Re: System−wide hooking, VB+ASM

Source: http://www.tech−archive.net/Archive/VB/microsoft.public.vb.winapi/2005−06/msg00234.html

- *From*: erewhon@xxxxxxxxxx (J French)
- *Date*: Mon, 6 Jun 2005 09:23:41 +0000 (UTC)

On Sun, 5 Jun 2005 12:41:59 −0700, "Sam Hobbs"
<samuel@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote:

>"J French" <erewhon@xxxxxxxxxx> wrote in message
>news:42a2bf6e.3410656@xxxxxxxxxxxxxxxxxx
>>
>>>I don't know if VB uses a DLL entry point (called DllMain, LibMain or
>>>whatever) but it probably does. We don't know what it is doing but it
>>>could
>>>be causing a problem somehow.
>>
>> My guess (and it is just a guess) is that all DLLs have this as a
>> standard entry point.

>That depends on what you mean by "standard". I think you mean that all DLLs
>have an entry point function, since it is a standard. Not all DLLs have a
>DLL entry point. I am not guessing, I know that the DLL entry point is not
>required.

>Perhaps you are thinking that when other languages, such as C++, are used to
>create a DLL, that the source code is generate with a DLL entry point
>function. When using VC 6 and a "Win32 Dynamic−Link Library" project is
>created, we have a choice of three options. One is to create a project
>without any source code generated, another is a project with a simple DLL,
>and another is a project for a DLL that exports symbols. The source code
>generated for the simple DLL does have a skeleton DllMain and that is all
>that is in the source code. The third type of project generates source code
>with a DllMain that also has a switch (select) statement and it has a sample
>class that exports a couple of functions.
>
>The DllMain can of course be deleted in a second from the generated source
>code if it is not needed.

I've left that unedited

It looks as if you are saying that a DLL need not have an entry point.

Somehow I doubt that, if an /explicit/ entry point is deleted then the
compiler (or maybe the linker) would supply an implicit entry point.

Or more likely the 'implicit' entry point looks for the 'explicit'
entry point and calls it if it is not Null.

Have a look at www.masm32.com
and download their programming kit

You'll find that the DLL example is actually an EXE and the last line
is:
End LibMain

Which is the standard way in which one determines the entry point of
an EXE or a COM

Remove that (eg: End ; LibMain) and the linker moans:

Creating library tstdll.lib and object tstdll.exp
LINK : error LNK2001: unresolved external symbol
__DllMainCRTStartup@12
tstdll.dll : fatal error LNK1120: 1 unresolved externals

.

- ***Follow−Ups***:
  - ♦ ***Re: System−wide hooking, VB+ASM***
    - ◊ *From:* Sam Hobbs

- ***References***:
  - ♦ ***Re: System−wide hooking, VB+ASM***
    - ◊ *From:* J French
  - ♦ ***Re: System−wide hooking, VB+ASM***
    - ◊ *From:* Sam Hobbs
  - ♦ ***Re: System−wide hooking, VB+ASM***
    - ◊ *From:* J French
  - ♦ ***Re: System−wide hooking, VB+ASM***
    - ◊ *From:* Karl E. Peterson
  - ♦ ***Re: System−wide hooking, VB+ASM***
    - ◊ *From:* J French
  - ♦ ***Re: System−wide hooking, VB+ASM***
    - ◊ *From:* Jonathan Wood
  - ♦ ***Re: System−wide hooking, VB+ASM***
    - ◊ *From:* J French
  - ♦ ***Re: System−wide hooking, VB+ASM***
    - ◊ *From:* Sam Hobbs
  - ♦ ***Re: System−wide hooking, VB+ASM***
    - ◊ *From:* J French
  - ♦ ***Re: System−wide hooking, VB+ASM***
    - ◊ *From:* Sam Hobbs

- Prev by Date: ***Mapping Network−Drive***
- Next by Date: ***Re: how to find app termination event?***
- Previous by thread: ***Re: System−wide hooking, VB+ASM***
- Next by thread: ***Re: System−wide hooking, VB+ASM***
- Index(es):
  - ***Date***
  - ***Thread***