IOException Class

.NET Framework 2.0

The exception that is thrown when an I/O error occurs.

Namespace: System.IO
Assembly: mscorlib (in mscorlib.dll)

Syntax

C#

```
[SerializableAttribute]
[ComVisibleAttribute(true)]
public class IOException : SystemException
```

J#

```
/** @attribute SerializableAttribute() */
/** @attribute ComVisibleAttribute(true) */
public class IOException extends SystemException
```

JScript

```
SerializableAttribute
ComVisibleAttribute(true)
public class IOException extends SystemException
```

Remarks

IOException is the base class for exceptions thrown while accessing information using streams, files and directories.

The Base Class Library includes the following types, each of which is a derived class of **IOException** :

- DirectoryNotFoundException
- EndOfStreamException
- FileNotFoundException
- FileLoadException

- PathTooLongException

Where appropriate, use these types instead of IOException.

IOException uses the HRESULT COR_E_IO which has the value 0x80131620.

Example

This code example is part of a larger example provided for the FileStream.Lock method.

C#

```
// Catch the IOException generated if the
// specified part of the file is locked.
catch(IOException e)
{
    Console.WriteLine(
        "{0}: The write operation could not " +
        "be performed because the specified " +
        "part of the file is locked.",
        e.GetType().Name);
}
```

J#

```
// Catch the IOException generated if the
// specified part of the file is locked.
catch(IOException e) {
    Console.WriteLine(
        "{0}: The write operation could not "
        + "be performed because the specified "
        + "part of the file is locked.",
        e.GetType().get_Name());
}
```

Inheritance Hierarchy

System.Object
  System.Exception
    System.SystemException
      System.IO.IOException
        Derived Classes

Thread Safety

Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Windows 98, Windows 2000 SP4, Windows CE, Windows Millennium Edition, Windows Mobile for Pocket PC, Windows Mobile for Smartphone, Windows Server 2003, Windows XP Media Center Edition, Windows XP Professional x64 Edition, Windows XP SP2, Windows XP Starter Edition

The .NET Framework does not support all versions of every platform. For a list of the supported versions, see System Requirements.

Version Information

.NET Framework
Supported in: 2.0, 1.1, 1.0
.NET Compact Framework
Supported in: 2.0, 1.0

See Also

Reference
IOException Members
System.IO Namespace
Exception
Other Resources
Handling and Throwing Exceptions
File and Stream I/O
How to: Read Text from a File
How to: Write Text to a File

Community Additions

HRESULT Not Always COR_E_IO

The documentation vaguely implies that the HRESULT Is always COR_E_IO which has the value 0x80131620. However, I have discovered by experiment that the actual code is E_HANDLE (0x80070006L) when the exception is the result of an invalid handle, such as arises when Console.Clear() is called by a process whose standard output handle (Console.Out) is redirected.

That being so, I am writing a non-throwing wrapper around Console.Clear() that verifies the HRESULT to determine whether the exception is one that can be anticipated, or something else worthy of reporting.

David A. Gray
3/30/2016 Edit Post