
Subject: Call External and Win32 API dll's Subroutine calls

Posted by [muswick](#) on Sun, 14 Jan 2001 02:59:50 GMT

[View Forum Message](#) <> [Reply to Message](#)

Maybe someone has solved these problems with calling standard Win32 api's without having to use a C-Wrapper routine. I search this newsgroup and I have a few examples but none that seems to address this problem directly.

There 3 examples below:

1: Calling a Win32 Subroutine with arguments

The problem is that no data is returned.

2: Calling a Win32 Function without arguments

No problem - This works (from this newsgroup)

3: Calling a Win32 Subrouting with arguments

The problem is an immediate crash.

Example 1

From Win32 API:

```
Public Declare Sub GetLocalTime Lib "kernel32" Alias "GetLocalTime"  
(lpSystemTime As SYSTEMTIME)
```

```
Public Type SYSTEMTIME
```

```
    wYear As Integer
```

```
    wMonth As Integer
```

```
    wDayOfWeek As Integer
```

```
    wDay As Integer
```

```
    wHour As Integer
```

```
    wMinute As Integer
```

```
    wSecond As Integer
```

```
    wMilliseconds As Integer
```

```
End Type
```

Here is the IDL code I have tried:

```
timedata = INTARR(8)
```

```
timedata2 = BYTARR(16)
```

```
SYSTEMTIME = { $
```

```
    wYear : 0, $
```

```
    wMonth : 0, $
```

```
    wDayOfWeek : 0, $
```

```
    wDay : 0, $
```

```
    wHour : 0, $
```

```
    wMinute : 0, $
```

```
    wSecond : 0, $
```

```
wMilliseconds : 0}
```

```
result = CALL_EXTERNAL('kernel32.dll','GetLocalTime',timedata)
help,result
print,timedata
```

```
result = CALL_EXTERNAL('kernel32.dll','GetLocalTime',timedata2)
help,result
print,timedata2
```

```
result = CALL_EXTERNAL('kernel32.dll','GetLocalTime',systemtime)
help,result
print,systemtime
```

```
RESULT      LONG    =      0
  0    0    0    0    0    0    0    0
RESULT      LONG    =      0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
RESULT      LONG    =      0
{  0    0    0    0    0    0    0    0  }
```

Example 2

Now a Win32 API function call with no arguments works:

From Win32 API:
Public Declare Function GetLogicalDrives Lib "kernel32" Alias
"GetLogicalDrives" () As Long

The IDL code:

```
drivemask=CALL_EXTERNAL('kernel32.dll','GetLogicalDrives')
help,drivemask
```

```
DRIVEMASK    LONG    =      125
```

Example 3

From Win32 API:
Public Declare Sub GetSystemInfo Lib "kernel32" Alias "GetSystemInfo"
(lpSystemInfo As SYSTEM_INFO)

```
Public Type SYSTEM_INFO
    dwOemID As Long
    dwPageSize As Long
```

```
lpMinimumApplicationAddress As Long
lpMaximumApplicationAddress As Long
dwActiveProcessorMask As Long
dwNumberOrfProcessors As Long
dwProcessorType As Long
dwAllocationGranularity As Long
dwReserved As Long
```

End Type

The IDL code:

```
sysinfo = {SYSTEM_INFO, $
    dwOemID : 0L, $
    dwPageSize : 0L, $
    lpMinimumApplicationAddress : 0L, $
    lpMaximumApplicationAddress : 0L, $
    dwActiveProcessorMask : 0L, $
    dwNumberOrfProcessors : 0L, $
    dwProcessorType : 0L, $
    dwAllocationGranularity : 0L, $
    dwReserved : 0L}

result = CALL_EXTERNAL('kernel32.dll','GetSystemInfo',sysinfo)
help,result
print,sysinfo
```

Causes an immediate Page Fault and crash of IDL.

I am most likely applying the arguments wrong. I am positive that I could get the above to work using a C-wrapper with (argc, argv) calling, but there are reasons that I would prefer not to go this route.

Any help would appreciated.

Gary Muswick
muswick@uhrad.com
216 844-7793

Sent via Deja.com
<http://www.deja.com/>