Subject: CreateProcessW doesn't compile with CLANG
Posted by forlano on Mon, 01 May 2023 13:09:44 GMT
View Forum Message <> Reply to Message

Hello,

my the app call the windows API

CreateProcessW()

```
#ifdef PLATFORM_WIN32
  #include <windows.h>
#endif

...

#ifdef PLATFORM_WIN32
   strcpy( a, commandline);
   wchar aa[1000];
   wcscpy(aa, commandline.ToWString());

   STARTUPINFOW si;  //STARTUPINFO si; //for char
   PROCESS_INFORMATION pi;
   ZeroMemory( &si, sizeof(si) );
   si.cb = sizeof(si);
   ZeroMemory( &pi, sizeof(pi) );

   CreateProcessW( NULL,   // No module name (use command line)
      aa,        // Command line for wchar
      NULL,          // Process handle not inheritable
      NULL,          // Thread handle not inheritable
      FALSE,         // Set handle inheritance to FALSE
      CREATE_NO_WINDOW, //0,             // No creation flags
      NULL,          // Use parent's environment block
      NULL,          // Use parent's starting directory
      &si,           // Pointer to STARTUPINFO structure
      &pi );         // Pointer to PROCESS_INFORMATION structure

   // Wait until child process exits.
   WaitForSingleObject( pi.hProcess, INFINITE );

   // Close process and thread handles.
   CloseHandle( pi.hProcess );
   CloseHandle( pi.hThread );
#else
...
```

Under MSVC19 it compiles and run without problem. Now I am using CLANG (no Microsoft SDK on my machine) and the compiler complains about wcscpy() and CreateProcessW():

C:\MyApps\Vega82\RoundSwiss.cpp (132): error: no matching function for call to 'wcscpy'
 (): w cscpy (aa, commandline.ToWString());
C:/upp/bin/clang/include/intrin.h (274): note: candidate function not viable: no known conversion from 'Upp::wchar[1000]' (aka 'unsigned long[1000]') to '
wchar_t *__restrict' for 1st argument
 (): __MACHINEIW64(wchar_t *__cdecl w cscpy(wchar_t * __restrict__ ,const wchar_t * __restrict__))
C:/upp/bin/clang/include/intrin.h (186): note: expanded from macro '__MACHINEIW64'
 (): #define __MACHINEIW64 _ _MACHINE
C:/upp/bin/clang/include/intrin.h (189): note: expanded from macro '__MACHINE'
 (): #define __MACHINE(X) X ;
C:\MyApps\Vega82\RoundSwiss.cpp (140): error: no matching function for call to 'CreateProcessW'
C:\MyApps\Vega82\RoundSwiss.cpp (222): error: no matching function for call to 'wcscpy'

Which include should I use? I tried to

#include <processthreadsapi.h>

and many others but without success.

Thanks,
Luigi

Subject: Re: CreateProcessW doesn't compile with CLANG
Posted by Oblivion on Mon, 01 May 2023 14:52:03 GMT
View Forum Message <> Reply to Message

Hello Luigi,

It seems that you've hit a problem that the move from 16 bit wchat to 32 bit wchar caused.

You need to convert the strings into WCHARS (16 bits) in order for CreateProcessW to work.

You can check my PtyProcess class (ln 53-90) to see how to -hopefully- solve this issue for both cmdline and env strings.

Link to the source code: https://github.com/ismail-yilmaz/Terminal/blob/300cb3311bf976a2f4a05fc3efefd9e7bc0ee439/PtyProcess/Win32Pty.cpp#L53

Best regards,
Oblivion

Subject: Re: CreateProcessW doesn't compile with CLANG
Posted by forlano on Mon, 01 May 2023 19:38:12 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Mon, 01 May 2023 16:52
You need to convert the strings into WCHARS (16 bits) in order for CreateProcessW to work.

You can check my PtyProcess class (ln 53-90) to see how to -hopefully- solve this issue for both cmdline and env strings.

Hello Oblivion,

thanks a lot for the answer. In fact

Vector<WCHAR> cmd = ToSystemCharsetW(commandline);
cmd.Add(0);

did the trick and converted the command line. I was not aware of this change, or I had not realised it could affect me.
Now I understand why on my production machine I had not this issue. There I had an U++ distro pre April 2022!

Thanks again!
Luigi
