

1 System.Reflection.ConstructorInfo

2 Structure

3
4

```
5 [ILASM]  
6 .class public abstract serializable ConstructorInfo extends  
7 System.Reflection.MethodBase  
  
8 [C#]  
9 public abstract class ConstructorInfo: MethodBase
```

10 Assembly Info:

- 11 • *Name*: mscorlib
- 12 • *Public Key*: [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- 13 • *Version*: 1.0.x.x
- 14 • *Attributes*:
 - 15 ○ CLSCompliantAttribute(true)

16 Summary

17

18 Provides access to constructor metadata.

19 Inherits From: System.Reflection.MethodBase

20

21 **Library:** Reflection

22

23 **Thread Safety:** This type is safe for multithreaded operations.

24

25

26

1 ConstructorInfo() Constructor

2 [ILASM]
3 family rtspecialname specialname instance void .ctor()

4 [C#]
5 protected ConstructorInfo()

6 Summary

7 Constructs a new instance of the
8 **System.Reflection.ConstructorInfo** class.

9

1 ConstructorInfo.ConstructorName Field

```
2 [ILASM]  
3 .field public static initOnly string ConstructorName  
4 [C#]  
5 public static readonly string ConstructorName
```

6 Summary

7 A string containing the name of an object constructor as it is stored in
8 metadata.

9 Description

10 This field is read-only.

11
12 This field is a **System.String** that contains the value ".ctor". An object
13 constructor will be named with this field if and only if it is not a type
14 initializer.

15
16 [Note: For more information on type initializers, see
17 **System.Reflection.ConstructorInfo.TypeConstructorName**.

18
19 For more information on object constructors, see Partition II of the CLI
20 Specification.]

21

1 ConstructorInfo.TypeConstructorName 2 Field

3 [ILASM]
4 `.field public static initOnly string TypeConstructorName`

5 [C#]
6 `public static readonly string TypeConstructorName`

7 Summary

8 A string containing the name of a type initializer as it is stored in
9 metadata.

10 Description

11 This field is read-only.

12
13 This field is a **System.String** that contains the value ".cctor".

14
15 [Note: A type initializer can be applied to all types. It allows the type
16 to perform any initialization required before any members declared
17 within the type are accessed. Type initializers accept no parameters
18 and always have a return type of void. A type constructor only has
19 access to a type's static fields and its usual purpose is to initialize
20 those fields. A type's constructor is guaranteed to run before any
21 instance of the type is created and before any static field or method of
22 the type is referenced.

23
24 Many languages (including C#) automatically generate type
25 constructors for all implementer-defined types. However, some
26 languages require that type constructors be explicitly implemented.

27
28 For more information on type initializers, see Partition II of the CLI
29 Specification.]

30

1 **ConstructorInfo.Invoke(System.Reflection**
2 **.BindingFlags, System.Reflection.Binder,**
3 **System.Object[],**
4 **System.Globalization.CultureInfo) Method**

```
5 [ILASM]  
6 .method public hidebysig virtual abstract object  
7 Invoke(valuetype System.Reflection.BindingFlags invokeAttr,  
8 class System.Reflection.Binder binder, class  
9 System.Object[] parameters, class  
10 System.Globalization.CultureInfo culture)
```

```
11 [C#]  
12 public abstract object Invoke(BindingFlags invokeAttr,  
13 Binder binder, object[] parameters, CultureInfo culture)
```

14 **Summary**

15 Invokes the constructor reflected by the current instance using the
16 specified arguments, under the constraints of the specified
17 **System.Reflection.Binder**.

18 **Parameters**

| Parameter | Description |
|-------------------|--|
| <i>invokeAttr</i> | A System.Reflection.BindingFlags value that controls the binding process. |
| <i>binder</i> | A System.Reflection.Binder that defines a set of properties and enables the binding, coercion of argument types, and invocation of members using reflection. If <i>binder</i> is null , then the default binder is used. |
| <i>parameters</i> | An array of objects that match the number, order and type of the parameters for the constructor reflected by the current instance. If the constructor reflected by the current instance takes no parameters, specify either an array with zero elements or null . [Note: Any object in this array that is not explicitly initialized with a value will contain the default value for that object type. For reference-type elements, this value is null . For value-type elements, this value is 0, 0.0, or false , depending on the specific element type.] |
| <i>culture</i> | The only defined value for this parameter is null . |

21
22 **Return Value**
23

1 An instance of the class that declared the constructor reflected by the
2 current instance.

3 Behaviors

4 Before calling the constructor, this method ensures that the caller has
5 access permission and that the parameters are of the correct number,
6 order and type.

7 Exceptions

8
9

| Exception | Condition |
|--|---|
| System.ArgumentException | The types of the elements of <i>parameters</i> do not match the types of the parameters accepted by the constructor reflected by the current instance, under the constraints of <i>binder</i> . |
| System.MethodAccessException | The caller does not have the required permissions. |
| System.Reflection.TargetInvocationException | The constructor reflected by the current instance threw an exception. |
| System.Reflection.TargetParameterCountException | <i>parameters.Length</i> does not equal the number of parameters required by the contract of the constructor reflected by the current instance. |

10
11
12
13

Permissions

| Permission | Description |
|---|---|
| System.Security.Permissions.ReflectionPermission | Requires permission to invoke non-public members of loaded assemblies. See System.Security.Permissions.ReflectionPermissionFlag.MemberAccess . |

14
15
16

1 ConstructorInfo.Invoke(System.Object[]) 2 Method

```
3 [ILASM]  
4 .method public hidebysig instance object Invoke(class  
5 System.Object[] parameters)  
  
6 [C#]  
7 public object Invoke(object[] parameters)
```

8 Summary

9 Invokes the constructor reflected by the current instance using the
10 specified parameters.

11 Parameters

12
13

| Parameter | Description |
|-------------------|--|
| <i>parameters</i> | An array of objects that match the number, order and type of the parameters for the constructor reflected by the current instance. If the constructor reflected by the current instance takes no parameters, specify either an array with zero elements or null . [Note: Any object in this array that is not explicitly initialized with a value will contain the default value for that object type. For reference-type elements, this value is null . For value-type elements, this value is 0, 0.0, or false , depending on the specific element type.] |

14
15
16

Return Value

17 An instance of the class that declared the constructor reflected by the
18 current instance.

19 Exceptions

20
21

| Exception | Condition |
|-------------------------------------|---|
| System.ArgumentException | The types of the elements of <i>parameters</i> do not match the types of the parameters accepted by the constructor reflected by the current instance, under the constraints of <i>binder</i> . |
| System.MethodAccessException | The caller does not have the required permissions. |
| Svstem.Reflection. | The constructor reflected by the current |

| | |
|--|---|
| TargetInvocationException | instance threw an exception. |
| System.Reflection.TargetParameterCountException | <i>parameters.Length</i> does not equal the number of parameters required by the contract of the constructor reflected by the current instance. |

1
2
3
4

Permissions

| Permission | Description |
|---|---|
| System.Security.Permissions.ReflectionPermission | Requires permission to invoke non-public members of loaded assemblies. See System.Security.Permissions.ReflectionPermissionFlag.MemberAccess . |

5
6