System.Reflection.ConstructorInfo Structure

```
[ILAsm]
.class public abstract serializable ConstructorInfo extends
System.Reflection.MethodBase

[C#]
public abstract class ConstructorInfo: MethodBase
```

Assembly Info:

- *Name:* mscorlib
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- *Version:* 2.0.x.x
- *Attributes:*
    - CLSCompliantAttribute(true)

Summary

Provides access to constructor metadata.

Inherits From: System.Reflection.MethodBase

Library: Reflection

Thread Safety: This type is safe for multithreaded operations.

Permissions

Permission	Description
System.Security.Permissions. ReflectionPermission	Requires permission to reflect non-public members of a type in loaded assemblies. See `System.Security.Permissions. ReflectionPermissionFlag.TypeInformation`.

ConstructorInfo() Constructor

```
[ILAsm]
family rtspecialname specialname instance void .ctor()

[C#]
protected ConstructorInfo()
```

Summary

Constructs a new instance of the `System.Reflection.ConstructorInfo` class.

ConstructorInfo.ConstructorName Field

[ILAsm]
```
.field public static initOnly string ConstructorName
```

[C#]
```
public static readonly string ConstructorName
```

Summary

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

Description

This field is read-only.

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

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

For more information on object constructors, see Partition II of the CLI Specification.

]

ConstructorInfo.TypeConstructorName Field

```
[ILAsm]
.field public static initOnly string TypeConstructorName

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

Summary

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

Description

This field is read-only.

This field is a `System.String` that contains the value ".cctor".

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

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

For more information on type initializers, see Partition II of the CLI Specification.

]

ConstructorInfo.Invoke(System.Reflection.BindingFlags, System.Reflection.Binder, System.Object[], System.Globalization.CultureInfo) Method

```
[ILAsm]
.method public hidebysig virtual abstract object Invoke(valuetype
System.Reflection.BindingFlags invokeAttr, class System.Reflection.Binder
binder, object[] parameters, class System.Globalization.CultureInfo
culture)

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

Summary

Invokes the constructor reflected by the current instance using the specified arguments, under the constraints of the specified `System.Reflection.Binder`.

Parameters

Parameter	Description
invokeAttr	A `System.Reflection.BindingFlags` value that controls the binding process.
binder	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 *binder* is `null`, then the default binder is used.
parameters	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.]
culture	The only defined value for this parameter is `null`.

Return Value

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

Behaviors

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

Exceptions

Exception	Condition
System.ArgumentException	The types of the elements of *parameters* do not match the types of the parameters accepted by the constructor reflected by the current instance, under the constraints of *binder*.
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	*parameters*.Length does not equal the number of parameters required by the contract of the constructor reflected by the current instance.

Permissions

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

ConstructorInfo.Invoke(System.Object[]) Method

```
[ILAsm]
.method public hidebysig instance object Invoke(object[] parameters)

[C#]
public object Invoke(object[] parameters)
```

Summary

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

Parameters

Parameter	Description
parameters	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.]

Return Value

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

Exceptions

Exception	Condition
System.ArgumentException	The types of the elements of *parameters* do not match the types of the parameters accepted by the constructor reflected by the current instance, under the constraints of the default binder.
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	*parameters*.Length does not equal the number of parameters required by the contract of the constructor reflected by the current instance.

Permissions

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