How to disable FPO optimisation on x64 VC++ compil

Posted by Vineel Kumar Reddy Kovvuri - 17 Nov 2015 - 07:08

Hi,

Recently I started using Windbg(x64), To play with it, I have compiled(x64) a sample program using

cl.exe ls.c /Zi /Od /GS-

Interestingly the functions generated for the executable contain FPO optimization. Below is the snippet of the disassembly of a routine.

```
0:000> uf ls!GetDateString
ls!GetDateString:
00007ff7`459a6d00 mov     qword ptr ,rcx
00007ff7`459a6d05 sub     rsp,88h
00007ff7`459a6d0c mov     qword ptr ,0
00007ff7`459a6d15 mov     qword ptr ,0
00007ff7`459a6d1e mov     eax,dword ptr
00007ff7`459a6d24 and     eax,1
00007ff7`459a6d27 test    eax,eax
00007ff7`459a6d29 je      ls!GetDateString+0x3e (00007ff7`459a6d3e)
```

May I know how can I disable this FPO optimization on x64 VC++ compilers. Looking at the official documentation it seems we cannot disable it via /Oy- switch
https://msdn.microsoft.com/en-us/library/2kxx5t2c.aspx

/Oy enables frame-pointer omission and /Oy- disables omission. /Oy is available only in x86 compilers.

I am wondering if this flag is only available for x86, then how come the x64 windows routines like FileTimeToSystemTime has FPO disabled!

```
0:000> uf .
KERNELBASE!FileTimeToSystemTime:
00007ffc`a03ad120 mov     qword ptr ,rbx
00007ffc`a03ad125 push    rbp
00007ffc`a03ad126 mov     rbp,rsp
00007ffc`a03ad129 sub     rsp,40h
00007ffc`a03ad12d mov     rax,qword ptr
00007ffc`a03ad134 xor     rax,rsp
00007ffc`a03ad137 mov     qword ptr ,rax
00007ffc`a03ad13b mov     eax,dword ptr
00007ffc`a03ad13d mov     rbx,rdx
00007ffc`a03ad140 mov     dword ptr ,eax
00007ffc`a03ad143 mov     eax,dword ptr
00007ffc`a03ad146 mov     dword ptr ,eax
```

Thanks
Vineel

==========