



Native Calls

Aufruf einer Funktion aus einer WinDLL



Signatur der aufzurufenden Win32-Funktion

```
int MessageBox (HWND hWnd, LPTSTR lpText, LPTSTR lpCaption, UINT uType);
```

Aufruf aus C#

```
using System;  
using System.Runtime.InteropServices;
```

```
class Test {
```

```
[DllImport("user32.dll")]
```

```
static extern int MessageBox(int hWnd, string text, string caption, int type); // kein Rumpf
```

```
static void Main() {
```

```
    int res = MessageBox(0, "Isn't that cool?", "", 1);
```

```
    Console.WriteLine("res = " + res);
```

```
}
```

```
}
```



DllImport-Attribut



```
[DllImport(dll.name
    [, EntryPoint=function-name]
    [, CharSet=charset-enum]           // für String-Marshaling
    [, ExactSpelling= (true|false)]    // Name entsprechend CharSet modifizieren?
    [, SetLastError= (true|false)]    // true: liefert Win32-Fehlerinfo
    [, CallingConvention=callconv-enum]
)]
```

charset-enum: CharSet.Auto (default), CharSet.Ansi, CharSet.Unicode, ...

callconv-enum: CallingConvention.StdCall (default), CallingConvention.FastCall, ...

Parameter-Marshaling



Standardtypen von C# werden automatisch auf richtige Win32-Typen abgebildet.

Standard-Abbildung kann folgendermaßen überschrieben werden

```
using System.Runtime.InteropServices;
```

```
...
```

```
[DllImport("...")]  
static extern void Foo (  
    [MarshalAs(UnmanagedType.LPStr)] string s,  
    int x  
);
```

ANSI-String aus Bytes

LPStr ANSI-String aus einzelnen Bytes

LPWStr Unicode-String

LPTStr Default: Windows NT/2000 => Unicode, Windows 98 => ANSI

Übergabe von Klassen und Structs



```
using System.Runtime.InteropServices;
```

```
[StructLayout(LayoutKind.Sequential)] // Members werden sequentiell angelegt  
class Time { // und dicht gepackt  
    public ushort year;  
    public ushort month;  
    ...  
}
```

```
[StructLayout(LayoutKind.Explicit)] // Members werden an spezifizierte  
struct S { // Offsets gepackt  
    [FieldOffset(0)] ushort x;  
    [FieldOffset(2)] int y;  
}
```

```
[DllImport("...")] static extern void Foo(Time t);
```

```
[DllImport("...")] static extern void Foo1([MarshalAs(UnmanagedType.LPStruct)] Time t);
```

```
[DllImport("...")] static extern void Bar(S s);
```

```
Foo(new Time());
```

```
Bar(new S());
```

StructLayout-Attribut



```
[StructLayout ( layout-enum // Sequential | Explicit | Auto
                [, Pack=packing-size] // 0, 1, 2, 4, 8, 16, ...
                [, CharSet=charset-enum] // Ansi | Unicode | Auto
            )]
```