

1 System.Reflection.CallingConventions Enum

```
2 [ILAsm]  
3 .class public serializable sealed System.Reflection.CallingConventions  
4 extends System.Enum  
  
5 [C#]  
6 public enum CallingConventions
```

7 Assembly Info:

- 8 • *Name:* mscorlib
- 9 • *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- 10 • *Version:* 4.0.0.0
- 11 • *Attributes:*
 - 12 ○ CLSCompliantAttribute(true)

13 Type Attributes:

- 14 • System.Flags

15 Summary

16 Defines the valid calling conventions for a method.

17 Inherits From: System.Enum

18

19 **Library:** RuntimeInfrastructure

20

21 Description

22 The native calling convention is the set of rules governing the order and layout of
23 arguments passed to compiled methods. It also governs how to pass the return value,
24 what registers to use for arguments, and whether the called or the calling method
25 removes arguments from the stack.

26

1 CallingConventions.Any Field

```
2 [ILAsm]  
3 .field public static literal valuetype  
4 System.Reflection.CallingConventions Any = int32(0x00000003)  
5 [C#]  
6 Any
```

7 Summary

8 Specifies that either the Standard or the VarArgs calling convention may be used.

9

1 CallingConventions.ExplicitThis Field

```
2 [ILAsm]  
3 .field public static literal valuetype  
4 System.Reflection.CallingConventions ExplicitThis = int32(0x00000040)  
5 [C#]  
6 ExplicitThis
```

7 Summary

8 Specifies that the signature is a function-pointer signature, representing a call to an
9 instance or virtual method (not a static method). If ExplicitThis is set, HasThis must
10 also be set. The first argument passed to the called method is still a this pointer, but
11 the type of the first argument is now unknown. Therefore, a token that describes the
12 type (or class) of the this pointer is explicitly stored into its metadata signature.

13

1 CallingConventions.HasThis Field

```
2 [ILAsm]  
3 .field public static literal valuetype  
4 System.Reflection.CallingConventions HasThis = int32(0x00000020)  
5 [C#]  
6 HasThis
```

7 Summary

8 Specifies an instance or virtual method (not a static method). At run-time, the called
9 method is passed a pointer to the target object as its first argument (the `this` pointer).
10 Unless `ExplicitThis` is also set, the signature stored in metadata does not include the
11 type of this first argument, because the method is known and its owner class can be
12 discovered from metadata.

13

1 CallingConventions.Standard Field

```
2 [ILAsm]  
3 .field public static literal valuetype  
4 System.Reflection.CallingConventions Standard = int32(0x00000001)  
5 [C#]  
6 Standard
```

7 Summary

8 Specifies the default calling convention as determined by the common language
9 infrastructure. Use this calling convention for static methods. For instance or virtual
10 methods use `HasThis`.

11

1 CallingConventions.VarArgs Field

```
2 [ILAsm]  
3 .field public static literal valuetype  
4 System.Reflection.CallingConventions VarArgs = int32(0x00000002)  
5 [C#]  
6 VarArgs
```

7 Summary

8 Specifies the calling convention for methods with variable arguments.

9