

Re: SymGetTypeInfo and TI_GET_SYMNAME

Source:

<http://www.tech-archive.net/Archive/VisualStudio/microsoft.public.vsnnet.debugging/2004-03/0139.html>

From: Nicolas Hognon (*nicolash_at_virttools.com*)

Date: 03/26/04

Date: Fri, 26 Mar 2004 18:05:12 +0100

To: Oleg Starodumov <oleg_staro@hotmail.com>

Hello,

it is me again

Here is my new test

```
struct MyStruct
{
    int a;
    int b;
};

typedef int MyType;
typedef MyStruct MyStruct2;

bool Test(int pInt, char pChar, char* pStr)
{
    MyStruct mystruct;
    mystruct.a = 1;
    mystruct.b = 2;
    MyType mytype = 3;

    static const int staticconstint = 4;
    static int staticint = 5;
    const int constint = 6;
    const int* addr = &constint;

    MyStruct2 mystruct2;
    mystruct2.a = 7;
    mystruct2.b = 8;

    int array1[] = {11,12,13,14,15,16,17,18,19,20};
    short array2[] = {1,2,3,4,5,6,7,8,9,10};
    GetFunctionDebugInfo();
    return true;
}
```

Now i succed to have information about struct and typedef.

My last problem is pointer and array management.

How can i make the difference between:

- const int constint = 6;
- const int* addr = &constint;
- int array1[] = { 11,12,13,14,15,16,17,18,19,20};
- short array2[] = { 1,2,3,4,5,6,7,8,9,10};

all this variables type are basic. and this type is int (btInt is 6 from DIA SDK header file cvconst.h).

To make the difference between all type of int I can use the field Size from the SYMBOL_INFO structure. But size is 4 for addr, array1 and array2. I certainly mised something.

Some one got an idea ?

thanks

Oleg Starodumov wrote:

```
> Hi,
>
>
>>I alredy read the Under The Hood article called "Improved Error
>>Reporting with DBGHELP 5.1 APIs" but I cannot succed to get the name of
>>my parameters type when they are user defined (my own string class, ....).
>>
>
>
> Since I remember it was I who suggested you to use this article,
> I must apologize and say that the article is far from enough to complete
> the task. It shows how to work with the simplest types, but nothing more.
>
> To be able to work with all possible types, it is necessary to become familiar
> with symbol types (identified by "tags") and relationships between symbols.
>
> Since DbgHelp and DIA work with the same data and use the same concepts,
> DIA documentation can help here:
> http://msdn.microsoft.com/library/en-us/diasdk/html/vsoriDebugInterfaceAccessSDK.asp?frame=true
>
>
>>2. here is my callback function to test SymGetTypeInfo with TI_GET_SYMNAME
>>
>>static BOOL CALLBACK MyEnumSymbolsCallback(PSYMBOL_INFO symbol,
>> ULONG symbolSize,
>> PVOID userContext)
>>{
>> WCHAR* pwszTypeName = 0;
>> if (SymGetTypeInfo(ProcessHandle,
>> symbol->ModBase,
>> symbol->TypeIndex,
```

```
>> TI_GET_SYMNAME,
>> &pwszTypeName)) {
>> // ok I've got the name
>> LocalFree(pwszTypeName);
>> } else {
>> // FAILED :(((
>> }
>>}
>>
>>And it always failed :(((
>>
>
>
> Failed because "name" property (identified by TI_GET_SYMNAME) is not supported
> for the symbol identified by TypeIndex. It is possible to get the type name only if symbol's
> type is a class, structure, union, enum, typedef. For all other types, name is not available directly
> or at all (basic type – not available, use TI_GET_BASETYPE to get BasicType enumeration
> instead; pointer type – it is necessary to remember that it is a pointer and skip to the type
> the pointer points to; array – no name at all; function type – no name at all).
>
> So the basic procedure is the following:
> 1) Use TI_GET_SYMTAG with SYMBOL_INFO.TypeIndex to obtain the tag of the type symbol.
> 2) Depending on the tag, use the type symbol's properties to reconstruct the type declaration
> (not necessarily a name, think of array type as an example).
> 3) For some type symbols, it is necessary to analyze some additional symbols (may be children,
> obtained using TI_FINDCHILDREN, may be by an index obtained through a property).
>
>
>>and I succeed to get the following information
>>
>>TestCurrentFunctionInfo(p0=0x12fa58,
>> char p1="titi",
>> char p2="TITI",
>> char p3=c,
>> int p4=3,
>> int p5=1243756)
>>
>>but i want
>>void TestCurrentFunctionInfo(vkString p0="adresse of str",
>> const char* p1="titi",
>> char* p2="TITI",
>> char p3=c,
>> int p4=3,
>> int* p5="adresse of array")
>>
>
> The code from the article works only with the simplest types.
> With everything else (such as pointer types) it works incorrectly.
>
> Regards,
```

microsoft.public.vsnet.debugging: Re: SymGetTypeInfo and TI_GET_SYMNAME

> *Oleg*

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>

>