

# System.FieldAccessException Class

```
[ILAsm]
.class public serializable FieldAccessException extends
System.MemberAccessException

[C#]
public class FieldAccessException: MemberAccessException
```

## 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

Represents the error that occurs when there is an attempt to access a field outside the scope in which access is permitted.

## Inherits From: System.MemberAccessException

**Library:** RuntimeInfrastructure

**Thread Safety:** All public static members of this type are safe for multithreaded operations. No instance members are guaranteed to be thread safe.

## Description

[*Note:* This exception is typically thrown when the access level of a field in a class library is changed, and one or more assemblies referencing the library have not been recompiled.]

## Example

The following example demonstrates a scenario under which `System.FieldAccessException` is thrown.

The following code contains a class with a public field (`myField`). This class is compiled into a class library.

[C#]

```
using System;
namespace TestNameSpace
{
```

```

1  public class Class1
2  {
3      public Class1()
4      {
5          Console.WriteLine ("Constructing with public field");
6      }
7      public int myField = -1;
8  }
9  }

```

10  
11 The following code references the class library above, and accesses  
12 TestNameSpace.Class1.myField. This code is compiled into an application.

```

13
14 [C#]
15 using System;
16 using TestNameSpace;
17 class AppTest
18 {
19     public static void Main()
20     {
21         Class1 test = new Class1();
22         Console.WriteLine("Accessing member {0}.", test.myField);
23     }
24 }
25

```

26 The output of the application is

```

27
28 Constructing with public field
29
30
31 Accessing member -1.
32
33

```

34 The code for the class library is changed and recompiled so that  
35 TestNameSpace.Class1.myField is no longer public. The following code changes myField  
36 from public to private.

```

37
38 [C#]
39 using System;
40 namespace TestNameSpace
41 {
42     public class Class1
43     {
44         public Class1()
45         {
46             Console.WriteLine ("Constructing with private field");
47         }
48         private int myField = -1;
49     }
50 }
51

```

```
1  When the application is executed again without being recompiled, the output is
2
3  Unhandled Exception: System.FieldAccessException:
4  TestNameSpace.Class1.myField
5
6
7  at AppTest.Main()
8
9
```

# FieldAccessException() Constructor

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

[C#]
public FieldAccessException()
```

## Summary

Constructs and initializes a new instance of the `System.FieldAccessException` class.

## Description

This constructor initializes the `System.FieldAccessException.Message` property of the new instance to a system-supplied message that describes the error, such as "Attempted to access a private or protected field inside a type." This message takes into account the current system culture.

The `System.FieldAccessException.InnerException` property of the new instance is initialized to `null`.

# FieldAccessException(System.String)

## Constructor

```
[ILAsm]
public rtspecialname specialname instance void .ctor(string message)

[C#]
public FieldAccessException(string message)
```

### Summary

Constructs and initializes a new instance of the `System.FieldAccessException` class.

### Parameters

Parameter	Description
<i>message</i>	A <code>System.String</code> that describes the error. The content of <i>message</i> is intended to be understood by humans. The caller of this constructor is required to ensure that this string has been localized for the current system culture.

### Description

This constructor initializes the `System.FieldAccessException.Message` property of the new instance using *message*. If *message* is null, the `System.FieldAccessException.Message` property is initialized to the system-supplied message provided by the constructor that takes no arguments.

The `System.FieldAccessException.InnerException` property of the new instance is initialized to null.

# FieldAccessException(System.String, System.Exception) Constructor

```
[ILAsm]
public rtspecialname specialname instance void .ctor(string message, class
System.Exception inner)

[C#]
public FieldAccessException(string message, Exception inner)
```

## Summary

Constructs and initializes a new instance of the `System.FieldAccessException` class.

## Parameters

Parameter	Description
<i>message</i>	A <code>System.String</code> that describes the error. The content of <i>message</i> is intended to be understood by humans. The caller of this constructor is required to ensure that this string has been localized for the current system culture.
<i>inner</i>	An instance of <code>System.Exception</code> that is the cause of the current exception. If <i>inner</i> is not a null reference, the current exception was raised in a catch block handling <i>inner</i> .

## Description

This constructor initializes the `System.FieldAccessException.Message` property of the new instance using *message* and the `System.FieldAccessException.InnerException` property using *inner*. If *message* is null, the `System.FieldAccessException.Message` property is initialized to the system-supplied message provided by the constructor that takes no arguments.

[*Note:* For more information on inner exceptions, see `System.Exception.InnerException`.]