

Re: KB192570 done right: Asynchronous sockets in threads

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.mfc/2007-04/msg00623.html>

- *From:* Joseph M. Newcomer <newcomer@xxxxxxxxxxxxx>
 - *Date:* Fri, 06 Apr 2007 10:25:02 -0400
-

With few exceptions, precompiled headers apply primarily to the very large windows.h files, tens of thousands to hundreds of thousands of lines of code. Unless your project is truly massive, the header files do not need to be precompiled because most header files are very small. Putting eight lines of declarations in stdafx.h buys no advantage for precompiled headers.

At least one of our MVPs is working on a project with something like a megaline of code, and precompiled headers that include project headers buys a lot of performance. In an example like the KB article, there is no real advantage.

People who have the experience to work on megaline projects have the experience to know when to apply generalizations and when to ignore them. For smaller projects, it makes no sense to think of stdafx.h as a place to put little lists of constants, because there is no benefit in precompilation. In fact, it actually works actively against you, because small local changes create massive rebuilds. The typical size project I work on is 50K–150K SLOC total, spread across 300–600 modules. Triggering a 400–module rebuild to change one line of header is counterproductive, and one such compilation swamps all the advantage that precompilation nominally gains. It is a matter of performance optimization. You don't preoptimize. Only when your header files begin to be a performance bottleneck in builds do you consider moving them into precompilation, and that's usually because on projects of the megaline range, massive parts of the project are stable for long periods of time. Also, if you create interfaces (instead of just using simplistic classes), the interfaces can be held stable while the implementations are free to change, and the implementation files do not become part of the precompiled headers. This is good project management.

So if you work on the premise that stdafx.h is reserved solely for fixed library files, you are doing the right thing for projects in the range I mentioned. When the project scales up, by another order of magnitude, you have a different set of issues.

The KB article encourages bad practice. Consider the strawberry shortcake analogy; you don't hire a hall and caterers to server four to sixteen servings. The overhead swamps the advantage.

joe

On Fri, 06 Apr 2007 11:01:35 GMT, MrAsm <mrasm@xxxxxxx> wrote:

Re: KB192570 done right: Asynchronous sockets in threads

On Fri, 06 Apr 2007 11:11:22 +0200, Norbert Unterberg
<nunterberg@xxxxxxxxxxxxxxxxxxxx> wrote:

Example:

You say that stdafx.h is reserved for standard header files. How would I know?

The MSDN says stdafx.h is for standard header files and project specific include files. In the past I thought the only use is to speed up compilation due to precompiled header files.

I think that if you put in stdafx.h your custom headers that are subject to change frequently, the advantage or precompilation vanishes.

You must put it in stdafx.h headers of things that are very stable and don't change, like Windows headers, or C++ STL headers, etc.

Question is:

Where are all the "official Best Practices" documents? Is all this specified by MS or in MSDN? Or is all your knowledge "common sense of an experienced developer"?

There are many helpful articles on your MVP tips site. Did you ever write a book about MFC/C++ best practices or considered writing one?

Norbert is right.

Joe: you really should think about writing such a book!

Thanks for your invaluable help

Yes: thank you very much!

MrAsm

Joseph M. Newcomer [MVP]

email: newcomer@xxxxxxxxxxxxxx

Web: <http://www.flounder.com>

MVP Tips: http://www.flounder.com/mvp_tips.htm

.