

RE: passing enum value as an argument

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.general/2005-07/msg00651.html>

- *From:* Jon Skeet [C# MVP] <skeet@xxxxxxxx>
 - *Date:* Wed, 13 Jul 2005 18:32:32 +0100
-

Madestro <me_no_like_spam_juanDOTromero@bowneDOTcom> wrote:

> I am not sure what language you are coding in

C#.

> but at least in VB.NET you are incorrect.

Not with option strict on. If you're programming with option strict off, that's a different matter – you've given up any vestige of compile time type safety already.

> Here are the TRUE results:

>

- > TakeFirstEnum(Foo) --> FAILS
- > TakeFirstEnum(FirstEnum.Foo) --> SUCCEEDS
- > TakeSecondEnum(Fred) --> FAILS
- > TakeSecondEnum(SecondEnum.Fred) --> SUCCEEDS
- >
- > TakeFirstEnum(Fred) --> FAILS
- > TakeFirstEnum(SecondEnum.Fred) --> SUCCEEDS
- > TakeSecondEnum(Foo) --> FAILS
- > TakeSecondEnum(FirstEnum.Foo) --> SUCCEEDS

The failures are all because the language doesn't allow what's being proposed, of course.

However, I don't get the same results as you. Here's a short but complete program to demonstrate:

```
Option Strict On
```

```
Imports System
```

```
Public Enum FirstEnum
```

```
Foo
```

```
Bar
```

```
Baz
```

```
End Enum
```

RE: passing enum value as an argument

```
Public Enum SecondEnum
Fred
George
Harry
End Enum
```

```
Public Class Test
```

```
Shared Sub TakeFirstEnum (value As FirstEnum)
Console.WriteLine (value)
End Sub
```

```
Shared Sub TakeSecondEnum (value As SecondEnum)
Console.WriteLine (value)
End Sub
```

```
Shared Sub Main()
TakeFirstEnum(FirstEnum.Foo)
TakeFirstEnum(SecondEnum.Fred)
TakeSecondEnum(FirstEnum.Foo)
TakeSecondEnum(SecondEnum.Fred)
End Sub
End Class
```

Compiling the above gives:

```
c:\test\Test.vb(30) : error BC30512: Option Strict On disallows
implicit conversions from 'SecondEnum' to 'FirstEnum'.
```

```
TakeFirstEnum(SecondEnum.Fred)
```

```
~~~~~
```

```
c:\test\Test.vb(31) : error BC30512: Option Strict On disallows
implicit conversions from 'FirstEnum' to 'SecondEnum'.
```

```
TakeSecondEnum(FirstEnum.Foo)
```

```
~~~~~
```

Turning option strict off, it compiles fine, but as I said, if you're running with option strict off I have very little sympathy with any type safety issues you may run into :)

> Like I said before, ultimately, the enum evaluates to a constant

Of the appropriate type though – the type of the expression is known to the compiler.

> so there

> is really no "type safety" here, except for the type of the constant and the

> range of valid values the enum provides.

RE: passing enum value as an argument

Actually, the range of valid values the enum provides **isn't** part of the safety – you can cast any value from the underlying type.

- > Because of this, I could call the first function with a value from
- > the second Enum so long as it evaluates to a value within the range
- > of the first Enum.

That's **entirely** wrong. With option strict on you can't call it at all, and with option strict you can call it whether it's in the range or not.

- > What I cannot do however is call one of the values of the Enum
- > without fully qualifying it because once again, like I said before)
- > you would run into ambiguity problems.

Well, you can't do it at the moment because the language specification says you can't. There's nothing to stop the language specification from changing to allow you to do it where there was no ambiguity – and that change **wouldn't** lose the whole point of enums.

- > It may not be the case in the language you are using, but it certainly is
- > the case with VB.NET. Try it, you will see.

I tried it, and found what you asserted to be wrong – did you actually try it, or were you making assumptions?

—

Jon Skeet – <skeet@xxxxxxxx>

<http://www.pobox.com/~skeet>

If replying to the group, please do not mail me too

.

-
- *Follow-Ups:*
 - ◆ **RE: passing enum value as an argument**
 - ◇ From: Madestro
 - *References:*
 - ◆ **passing enum value as an argument**
 - ◇ From: Glenn Venzke
 - ◆ **RE: passing enum value as an argument**
 - ◇ From: Madestro
 - ◆ **RE: passing enum value as an argument**
 - ◇ From: Jon Skeet [C# MVP]
 - ◆ **RE: passing enum value as an argument**
 - ◇ From: Madestro
 - ◆ **RE: passing enum value as an argument**
 - ◇ From: Jon Skeet [C# MVP]
 - ◆ **RE: passing enum value as an argument**
 - ◇ From: Madestro

RE: passing enum value as an argument

◆ **RE: passing enum value as an argument**

◇ From: Jon Skeet [C# MVP]

◆ **RE: passing enum value as an argument**

◇ From: Madestro

- Prev by Date: **Re: Read binary value from registry (Convert to date?)**
- Next by Date: **Public variable QueryString issue**
- Previous by thread: **Re: passing enum value as an argument**
- Next by thread: **RE: passing enum value as an argument**
- Index(es):
 - ◆ **Date**
 - ◆ **Thread**