

System.TypeInitializationException Class

```
[ILASM]
.class public sealed serializable
TypeInitializationException extends System.SystemException

[C#]
public sealed class TypeInitializationException:
SystemException
```

Assembly Info:

- Name: mscorlib
- Public Key: [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- Version: 1.0.x.x
- Attributes:
 - CLSCompliantAttribute(true)

Summary

Represents the error that occurs when an exception is thrown inside the static constructor of a type.

Inherits From: System.SystemException

Library: BCL

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

Description

When a static constructor fails to initialize a type, a **System.TypeInitializationException** instance is created and passed a reference to the exception thrown by the static constructor. The **System.TypeInitializationException.InnerException** property stores the exception that was thrown by the static constructor.

Example

The following example demonstrates an error that causes a **System.TypeInitializationException** exception.

```
[C#]
```

```
using System;
class TypeInit {
```

```

1      // Static constructor
2      static TypeInit () {
3          // Throw an application-defined exception.
4          throw new ApplicationException("Error in Class
5      TypeInit");
6      }
7      public TypeInit() {}
8  }
9  class TestTypeInit {
10     static public void Main() {
11         try {
12             TypeInit t = new TypeInit ();
13         }
14         catch (TypeInitializationException e) {
15             Console.WriteLine("Error: {0}",e);
16         }
17     }
18 }
19

```

20 The output is

```

21
22 Error: System.TypeInitializationException: The type
23 initializer for "TypeInit" threw an exception. --->
24 System.ApplicationException: Error in Class TypeInit
25     at TypeInit..cctor()
26     --- End of inner exception stack trace ---
27     at TypeInit..cctor()
28     at TestTypeInit.Main()
29
30

```

1 **TypeInitializationException.TypeName** 2 **Property**

```
3   [ILASM]  
4   .property string TypeName { public hidebysig specialname  
5   instance string get_TypeName() }  
  
6   [C#]  
7   public string TypeName { get; }
```

8 **Summary**

9 Gets the fully qualified name of the type that causes the exception.

10 **Property Value** 11

12 A **System.String** that contains the fully qualified name of the type
13 that caused the exception.

14 **Description**

15 This property is read-only.

16
17 The **System.TypeInitializationException.TypeName** property
18 returns the same value as was passed into the constructor.

19