. It works fine if I'm the only one creating the code. However, getting a team to adhere to this rigorously is prone to error. And this is complicated due to what I think is a bug in the new VS C++.NET 2005 (Express). I reported it here:

http://lab.msdn.microsoft.com/productfeedback/viewfeedback.aspx?feedbackid=ef3c8d9e−75e2−4916−87ee−640a4de

Turns out if you use a reference (or value) parameter in a method that only accepts a pointer it will still compile! Thus, the convention of using a pointer if you own it and a reference if owned externally is broken (not your fault!) since both a pointer and a reference are accepted by a method asking for a pointer! I hope this isn't considered a 'feature' by MS...

> I tend to shun away from using lists of pointers, and generally don't
> have that big of a problem with it.

How do you deal with variable length lists? Imagine a picture editor. Would you set a limit as to how many pictures can be editted at a time, and keep all instances of them around regardless of how many you will actually need? It is probably better to keep a dynamic list of pointers to each of the pictures being editted. This extends to many situations where the application can't know ahead of time how many of something it will need, especially if this number is only limited by available memory or dictated by the end−user...

> I just go