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.