Logging

Logging

Logging is a means of auditing or keeping track of every activity that has taken place in the application or system.

Logging

This is crucial as this can tell us the activities that took place at any particular point in time during the lifecycle of the application.

Logging

Let’s say that we want to display a message on the Console “I’ve Started Logging” on the Console but also to log it to a file. How might we do that?

Logging

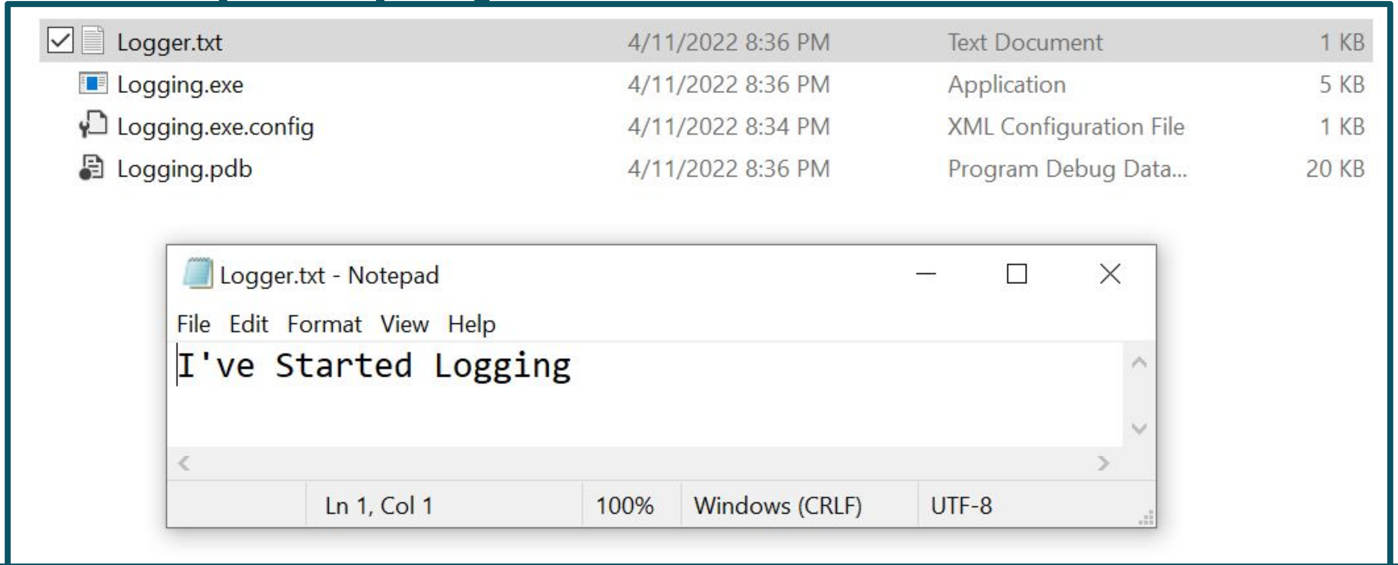
With the previous Knowledge, we can simply make a log file (.txt) and log the contents into that file.

Logging

```
using System;
using System.IO;

namespace Logging
{
    class Program
    {
        static void Main(string[] args)
        {
            StreamWriter logger = new StreamWriter("Logger.txt", true);
            Console.WriteLine("I've Started Logging");
            logger.WriteLine("I've Started Logging");
            logger.Close();
            Console.ReadKey();

        }
    }
}
```

Logging

Go to Your project directory and in the Bin folder. You'll see a file (Logger.txt) hanging out there in addition to your project files.



Logger.txt	4/11/2022 8:36 PM	Text Document	1 KB
Logging.exe	4/11/2022 8:36 PM	Application	5 KB
Logging.exe.config	4/11/2022 8:34 PM	XML Configuration File	1 KB
Logging.pdb	4/11/2022 8:36 PM	Program Debug Data...	20 KB

```
I've Started Logging
```

Logging

This is the basic idea of Logging your program activities in the file for future reference.

Logging

Logging is the practice of determining what information is useful to capture and then recording it somewhere for future access.

Logging

Is that all we need to know about logging?

Of course not..!!

This is a lot more goes into a proper logging strategy than just randomly dumping text.

Logging

Let's see some of the components that go into creating entries in your log.

Conceptually, you can think of each entry in a log as an event.

Logging

Here are some things that you'll usually want to capture for each event:

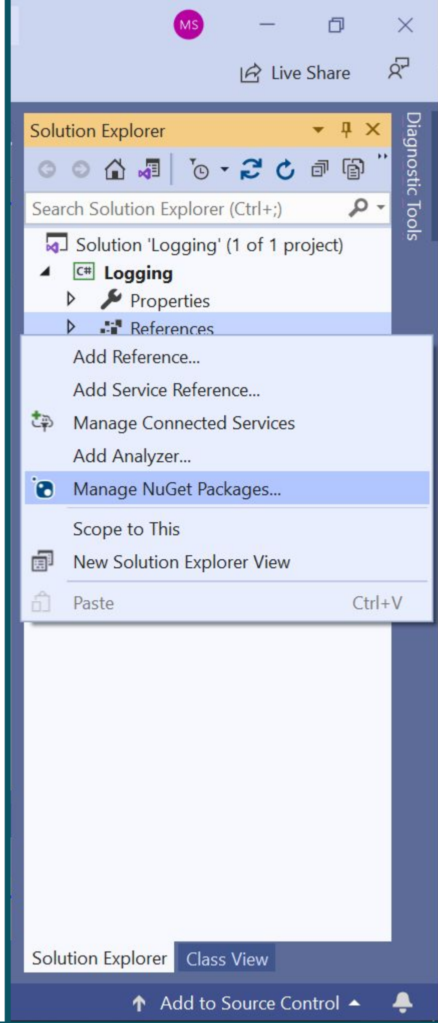
- A unique identifier for the event.
- A timestamp. Exactly when did this event take place?
- Context that makes it clear to a reader what's going on. For instance, just recording "I've Started Logging" might prove confusing weeks or months later. A better message, including context, might say, "Recorded the activity at 23 April 2022 of 'I've Started Logging'"
- Tags and categories for each entry for later search and classification.
- Log levels, such as "error," "warning," or "information," that allow further filtering and provide additional context.

Logging: Log4Net Package

- We do not have to do that all by ourselves. Programmers have provided us packages for logging.
- One such package is Log4Net.

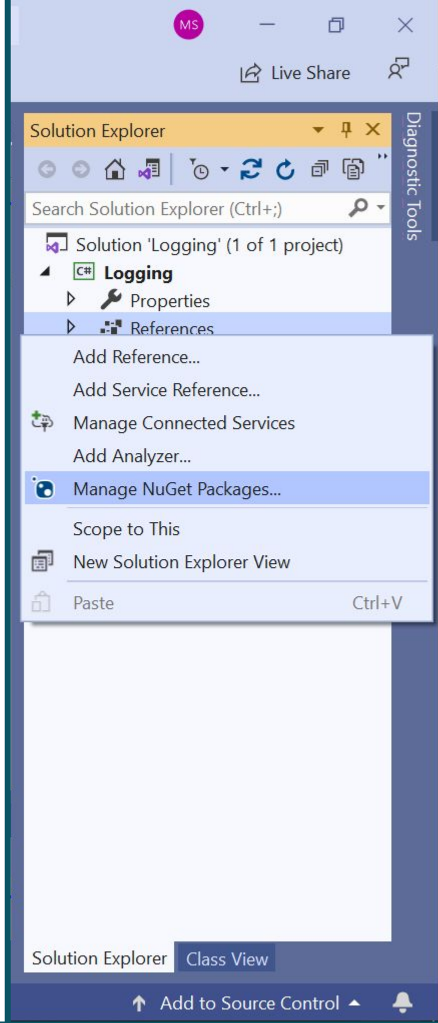
Logging: Log4Net Package

- In order to install the Package, right click on the references in Solution Explorer Window



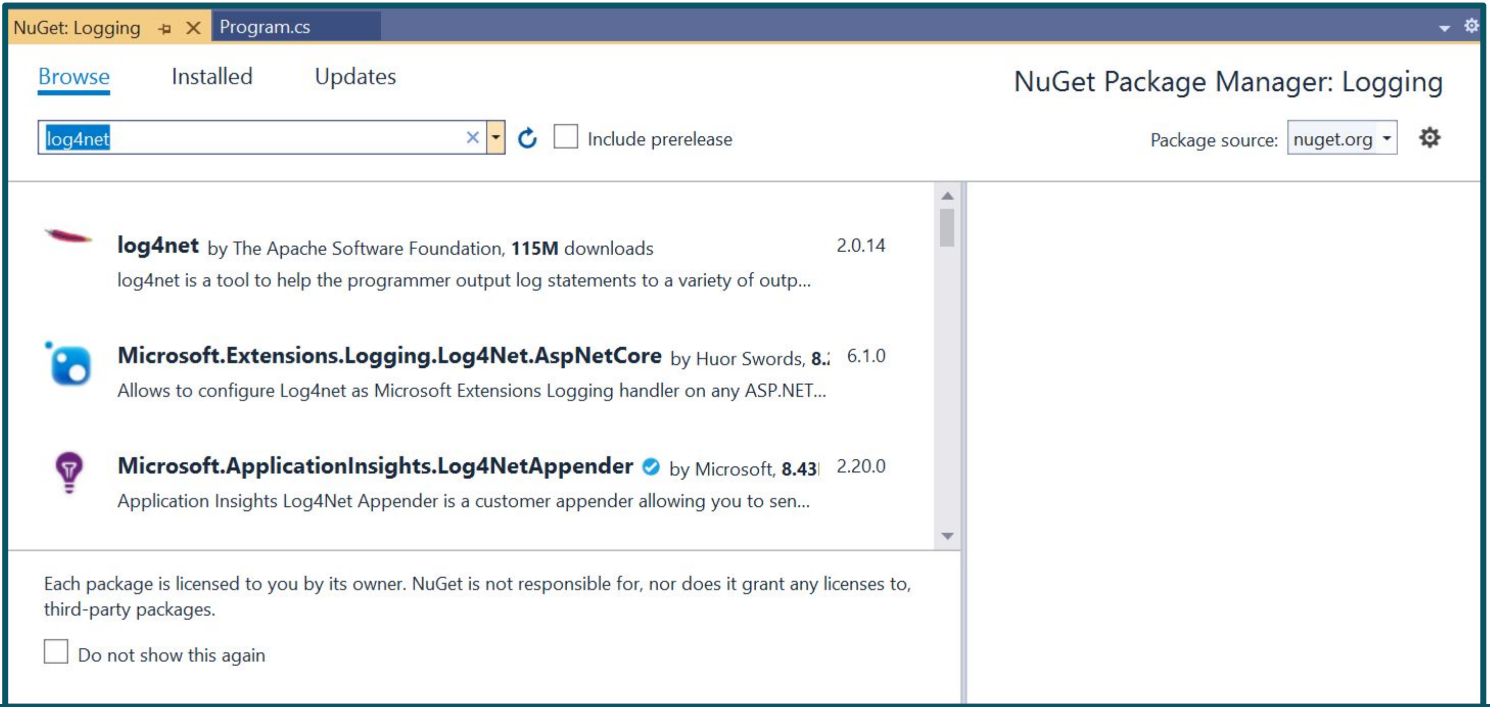
Logging: Log4Net Package

- Then Click on the Manage NuGet Packages.



Logging: Log4Net Package

- Then Search Log4Net and install the package.



NuGet Package Manager: Logging

log4net

log4net by The Apache Software Foundation, 115M downloads 2.0.14
log4net is a tool to help the programmer output log statements to a variety of outp...

Microsoft.Extensions.Logging.Log4Net.AspNetCore by Huor Swords, 8.: 6.1.0
Allows to configure Log4net as Microsoft Extensions Logging handler on any ASP.NET...

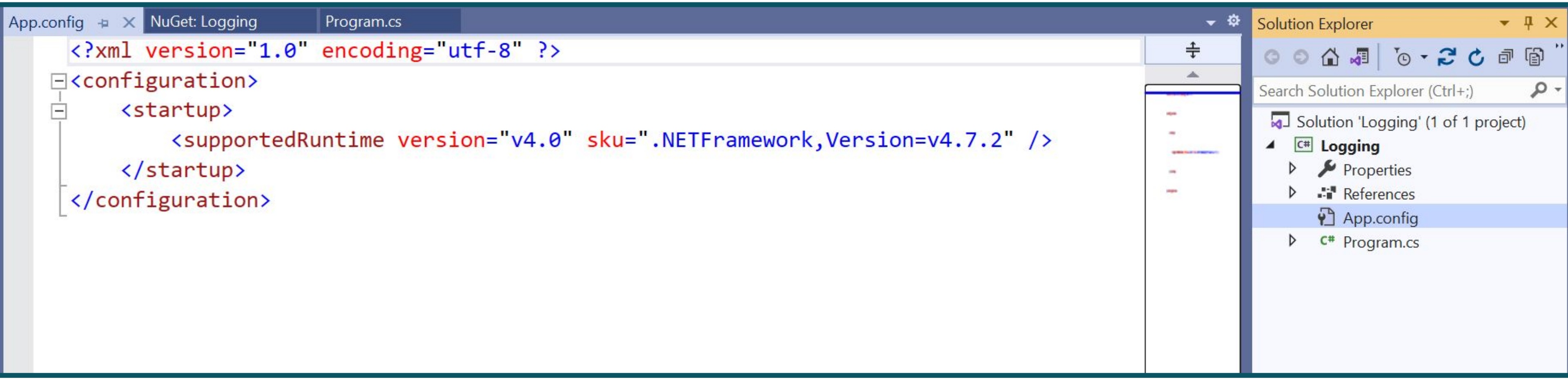
Microsoft.ApplicationInsights.Log4NetAppender by Microsoft, 8.43 2.20.0
Application Insights Log4Net Appender is a customer appender allowing you to sen...

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

☐ Do not show this again

Logging: Log4Net Package

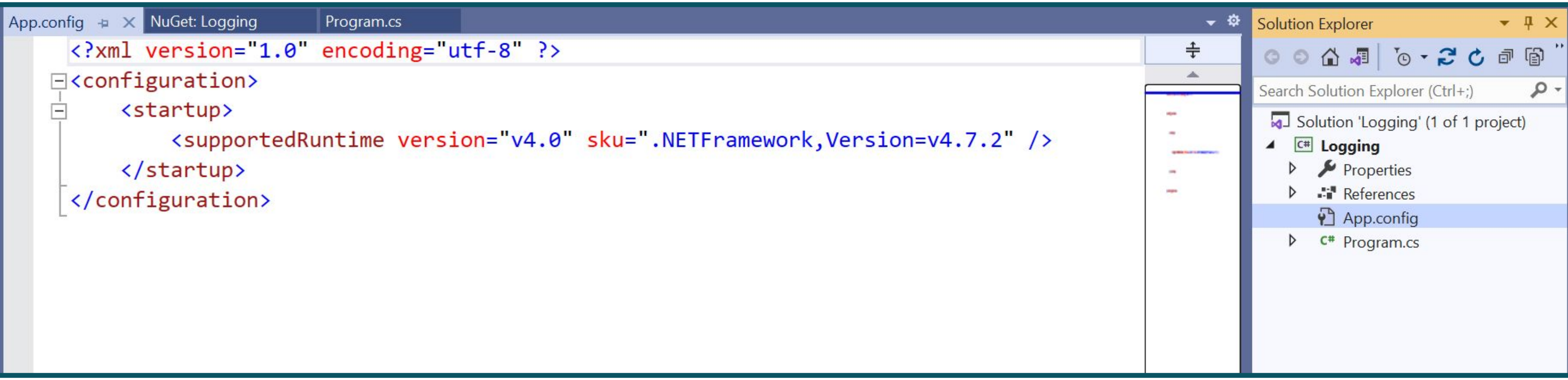
- After installing you have to Configure your App.config File. Click on the App.Config and following file will open



```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
    <startup>
        <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
    </startup>
</configuration>
```

Logging: Log4Net Package

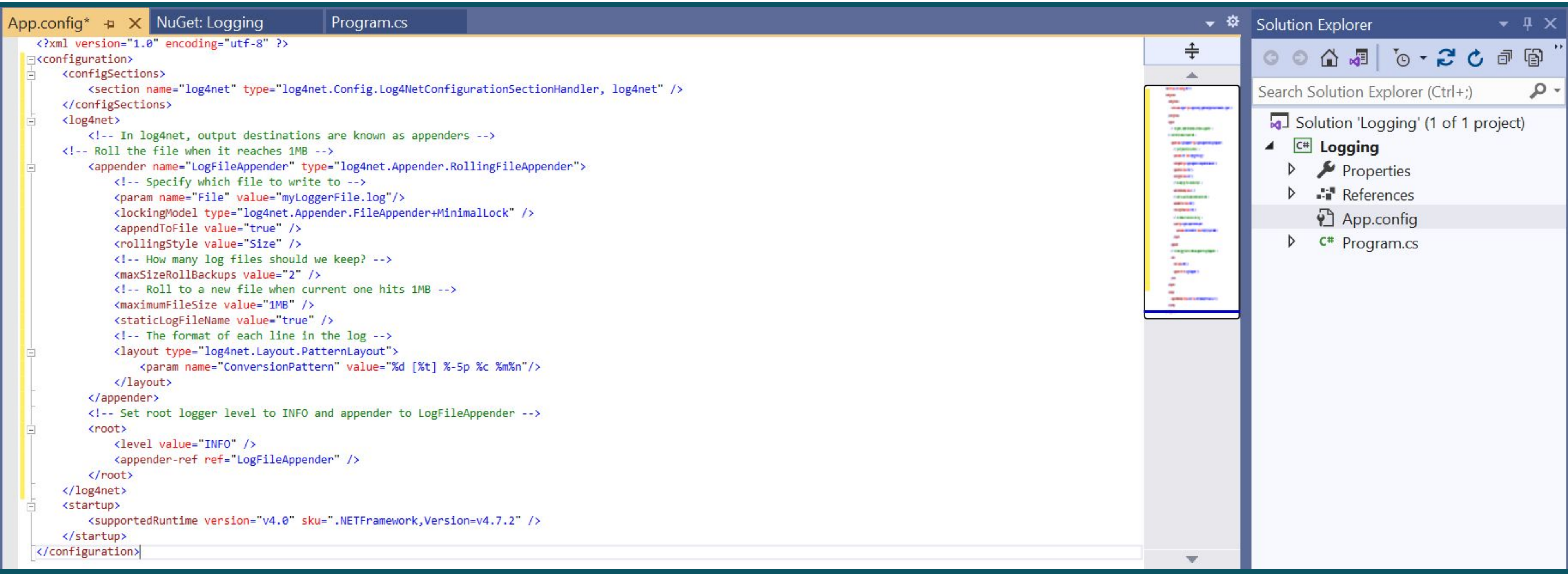
- Save the following within configuration tag.



```xml
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
    <startup>
        <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
    </startup>
</configuration>
```

Logging: Log4Net Package

- Save the following within configuration tag.



```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
    <configSections>
        <section name="log4net" type="log4net.Config.Log4NetConfigurationSectionHandler, log4net" />
    </configSections>
    <log4net>
        <!-- In log4net, output destinations are known as appenders -->
    <!-- Roll the file when it reaches 1MB -->
        <appender name="LogFileAppender" type="log4net.Appender.RollingFileAppender">
            <!-- Specify which file to write to -->
            <param name="File" value="myLoggerFile.log"/>
            <lockingModel type="log4net.Appender.FileAppender+MinimalLock" />
            <appendToFile value="true" />
            <rollingStyle value="Size" />
            <!-- How many log files should we keep? -->
            <maxSizeRollBackups value="2" />
            <!-- Roll to a new file when current one hits 1MB -->
            <maximumFileSize value="1MB" />
            <staticLogFileName value="true" />
            <!-- The format of each line in the log -->
            <layout type="log4net.Layout.PatternLayout">
                <param name="ConversionPattern" value="%d [%t] %-5p %c %m%n"/>
            </layout>
        </appender>
        <!-- Set root logger level to INFO and appender to LogFileAppender -->
        <root>
            <level value="INFO" />
            <appender-ref ref="LogFileAppender" />
        </root>
    </log4net>
    <startup>
        <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
    </startup>
</configuration>
```

Logging: Code

```csharp
using System;
using System.IO;
using log4net;
using log4net.Config;
using System.Reflection;

namespace Logging
{
    class Program
    {
        static readonly ILog logger =
LogManager.GetLogger(MethodBase.GetCurrentMethod().DeclaringType);
        static void Main(string[] args)
        {
            Console.WriteLine("I've Started Logging");
            XmlConfigurator.Configure();
            logger.Info("Info message: I've Started Logging");
            Console.ReadKey();
        }
    }
}
```

Logging: Step 1

Include these libraries

```
using System;
using System.IO;
using log4net;
using log4net.Config;
using System.Reflection;

namespace Logging
{
    class Program
    {
        static readonly ILog logger =
LogManager.GetLogger(MethodBase.GetCurrentMethod().DeclaringType);
        static void Main(string[] args)
        {
            Console.WriteLine("I've Started Logging");
            XmlConfigurator.Configure();
            logger.Info("Info message: I've Started Logging");
            Console.ReadKey();
        }
    }
}
```

Logging: Step 2

Declare this Variable

```
using System;
using System.IO;
using log4net;
using log4net.Config;
using System.Reflection;

namespace Logging
{
    class Program
    {
        static readonly ILog logger =
LogManager.GetLogger(MethodBase.GetCurrentMethod().DeclaringType);
        static void Main(string[] args)
        {
            Console.WriteLine("I've Started Logging");
            XmlConfigurator.Configure();
            logger.Info("Info message: I've Started Logging");
            Console.ReadKey();
        }
    }
}
```

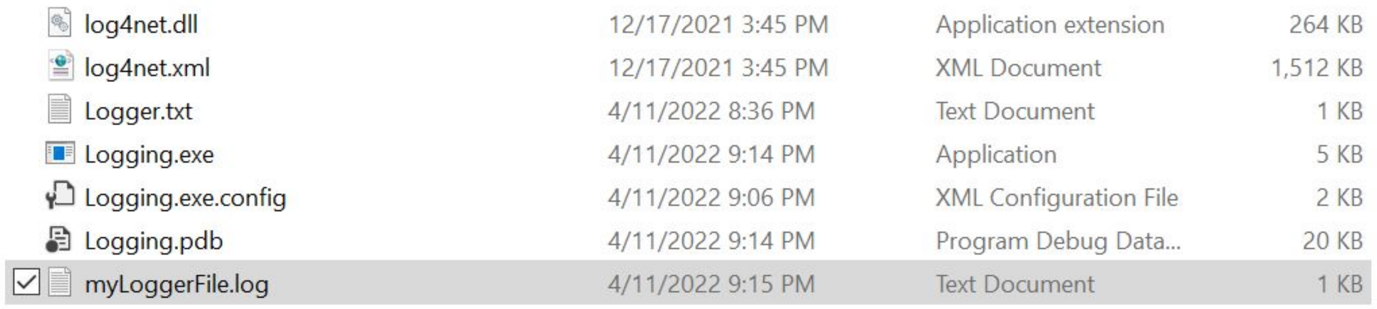
Logging: Step 3

Log the information

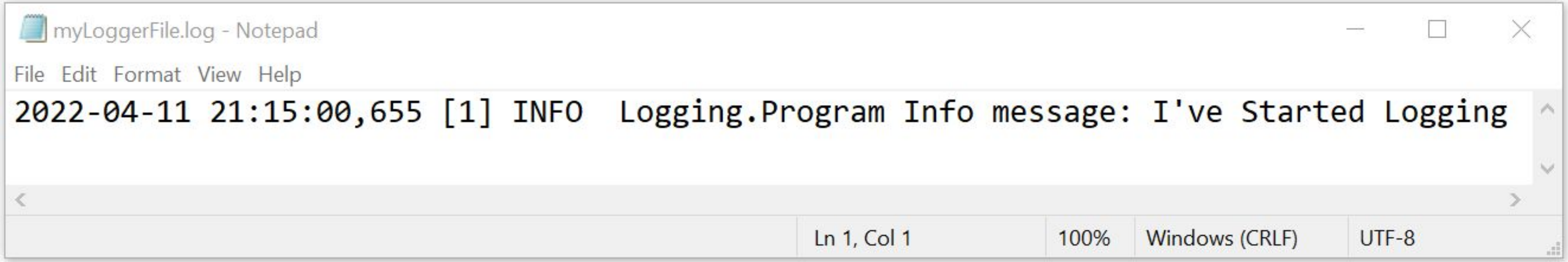
```
using System;
using System.IO;
using log4net;
using log4net.Config;
using System.Reflection;

namespace Logging
{
    class Program
    {
        static readonly ILog logger =
LogManager.GetLogger(MethodBase.GetCurrentMethod().DeclaringType);
        static void Main(string[] args)
        {
            Console.WriteLine("I've Started Logging");
            XmlConfigurator.Configure();
            logger.Info("Info message: I've Started Logging");
            Console.ReadKey();
        }
    }
}
```

Logging

myLoggerFile.log is created in the bin folder.



log4net.dll	12/17/2021 3:45 PM	Application extension	264 KB
log4net.xml	12/17/2021 3:45 PM	XML Document	1,512 KB
Logger.txt	4/11/2022 8:36 PM	Text Document	1 KB
Logging.exe	4/11/2022 9:14 PM	Application	5 KB
Logging.exe.config	4/11/2022 9:06 PM	XML Configuration File	2 KB
Logging.pdb	4/11/2022 9:14 PM	Program Debug Data...	20 KB
✓ myLoggerFile.log	4/11/2022 9:15 PM	Text Document	1 KB



```
2022-04-11 21:15:00,655 [1] INFO  Logging.Program Info message: I've Started Logging
```

Logging

This is just the basic on Logging.

For further reading, Explore this link:

https://www.papertrail.com/solution/tips/7-best-practices-for-c-logging-with-examples/

Conclusion

- Logging is necessary for the development and operations teams to track down and fix bugs quickly
- Instead of reinventing the wheel, we can use an existing logging framework such as log4net.
- Use context-rich logging so that we'll have all the information we might need when troubleshooting.

Learning Objective

Log the information in the Programs to trace down the changes while the execution of Program.