Subject: U++ and MinGW 4.7 (-std=c++11) issues
Posted by Ptomaine on Sat, 21 Jul 2012 19:40:52 GMT
View Forum Message <> Reply to Message

First, I want to say that U++ compiles very well when used with MinGW 4.7 even when using c++11 compiler option for U++'s native libraries. But there are some issues pertaining to compilation.

To resolve these issues I've had to do the following changes:

1. Right before inclusion of the <shlobj.h> file, the CY definition must be undefined:

```
#undef CY
#include <shlobj.h>
```

I found two files that need to be fixes this way: App.cpp and Path.cpp

2. WINVER definition must be corrected in <Core.h>:
from
```
 #ifdef COMPILER_MINGW
  #define WINVER 0xFFFF
 #endif
```

to
```
 #if defined(COMPILER_MINGW) && !defined(WINVER)
  #define WINVER 0xFFFF
 #endif
```

3. The compiler asks to include <winsock2.h> before <windows.h>, so the definition block with <winsock2.h> need to be placed higher than the first <windows.h> inclusion in the <Core.h> file.

4. The line in the App.cpp source file must be chaged:
from
```
if (int(ShellExecuteW(NULL, L"open", wurl, NULL, L".", SW_SHOWDEFAULT)) <= 32) {
```

to
```
if (reinterpret_cast<std::ptrdiff_t>(ShellExecuteW(NULL, L"open", wurl, NULL, L".",
SW_SHOWDEFAULT)) <= 32) {
```

5. In the Defs.h header file, the code must be changed:
from
```
#ifdef PLATFORM_WIN32
inline bool IsNaN(double d)       { return _isnan(d); }
```

to
```
#ifdef PLATFORM_WIN32
inline bool IsNaN(double d)       { return std::isnan(d); }
```

So, if I make all these changes, the compilation goes smooth and fine even with c++11 compiler option.
