

Re: Help With PInvoke and SetupDiEnumDriverInfo/SP_DRVINFO_DATA

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2007-03/msg01317.html>

- *From:* "Rymfax" <cwalker@xxxxxxxxxxxxxxxx>
 - *Date:* 8 Mar 2007 14:26:10 -0800
-

On Mar 8, 3:50 pm, "Willy Denoyette [MVP]" <willy.denoye...@xxxxxxxxxx> wrote:

"Rymfax" <cwal...@xxxxxxxxxxxxxxxx> wrote in message

news:1173384460.483366.285820@xx

Willy,

I made that change and I still get the 1784L error. I'm wondering if the problem is not with SP_DRVINFO_DATA (cause I have written that thing every possible way I can think of), but with some other structure or DllImport I'm using further up the chain. I'm literally at my whits end on this. I have very similar code in C++ that works just fine. I'm posting a C# class that has everything I'm doing in it. I would be VERY grateful if you could take a look at it and see if you see a problem. As above, everything works fine until it gets to SetupDiEnumDriverInfo.

Thanks a ton!

```
---- CODE START ----
class GroupExample
{
    public const int CR_SUCCESS = (0x00000000);
    public const int DIGCF_PRESENT = (0x00000002);
}
```

Re: Help With PInvoke and SetupDiEnumDriverInfo/SP_DRVINFO_DATA

```
public const int DIGCF_ALLCLASSES = (0x00000004);
```

```
public const int ERROR_NO_MORE_ITEMS = 259;
```

```
public const int SPDIT_COMPATDRIVER = (0x00000002);
```

```
public const int SPDIT_CLASSDRIVER = (0x00000001);
```

```
[StructLayout(LayoutKind.Sequential, CharSet =  
CharSet.Unicode)]
```

```
public struct SP_DEVINFO_DATA
```

```
{  
    public int cbSize;  
    public Guid ClassGuid;  
    public IntPtr DevInst;  
    public int Reserved;  
}
```

```
[StructLayout(LayoutKind.Sequential, CharSet =  
CharSet.Unicode)]
```

```
public struct SP_DRVINSTALL_PARAMS
```

```
{  
    public int cbSize;  
    public int Rank;  
    public int Flags;  
    public int PrivateData;  
    public int Reserved;  
}
```

```
[StructLayout(LayoutKind.Sequential, CharSet =  
CharSet.Unicode)]
```

```
public struct SP_DEVINSTALL_PARAMS
```

```
{  
    public int cbSize;  
    public int Flags;  
    public int FlagsEx;  
    public IntPtr HwndParent;  
    public IntPtr InstallMsgHandler;  
    public IntPtr InstallMsgHandlerContext;  
    public IntPtr FileQueue;  
    public ulong CallInstallReserved;  
    public int Reserved;  
    [MarshalAs(UnmanagedType.ByValTStr, SizeConst = 260)]  
    public string DriverPath;
```

```
}
```

```
[StructLayout(LayoutKind.Sequential, CharSet =  
CharSet.Unicode)]  
public struct SP_DRVINFO_DATA  
{  
public int cbSize;  
public int DriverType;  
public ulong Reserved;  
[MarshalAs(UnmanagedType.ByValTStr, SizeConst = 256)]  
public string Description;  
[MarshalAs(UnmanagedType.ByValTStr, SizeConst = 256)]  
public string MfgName;  
[MarshalAs(UnmanagedType.ByValTStr, SizeConst = 256)]  
public string ProviderName;  
public System.Runtime.InteropServices.ComTypes.FILETIME  
DriverDate;  
public ulong DriverVersion;  
}
```

```
[DllImport("setupapi.dll", SetLastError = true, CharSet =  
CharSet.Unicode)]  
public static extern Boolean  
SetupDiDestroyDeviceInfoList(IntPtr DeviceInfoSet);
```

```
[DllImport("setupapi.dll", SetLastError = true, CharSet =  
CharSet.Unicode)]  
public static extern IntPtr SetupDiGetClassDevs(ref Guid  
ClassGuid, IntPtr Enumerator,  
IntPtr hwndParent, int Flags);
```

```
[DllImport("setupapi.dll", SetLastError = true, CharSet =  
CharSet.Unicode)]  
public static extern bool SetupDiEnumDeviceInfo(IntPtr  
DeviceInfoSet, int MemberIndex,  
ref SP_DEVINFO_DATA DeviceInfoData);
```

```
[DllImport("cfgmgr32.dll", SetLastError = true, CharSet =  
CharSet.Unicode)]
```

Re: Help With PInvoke and SetupDiEnumDriverInfo/SP_DRVINFO_DATA

```
private static extern int CM_Get_DevNode_Status(ref ulong
status, ref ulong probNum,
IntPtr devInst, int Flag);
```

```
[DllImport("setupapi.dll", SetLastError = true, CharSet =
CharSet.Unicode)]
public static extern bool
SetupDiGetDeviceRegistryProperty(IntPtr DeviceInfoSet,
ref SP_DEVINFO_DATA DeviceInfoData, int Property, ref int
PropertyRegDataType,
StringBuilder PropertyBuffer, int PropertyBufferSize, ref
int RequiredSize);
```

```
[DllImport("setupapi.dll", SetLastError = true, CharSet =
CharSet.Unicode)]
public static extern bool SetupDiGetDeviceInstallParams(IntPtr
DeviceInfoSet,
ref SP_DEVINFO_DATA DeviceInfoData, ref
SP_DEVINSTALL_PARAMS DeviceInstallParams);
```

```
[DllImport("setupapi.dll", SetLastError = true, CharSet =
CharSet.Unicode)]
private static extern bool SetupDiDestroyDriverInfoList(IntPtr
DeviceInfoSet,
ref SP_DEVINFO_DATA DeviceInfoData, int DriverType);
```

```
[DllImport("setupapi.dll", SetLastError = true, CharSet =
CharSet.Unicode)]
private static extern bool SetupDiBuildDriverInfoList(IntPtr
DeviceInfoSet,
ref SP_DEVINFO_DATA DeviceInfoData, int DriverType);
```

```
[DllImport("setupapi.dll", SetLastError = true, CharSet =
CharSet.Unicode)]
private static extern bool SetupDiEnumDriverInfo(IntPtr
DeviceInfoSet,
ref SP_DEVINFO_DATA DeviceInfoData, int DriverType, int
MemberIndex,
ref SP_DRVINFO_DATA DriverInfoData);
```

Re: Help With PInvoke and SetupDiEnumDriverInfo/SP_DRVINFO_DATA

```
public void RunExample()
{
    Guid newG = Guid.Empty;
    IntPtr NewDeviceInfoSet = SetupDiGetClassDevs(ref newG,
    IntPtr.Zero, IntPtr.Zero,
    DIGCF_ALLCLASSES);
    SP_DEVINFO_DATA DeviceInfoData = new SP_DEVINFO_DATA();
    DeviceInfoData.cbSize =
    Marshal.SizeOf(typeof(SP_DEVINFO_DATA));
    int deviceCounter = 0;
    bool devInfoRetVal =
    SetupDiEnumDeviceInfo(NewDeviceInfoSet, deviceCounter, ref
    DeviceInfoData);
    int devInfoErr = Marshal.GetLastWin32Error();
    while (devInfoRetVal && devInfoErr != ERROR_NO_MORE_ITEMS
    && deviceCounter < 3) // < 3 is to keep the loop short for testing
    {
        ulong status = 0;
        ulong problem = 0;
        if (CM_Get_DevNode_Status(ref status, ref problem,
        DeviceInfoData.DevInst, 0) == CR_SUCCESS)
        {
            StringBuilder sbDesc = new StringBuilder("");
            int MAX_LEN = 512;
            int propRegDataType = 0;
            sbDesc.Capacity = MAX_LEN;
            int reqSize = 0;
            SetupDiGetDeviceRegistryProperty(NewDeviceInfoSet,
            ref DeviceInfoData,
            (int)Setup.SPDRPCodes.SPDRP_FRIENDLYNAME, ref
            propRegDataType, sbDesc,
            MAX_LEN, ref reqSize);
            if (sbDesc.ToString() == "")
            {
                SetupDiGetDeviceRegistryProperty(NewDeviceInfoSet, ref DeviceInfoData,
                (int)Setup.SPDRPCodes.SPDRP_DEVICEDESC,
                ref propRegDataType, sbDesc,
                MAX_LEN, ref reqSize);
            }
            if (sbDesc.ToString() == "")
            sbDesc.Append("Unknown Description");
            StringBuilder sbHWID = new StringBuilder("");
            sbHWID.Capacity = MAX_LEN;
            SetupDiGetDeviceRegistryProperty(NewDeviceInfoSet,
            ref DeviceInfoData,
            (int)Setup.SPDRPCodes.SPDRP_HARDWAREID, ref
            propRegDataType, sbHWID,
            MAX_LEN, ref reqSize);
```

Re: Help With PInvoke and SetupDiEnumDriverInfo/SP_DRVINFO_DATA

```
ScanResults scanResult = new
ScanResults(sbHWID.ToString(), sbDesc.ToString());
Console.WriteLine("Device Detected: {0} - {1}",
sbDesc.ToString(), sbHWID.ToString());

SP_DEVINSTALL_PARAMS deviceInstallParams = new
SP_DEVINSTALL_PARAMS();
deviceInstallParams.cbSize =
Marshal.SizeOf(typeof(SP_DEVINSTALL_PARAMS));
SetupDiGetDeviceInstallParams(NewDeviceInfoSet,
ref DeviceInfoData,
ref deviceInstallParams);
deviceInstallParams.DriverPath = @"C:
\MyDriverPath";
if (!SetupDiBuildDriverInfoList(NewDeviceInfoSet,
ref DeviceInfoData, SPDIT_COMPATDRIVER))
{
int eCode = Marshal.GetLastWin32Error();
Console.WriteLine("\tError Building Driver
Info List. Code: {0}", eCode);
}
int memIndex = 0;
int drvErr = 0;
SP_DRVINFO_DATA drvData = new SP_DRVINFO_DATA();
drvData.cbSize =
Marshal.SizeOf(typeof(SP_DRVINFO_DATA));
bool enumResult =
SetupDiEnumDriverInfo(NewDeviceInfoSet, ref DeviceInfoData,
SPDIT_COMPATDRIVER,
memIndex, ref drvData);
if (!enumResult)
{
drvErr = Marshal.GetLastWin32Error();
Console.WriteLine("\tError Enumerating Driver
Info. Code: {0}", drvErr);
}
while (enumResult && drvErr !=
ERROR_NO_MORE_ITEMS)
{
memIndex++;

}
SetupDiDestroyDriverInfoList(NewDeviceInfoSet, ref
DeviceInfoData, SPDIT_COMPATDRIVER);
}
deviceCounter++;
devInfoRetVal =
```

Re: Help With PInvoke and SetupDiEnumDriverInfo/SP_DRVINFO_DATA

```
SetupDiEnumDeviceInfo(NewDeviceInfoSet, deviceCounter, ref
DeviceInfoData);
devInfoErr = Marshal.GetLastWin32Error();
}
SetupDiDestroyDeviceInfoList(NewDeviceInfoSet);
}
}
--- CODE END ---
```

Your struct should look like this (partly):

```
[StructLayout(LayoutKind.Sequential, Pack = 4, CharSet = CharSet.Unicode)] <-----1
public struct SP_DRVINFO_DATA
{
public int cbSize;
public int DriverType;
public IntPtr Reserved;
...
}
```

read more »– Hide quoted text –

– Show quoted text –

Willy, You are a GOD among men!!!!

Do you think you could explain to me what the Pack = 4 does and why I need it for this structure but not any of my others?

Thanks! Would you like my first born? He's yours! :)

.