

How to explicitly throw exceptions

03/30/2017 • 2 minutes to read •  +5

In this article

[See also](#)

You can explicitly throw an exception using the C# [throw](#) or the Visual Basic [Throw](#) statement. You can also throw a caught exception again using the `throw` statement. It is good coding practice to add information to an exception that is re-thrown to provide more information when debugging.

The following code example uses a `try/catch` block to catch a possible [FileNotFoundException](#). Following the `try` block is a `catch` block that catches the [FileNotFoundException](#) and writes a message to the console if the data file is not found. The next statement is the `throw` statement that throws a new [FileNotFoundException](#) and adds text information to the exception.

C#

 Copy

```
using System;
using System.IO;

public class ProcessFile
{
    public static void Main()
    {
        FileStream fs;
        try
        {
            // Opens a text file.
            fs = new FileStream(@"C:\temp\data.txt", FileMode.Open);
            var sr = new StreamReader(fs);

            // A value is read from the file and output to the console.
            string line = sr.ReadLine();
            Console.WriteLine(line);
        }
        catch (FileNotFoundException e)
        {
            Console.WriteLine($"[Data File Missing] {e}");
            throw new FileNotFoundException(@"[data.txt not in c:\temp di-
rectory]", e);
        }
        finally
        {
            if (fs != null)
                fs.Close();
        }
    }
}
```

```
}  
    }  
}
```

See also

- [Exceptions](#)

Is this page helpful?

 Yes  No

Recommended content

Creating and Throwing Exceptions

Learn about creating and throwing exceptions. Exceptions are used to indicate that an error has occurred while running a program.

How to convert a string to a number - C# Programming Guide

Learn how to convert a string to a number in C# by calling the Parse, TryParse, or Convert class methods.

C# switch statement

switch (C# reference)

Boolean logical operators - C# reference

Learn about C# operators that perform logical negation, conjunction (AND), and inclusive and exclusive disjunction (OR) operations with Boolean operands.

Show more 