Subject: Re: new operator declared twice in MinGW TDM
Posted by mirek on Fri, 10 Mar 2017 13:41:20 GMT
View Forum Message <> Reply to Message

koldo wrote on Fri, 10 March 2017 09:36Hello all

Compiling in Windows 7 with MinGW TDM 64 5.1.0 I have got this error:

```
C:\upp\uppsrc\Core\heap.cpp: In function 'void* operator new(size_t)':
C:\upp\uppsrc\Core\heap.cpp:245:53: error: declaration of 'void* operator new(size_t) throw (std::bad_alloc)' has a different exception specifier
 void *operator new(size_t size) throw(std::bad_alloc) { void *ptr = UPP::MemoryAlloc(size); return ptr; }
In file included from C:/Desarrollo/Packages/MinGW64/lib/gcc/x86_64-w64-mingw32/5.1.0/include/c++/functional:54:0,
                 from C:\upp\uppsrc\Core\Core.h:222,
                 from C:\upp\uppsrc\Core\heap.cpp:1:
C:/Packages/MinGW64/lib/gcc/x86_64-w64-mingw32/5.1.0/include/c++/new:111:7: error: from previous declaration 'void* operator new(std::size_t)'
 void* operator new(std::size_t) _GLIBCXX_THROW (std::bad_alloc)
```

It is like if new operators are declared twice.

It works with mingw64 supplied with U++. And in POSIX too. Perhaps TDM related bug?

Note that MIGW64 coming with U++ is 6.2.0, so what is the reason for using TDM?