

# System.FieldAccessException Class

1  
2  
3

```
4 [ILASM]
5 .class public serializable FieldAccessException extends
6 System.MemberAccessException
7
8 [C#]
9 public class FieldAccessException: MemberAccessException
```

## Assembly Info:

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

## Summary

15  
16

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

## Inherits From: System.MemberAccessException

19  
20

21 **Library:** RuntimeInfrastructure

22

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

25

## Description

26

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

## Example

30  
31

32 The following example demonstrates a scenario under which  
33 **System.FieldAccessException** is thrown.

34

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

37

38 [C#]

```
39 using System;
40 namespace TestNameSpace
```

```
1      {
2      public class Class1
3      {
4      public Class1()
5      {
6      Console.WriteLine ("Constructing with public field");
7      }
8      public int myField = -1;
9      }
10     }
11
```

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

15 [C#]  
16

```
17 using System;
18 using TestNameSpace;
19 class AppTest
20 {
21     public static void Main()
22     {
23         Class1 test = new Class1();
24         Console.WriteLine("Accessing member {0}.",
25 test.myField);
26     }
27 }
28
```

29 The output of the application is

30  
31 Constructing with public field

32  
33  
34 Accessing member -1.

35  
36  
37 The code for the class library is changed and recompiled so that

1 TestNameSpace.Class1.myField is no longer public. The following code  
2 changes myField from public to private.

```
3 [C#]  
4  
5 using System;  
6 namespace TestNameSpace  
7 {  
8     public class Class1  
9     {  
10         public Class1()  
11         {  
12             Console.WriteLine ("Constructing with private field");  
13         }  
14         private int myField = -1;  
15     }  
16 }  
17
```

18 When the application is executed again without being recompiled, the  
19 output is

```
20 Unhandled Exception: System.FieldAccessException:  
21 TestNameSpace.Class1.myField  
22  
23  
24  
25 at AppTest.Main()  
26
```

27

# 1 `FieldAccessException()` Constructor

```
2 [ILASM]  
3 public rtspecialname specialname instance void .ctor()  
4 [C#]  
5 public FieldAccessException()
```

## 6 **Summary**

7 Constructs and initializes a new instance of the  
8 **System.FieldAccessException** class.

## 9 **Description**

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

15  
16 The **System.FieldAccessException.InnerException** property of the  
17 new instance is initialized to **null**.

18

# 1 FieldAccessException(System.String)

## 2 Constructor

```
3 [ILASM]  
4 public rtspecialname specialname instance void .ctor(string  
5 message)  
  
6 [C#]  
7 public FieldAccessException(string message)
```

### 8 Summary

9 Constructs and initializes a new instance of the  
10 **System.FieldAccessException** class.

### 11 Parameters

12  
13

Parameter	Description
<i>message</i>	A <b>System.String</b> 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.

14

### 15 Description

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

22

23 The **System.FieldAccessException.InnerException** property of the  
24 new instance is initialized to **null**.

25

# 1 FieldAccessException(System.String, 2 System.Exception) Constructor

```
3 [ILASM]  
4 public rtspecialname specialname instance void .ctor(string  
5 message, class System.Exception inner)  
  
6 [C#]  
7 public FieldAccessException(string message, Exception  
8 inner)
```

## 9 Summary

10 Constructs and initializes a new instance of the  
11 **System.FieldAccessException** class.

## 12 Parameters

13  
14

Parameter	Description
<i>message</i>	A <b>System.String</b> 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 <b>System.Exception</b> that is the cause of the current exception. If <i>inner</i> is not a <b>null</b> reference, the current exception was raised in a catch block handling <i>inner</i> .

15  
16

## Description

17 This constructor initializes the  
18 **System.FieldAccessException.Message** property of the new  
19 instance using *message* and the  
20 **System.FieldAccessException.InnerException** property using  
21 *inner*. If *message* is **null**, the  
22 **System.FieldAccessException.Message** property is initialized to the  
23 system-supplied message provided by the constructor that takes no  
24 arguments.

25  
26  
27

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

28