

# 1 System.ObjectDisposedException Class

```
2 [ILAsm]  
3 .class public serializable ObjectDisposedException extends  
4 System.InvalidOperationException  
  
5 [C#]  
6 public class ObjectDisposedException: InvalidOperationException
```

## 7 Assembly Info:

- 8 • *Name:* mscorlib
- 9 • *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- 10 • *Version:* 2.0.x.x
- 11 • *Attributes:*
  - 12 ○ CLSCompliantAttribute(true)

## 13 Summary

14 Represents the error that occurs when an operation is performed on a disposed object.

## 15 Inherits From: System.InvalidOperationException

16

17 **Library:** BCL

18

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

21

## 22 Description

23 [Note: For additional information about disposing objects, see the System.IDisposable  
24 interface.]

25

26

## 27 Example

28 The following example demonstrates an error that causes the  
29 System.ObjectDisposedException exception to be thrown.

30

31 [C#]

```
32 using System;  
33 using System.IO;  
34  
35 public class ObjectDisposedExceptionTest {  
36     public static void Main() {  
37         MemoryStream ms = new MemoryStream(16);  
38         ms.Close();  
39         try {
```

```
1         ms.ReadByte();
2     }
3     catch (ObjectDisposedException e) {
4         Console.WriteLine("Caught: {0}", e.Message);
5     }
6 }
7 }
```

8 The output is

9  
10 Caught: Cannot access a closed Stream.

11

12

# 1 ObjectDisposedException(System.String)

## 2 Constructor

```
3 [ILAsm]  
4 public rtspecialname specialname instance void .ctor(string objectName)  
5 [C#]  
6 public ObjectDisposedException(string objectName)
```

### 7 Summary

8 Constructs and initializes a new instance of the System.ObjectDisposedException  
9 class.

### 10 Parameters

Parameter	Description
<i>objectName</i>	A System.String containing the name of the disposed object.

### 11 Description

13 This constructor initializes the System.ObjectDisposedException.ObjectName property  
14 of the new instance using *objectName*. The  
15 System.ObjectDisposedException.Message property is initialized to a system-supplied  
16 message that describes the error and includes *objectname*. This message takes into  
17 account the current system culture.

18  
19 The System.ObjectDisposedException.InnerException property of the new instance  
20 is initialized to null.

21  
22 [Note: If *objectName* is null, the System.ObjectDisposedException.Message property  
23 contains only an error message.]

### 26 Example

27 The following example displays the error message of a  
28 System.ObjectDisposedException instance created using this constructor.

```
29 [C#]  
30  
31 using System;  
32  
33 public class ExampleDisposableObject: IDisposable {  
34     public static void Main() {  
35  
36         ExampleDisposableObject obj = new ExampleDisposableObject();
```

```
1
2  obj.Close();
3
4  try {
5  Console.WriteLine(obj);
6  } catch (ObjectDisposedException e) {
7  Console.WriteLine("Caught: {0}", e.Message);
8  }
9  }
10
11
12  public ExampleDisposableObject() {
13  isDisposed = false;
14  }
15
16  ~ExampleDisposableObject() {
17  Dispose(true);
18  }
19
20  public void Close() {
21  Dispose(true);
22  }
23
24  public void Dispose() {
25  Dispose(true);
26  }
27
28  public void Dispose(bool disposing) {
29  isDisposed = true;
30  }
31
32  public override String ToString() {
33  if(isDisposed)
34  throw new ObjectDisposedException("ExampleDisposableObject");
35  else
36  return "This is an instance of ExampleDisposableObject.";
37  }
38
39  private bool isDisposed;
40  }
41
42  The output is
43
44  Caught: Cannot access a disposed object named "ExampleDisposableObject".
45
46
47  Object name: "ExampleDisposableObject".
48
49
```

# 1 ObjectDisposedException(System.String, 2 System.String) Constructor

```
3 [ILAsm]  
4 public rtspecialname specialname instance void .ctor(string objectName,  
5 string message)  
  
6 [C#]  
7 public ObjectDisposedException(string objectName, string message)
```

## 8 Summary

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

## 11 Parameters

Parameter	Description
<i>objectName</i>	A System.String containing the name of the disposed object.
<i>message</i>	A System.String 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.

12

## 13 Description

14 This constructor initializes the System.ObjectDisposedException.Message property of  
15 the new instance using *message*, and the  
16 System.ObjectDisposedException.ObjectName property using *objectName*. If *message*  
17 is null, the System.ObjectDisposedException.Message property is initialized to the  
18 system-supplied message provided by the constructor that takes no arguments.

19

20 The System.ObjectDisposedException.InnerException property of the new instance  
21 is initialized to null.

## 22 Example

23 The following example throws a System.ObjectDisposedException instance created  
24 using this constructor.

25

26 [C#]

27 using System;

28

```
29 public class ExampleDisposableObject: IDisposable {  
30     public static void Main() {
```

31

```

1  ExampleDisposableObject obj = new ExampleDisposableObject();
2
3  obj.Close();
4
5  try {
6  Console.WriteLine(obj);
7  } catch (ObjectDisposedException e) {
8  Console.WriteLine("Caught: {0}", e.Message);
9  }
10 }
11
12
13 public ExampleDisposableObject() {
14     isDisposed = false;
15 }
16
17 ~ExampleDisposableObject() {
18     Dispose(true);
19 }
20
21 public void Close() {
22     Dispose(true);
23 }
24
25 public void Dispose() {
26     Dispose(true);
27 }
28
29 public void Dispose(bool disposing) {
30     isDisposed = true;
31 }
32
33 public override String ToString() {
34     if(isDisposed) {
35         string message = "Oh-oh! This object has been disposed!";
36         string objectName = "ExampleDisposableObject";
37         throw new ObjectDisposedException(objectName, message);
38     }
39     else
40         return "Hello, World!";
41 }
42
43 private bool isDisposed;
44 }
45 The output is
46
47 Caught: Oh-oh! This object has been disposed!
48
49
50 Object name: "ExampleDisposableObject".
51
52

```

# 1 ObjectDisposedException.Message Property

```
2 [ILAsm]  
3 .property string Message { public hidebysig virtual specialname string  
4 get_Message() }  
5 [C#]  
6 public override string Message { get; }
```

## 7 Summary

8 Gets the message that describes the error.

## 9 Property Value

10 A System.String that describes the error.

## 11 Description

12 If the System.ObjectDisposedException.ObjectName property is not null, the  
13 message includes the name of the object.

14  
15 This property is read-only.

16  
17 [*Note:* This property overrides System.Exception.Message.]  
18  
19

20

# 1 ObjectDisposedException.ObjectName

## 2 Property

```
3 [ILAsm]  
4 .property string ObjectName { public hidebysig specialname instance string  
5 get_ObjectName() }  
  
6 [C#]  
7 public string ObjectName { get; }
```

### 8 Summary

9 Gets the name of the disposed object.

### 10 Property Value

11 A `System.String` containing the name of the disposed object.

### 12 Description

13 [*Note:* If this property is not null or `System.String.Empty`, the value of this property is  
14 included in the string returned by the `System.ObjectDisposedException.Message`  
15 property.]

16  
17

18 This property is read-only.  
19

20