Subject: Pre-compiled headers support for MinGW
Posted by cc2000 on Mon, 15 Jan 2007 19:44:15 GMT

View Forum Message <> Reply to Message

Hi all.

I'm a new guest on your forum. So hello to all of you!

Do you consider to support GCH (PCH for GCC) for your build system? I got speedup 2 times (and more) with GCH enabled!
You may give me a hint where to start to integrate in your compile system. I'd like to test it bit...

regards,
steve

Subject: Re: Pre-compiled headers support for MinGW
Posted by cc2000 on Mon, 15 Jan 2007 19:51:55 GMT

View Forum Message <> Reply to Message

ok, first testing is done! notice, that i have my upp installed in "d:\upp" and disabled blitz and enabled "use shared libraries", full debug symbols. GCC build of course . MINGW/GCC is the one shipped with upp 701-dev1 (3.4.??).

here is all you have to do to get GCH-support for CtrlLib.h:

```
/d/upp/mingw/bin/c++ -c  -I"d:\upp\uppsrc" -I"d:\upp\mingw\include" -DflagGUI -DflagGCC
-DflagDEBUG -DflagSHARED -DflagDEBUG_FULL -DflagWIN32 \
 -g2  -fexceptions  -D_DEBUG -O0 -x c++ "d:\upp\uppsrc\CtrlLib\CtrlLib.h"
```

May you give me an hint, where to put the compile-option -Winvalid-pch that warns about unusable GCH-files and gives very helpful hints about possible problems (mostly mismatching compile flags or macros)

regards,
steve

Subject: Re: Pre-compiled headers support for MinGW
Posted by mirek on Mon, 15 Jan 2007 22:32:07 GMT

View Forum Message <> Reply to Message

cc2000 wrote on Mon, 15 January 2007 14:44Hi all.

I'm a new guest on your forum. So hello to all of you!

Do you consider to support GCH (PCH for GCC) for your build system? I got speedup 2 times (and more) with GCH enabled!

We got 4x times speedup with BLITZ (Well, it is more for compile phase alone, but linking has to be done too).

In fact, BLITZ solves the same problem, just without the need of compiler support and more effectively.

Mirek

Subject: Re: Pre-compiled headers support for MinGW
Posted by cc2000 on Tue, 16 Jan 2007 08:08:26 GMT

View Forum Message <> Reply to Message

Hello,
I heared about this but was not able to get this speedup in my compile process. Just let's give it a try and I will soon come up with some benchmarks.

BTW: blitz system is not that transparent to me - you may give more details to me.

Using compiler features is not that scary to me. MSVC is using it anyway (I suppose). As long as the suggested feature integrates well and is transparent on a certain level, it is acceptable to me.

regards,
stefan

Subject: Re: Pre-compiled headers support for MinGW
Posted by cc2000 on Tue, 16 Jan 2007 12:48:07 GMT

View Forum Message <> Reply to Message

hi,
here some benchmarking results from GCH usage.

Compiling reference\TreeCtrl\main.cpp
no blitz, no GCH => 13.7
no blitz, GCH => 3.3
blitz, no GCH => 13.4

I'm not sure, if my BLITZ works...
Could someone with a working setup proof the results?

thx,
steve

Subject: Re: Pre-compiled headers support for MinGW
Posted by zsolt on Tue, 16 Jan 2007 15:30:09 GMT
View Forum Message <> Reply to Message

check Build/Output Mode

Subject: Re: Pre-compiled headers support for MinGW
Posted by zsolt on Tue, 16 Jan 2007 15:31:26 GMT
View Forum Message <> Reply to Message

And Setup/Build methods as well.

Subject: Re: Pre-compiled headers support for MinGW
Posted by mirek on Tue, 16 Jan 2007 15:34:58 GMT
View Forum Message <> Reply to Message

BLITZ only has effect if you do large(r) rebuild. Like full rebuild of TheIDE.

I guess that recompiling a single file usually is hardly a problem.

Anyway, number posted look interesting. BLITZ is ortoghonal to precompiled headers, so perhaps we might give it a try...

Mirek

Subject: Re: Pre-compiled headers support for MinGW
Posted by cc2000 on Tue, 16 Jan 2007 16:49:23 GMT
View Forum Message <> Reply to Message

That sounds good! That feature is truly orthogonal and I think it pays off very well with litte effort.

For basic testing, I enabled verbose build output and modified the command by adding full path to mingw binaries and by removing the macro definitions for build time.

regards,
steve

Subject: Re: Pre-compiled headers support for MinGW
Posted by mirek on Tue, 16 Jan 2007 17:03:40 GMT
View Forum Message <> Reply to Message

Thinking about it, there is a catch however, but solvable.

Think is that quite often .h files are not files that get directly included into .cpp files (see e.g. CtrlLib).

We will need to detect which .h files are directly included from .cpp and precompile only those. But that should not be that hard, as system is already there to support header dependecies.

Mirek

Subject: Re: Pre-compiled headers support for MinGW
Posted by cc2000 on Tue, 16 Jan 2007 18:53:33 GMT
View Forum Message <> Reply to Message

The GCH docs say, that precompiled headers cannot be used in a include tree. thus only one pre-compiled header is valid (at the moment?) when compiling.
This is CtrlLib.h in many of your cpp files. I create the CtrlLib.h.gch in uppsrc/CtrlLib and the rebuild of most gui packages of upp ran much faster

Defining and precompiling a upp_everything.h may solve the problem of many include files, at least for users of upp, not upp framework itself.

Determining the prober list of header files required for the upp build process is part of the framework component design. I'm not on this topic.

current limitations (http://gcc.gnu.org/onlinedocs/gcc/Precompiled-Headers.html0
):
Only one precompiled header can be used in a particular compilation.
A precompiled header can't be used once the first C token is seen. You can have preprocessor directives before a precompiled header; you can even include a precompiled header from inside another header, so long as there are no C tokens before the #include.
The precompiled header file must be produced for the same language as the current compilation. You can't use a C precompiled header for a C++ compilation.
The precompiled header file must have been produced by the same compiler binary as the current compilation is using.
Any macros defined before the precompiled header is included must either be defined in the same way as when the precompiled header was generated, or must not affect the precompiled header, which usually means that they don't appear in the precompiled header at all.

Subject: Re: Pre-compiled headers support for MinGW
Posted by mirek on Sun, 28 Jan 2007 22:15:30 GMT
View Forum Message <> Reply to Message

"Only one precompiled header can be used in a particular compilation."

Well, would be interesting to know whether this is true for compilation of precompiled header itself....

If yes, we could use one precompiled header per package (placed in package out) and expect it to use precompiled header of "uses" package...

Mirek