

p/invoke gotcha: C++ bool is not Win32 BOOLEAN is not UnmanagedType.Bool

devblogs.microsoft.com/oldnewthing/20150817-00

August 17, 2015



Raymond Chen

Welcome to CLR Week. I hope you enjoy your stay.

A customer reported that their p/invoke was not working.

We aren't getting the proper return codes from the `AuditSetSystemPolicy`. When the call succeeds, the return code is 1, as expected. But in our tests, when we force the call to fail (insufficient access), the return code is not zero. Instead, the return code is some value of the form `0xFFxxxxxx`, where the x's vary, but the high byte is always `0xFF`.

For reference, the DllImport declaration we are using is

```
[DllImport("advapi32.dll", SetLastError=true)]
public static extern UInt32 AuditSetSystemPolicy(
    IntPtr pAuditPolicy,
    UInt32 policyCount);
```

The corresponding Win32 declaration is

```
BOOLEAN WINAPI AuditSetSystemPolicy(
    _In_ PCAUDIT_POLICY_INFORMATION pAuditPolicy,
    _In_ ULONG PolicyCount
);
```

Alas, the customer fell into one of the common gotchas when writing p/invoke: They confused `BOOLEAN` and `BOOL`.

`BOOL` is a 32-bit integer, whereas `BOOLEAN` is an 8-bit integer.

Since they were marshaling the return code as a `UInt32`, they were getting the byte returned by the function, plus three bonus uninitialized garbage bytes. If they studied more closely, they would have found that the erroneous return codes were all of the form

`0xFFxxxx00` where the bottom 8 bits are all zero. That's because the bottom 8 bits are the actual value; the rest are garbage.

The correct declaration is to use `UnmanagedType.U1` aka `byte` rather than `UnmanagedType.U4` aka `UInt32`.

```
[DllImport("advapi32.dll", SetLastError=true)]
public static extern byte AuditSetSystemPolicy(
    IntPtr pAuditPolicy,
    UInt32 policyCount);
```

The customer confirmed that switching to `UnmanagedType.U1` fixed the problem.

Raymond Chen

Follow

