Subject: Re: U++ as .lib
Posted by mirek on Sun, 23 Sep 2007 09:04:43 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Sun, 23 September 2007 04:19
And I think it would be very useful to have an article on exactly how BLITZ works, what do you have to do to achieve it and why is it faster. Maybe other projects could benefit from it
(like at my work place where the slightest modification in a file requires a 1-2 minute rebuild).

Hm, in ToDo for eons.

OK, so quick BLITZ overview:

First, BLITZ processes packages (not the whole program) - each package can have a single BLITZ block.

Only .cpp including files with guards (#ifdef H... #define H) can qualify to be part of BLITZ block. Alternatively, you can force inclusion by #pragma BLITZ_APPROVE (also for header) or exlusion by #pragma BLITZ_PROHIBIT.

AND only files older than one hour qualify for BLITZ block. (Because you do not want files you work on to be in BLITZ block).

Also, the whole package can be excluded based on .upp settings.

Then files are scanned for any #defines, these are undefined at the end of file. BLITZ block is in fact a file generated into output directory that include all BLITZ approved files and gets compiled instead (it is named $blitz.cpp, you can check the output directory for details).

Now a dirty trick:

```
#ifdef  flagBLITZ
#define MK__s          MK__s_(COMBINE(BLITZ_INDEX__, __LINE__))
#else
#define MK__s          MK__s_(__LINE__)
#endif
```

Blitz block defines BLITZ_INDEX__ for each file, in order to give a library code chance to define unique static variable names...

OK, I believe that is all about the BLITZ magic:)

Mirek