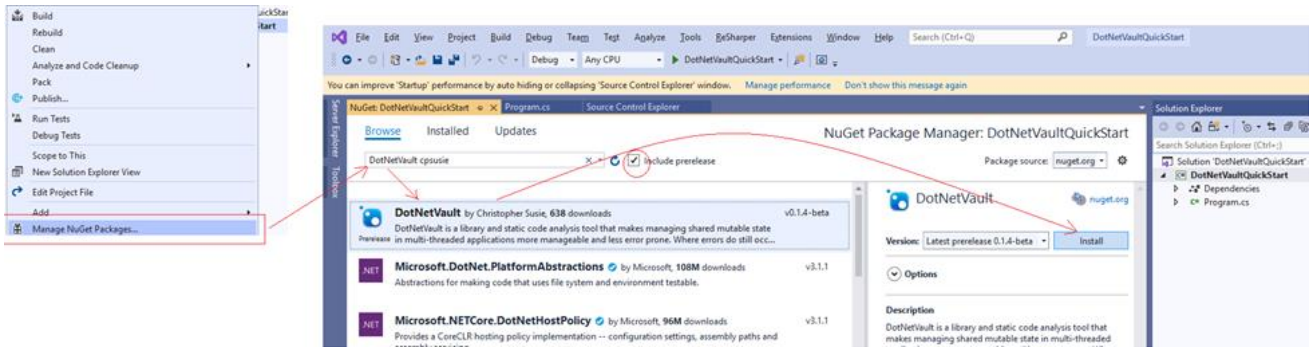
DotNetVault Quick Start Installation Guide Visual Studio 2019 (Windows 10)

1. Open Visual Studio and Create a new .NET Core 3.1+ or .NET Framework 4.8 Console Application.

2. **If you chose .NET Framework 4.8**, open the .csproj file and add the following Line `<LangVersion>8.0</LangVersion>` under each "Platform/Config" PropertyGroup as shown in highlight below. This is unnecessary if you chose a .NET Core 3.1+ based Console Application.

3. Right click on your project and chose "Manage NuGet packages". Click on the Browse option, select "Include Prerelease", enter "cpsusie dotnetvault" into the search block, select DotNetVault, then click "Install".



Build
Rebuild
Clean
Analyze and Code Cleanup
Pack
Publish...
Run Tests
Debug Tests
Scope to This
New Solution Explorer View
Edit Project File
Add
Manage NuGet Packages...
NuGet Package Manager: DotNetVaultQuickStart
Browse
Installed
Updates
DotNetVault cpsusie
Include prerelease
Package source: nuget.org
DotNetVault by Christopher Susie, 638 downloads v0.1.4-beta
DotNetVault is a library and static code analysis tool that makes managing shared mutable state in multi-threaded applications more manageable and less error prone. Where errors do still occ...
Microsoft.DotNet.PlatformAbstractions by Microsoft, 108M downloads v3.1.1
Abstractions for making code that uses file system and environment testable.
Microsoft.NETCore.DotNetHostPolicy by Microsoft, 96M downloads v3.1.1
Provides a CoreCLR hosting policy implementation -- configuration settings, assembly paths and
DotNetVault
Version: Latest prerelease 0.1.4-beta
Install
Options
Description
DotNetVault is a library and static code analysis tool that makes managing shared mutable state in multi-threaded
Solution Explorer
Solution 'DotNetVaultQuickStart'
DotNetVaultQuickStart
Dependencies
Program.cs

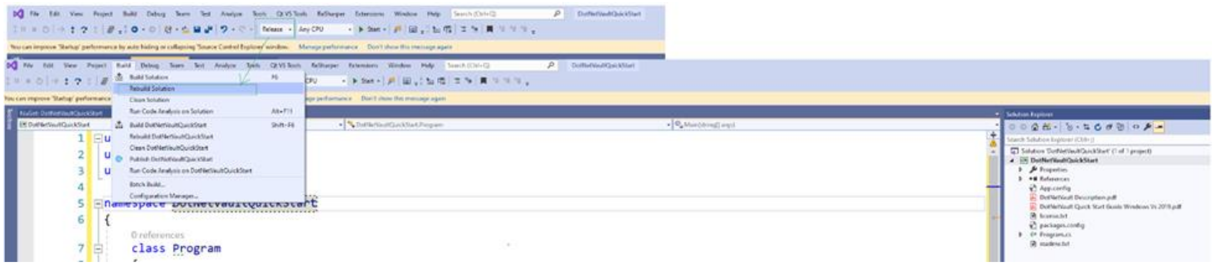
4. You are now set up to use DotNetVault in your project.

5. Test Program Entry

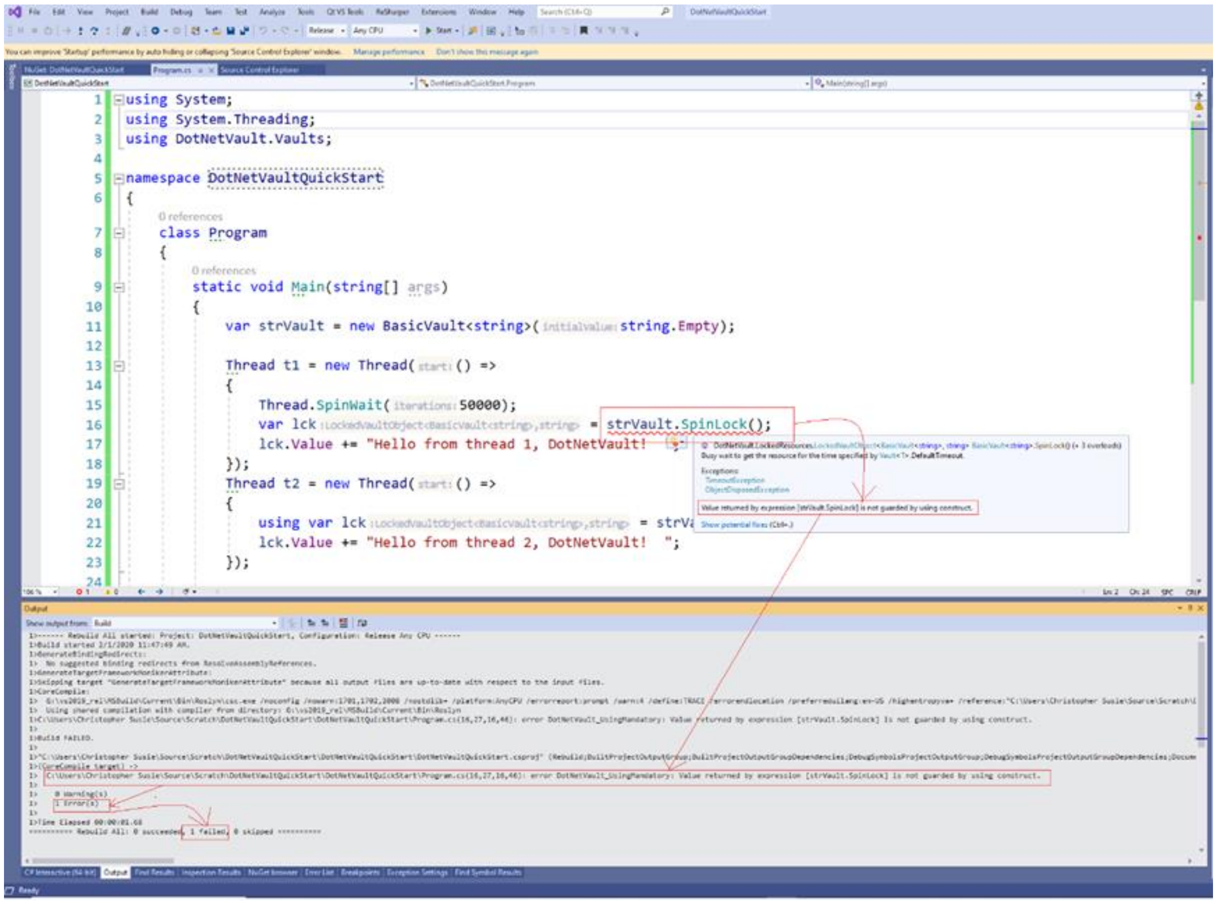
   - To ensure that the static analyzer installed correctly, delete the "Hello World" program entered in by default and replace it with the following code:

```
using System;
using System.Threading;
using DotNetVault.Vaults;

namespace DotNetVaultQuickStart
{
    class Program
    {
        static void Main(string[] args)
        {
            var strVault = new BasicVault<string>(string.Empty);

            Thread t1 = new Thread(() =>
            {
                Thread.SpinWait(50000);
                var lck = strVault.SpinLock();
                lck.Value += "Hello from thread 1, DotNetVault!  ";
            });
            Thread t2 = new Thread(() =>
            {
                using var lck = strVault.SpinLock();
                lck.Value += "Hello from thread 2, DotNetVault!  ";
            });

            t1.Start();
            t2.Start();
            t2.Join();
            t1.Join();

            string finalResult =
strVault.CopyCurrentValue(TimeSpan.FromMilliseconds(100));
            Console.WriteLine(finalResult);
        }
    }
}
```

- Change the Configuration to **Release** and Build the project:


Release
Any CPU
Build Solution
Rebuild Solution
Clean Solution
Run Code Analysis on Solution
Build DotNetVaultQuickStart
Rebuild DotNetVaultQuickStart
Clean DotNetVaultQuickStart
Publish DotNetVaultQuickStart
Run Code Analysis on DotNetVaultQuickStart
Batch Build...
Configuration Manager...

- *If you have installed everything correctly, the build should* **fail**. If it builds, consult Visual Studio documentation for how to enable Roslyn Analyzers. Do not attempt to use this library without static analysis enabled. Assuming it does not build, you should see the following as a result of your build attempt:



```
 1  using System;
 2  using System.Threading;
 3  using DotNetVault.Vaults;
 4
 5  namespace DotNetVaultQuickStart
 6  {
        0 references
 7      class Program
 8      {
            0 references
 9          static void Main(string[] args)
10          {
11              var strVault = new BasicVault<string>(initialValue: string.Empty);
12
13              Thread t1 = new Thread(start: () =>
14              {
15                  Thread.SpinWait(iterations: 50000);
16                  var lck :LockedVaultObject<BasicVault<string>,string> = strVault.SpinLock();
17                  lck.Value += "Hello from thread 1, DotNetVault!
18              });
19              Thread t2 = new Thread(start: () =>
20              {
21                  using var lck :LockedVaultObject<BasicVault<string>,string> = strVa
22                  lck.Value += "Hello from thread 2, DotNetVault!  ";
23              });
24
```

DotNetVault.LockedResources.LockedVaultObject<BasicVault<string>, string> BasicVault<string>.SpinLock() (+ 3 overloads)
Busy wait to get the resource for the time specified by Vault<T>.DefaultTimeout.

Exceptions:
TimeoutException
ObjectDisposedException

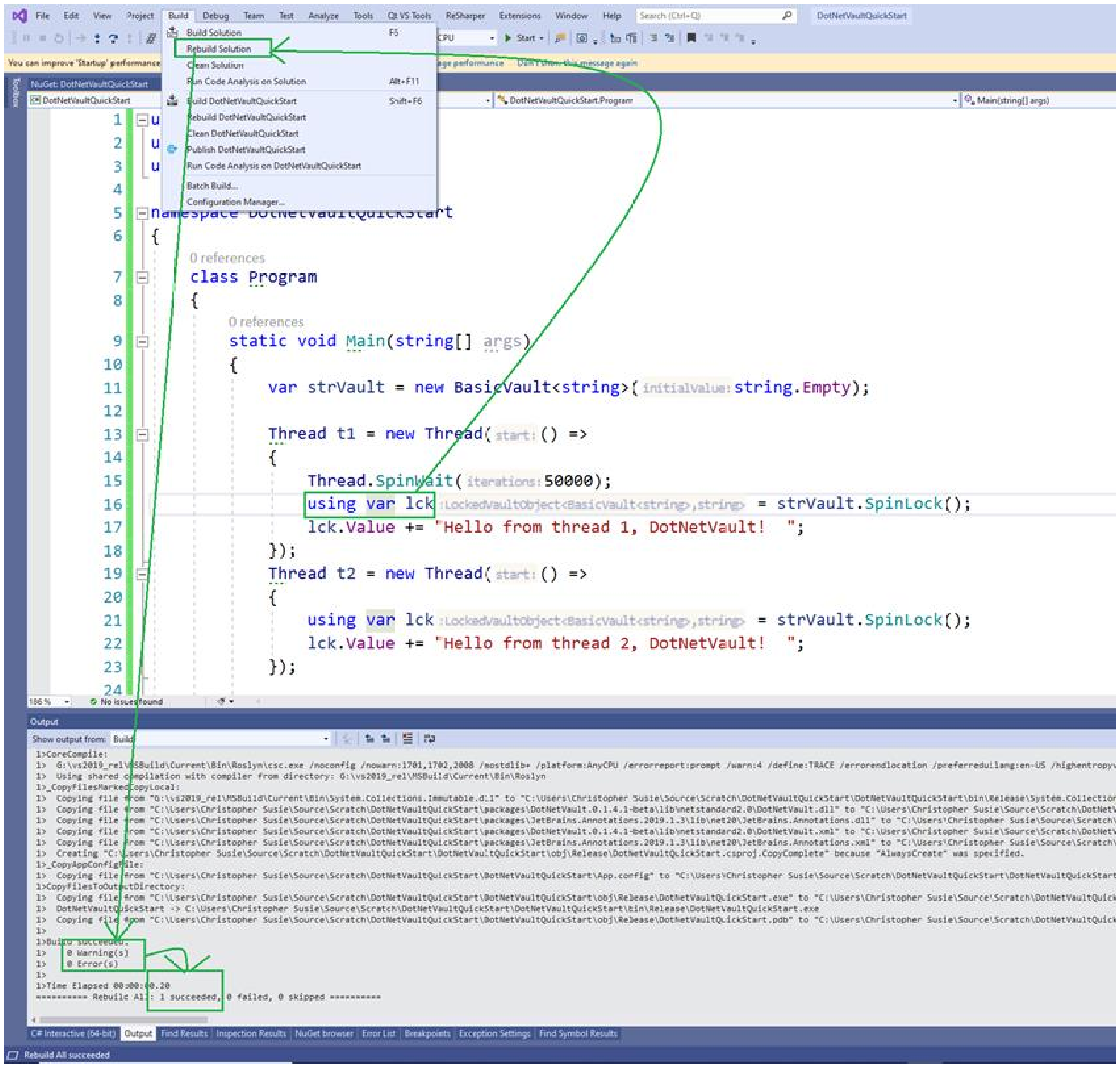
Value returned by expression [strVault.SpinLock] is not guarded by using construct.

Show potential fixes (Ctrl+.)

```
1>------ Rebuild All started: Project: DotNetVaultQuickStart, Configuration: Release Any CPU ------
1>Build started 2/1/2020 11:47:49 AM.
1>GenerateBindingRedirects:
1>  No suggested binding redirects from ResolveAssemblyReferences.
1>GenerateTargetFrameworkMonikerAttribute:
1>Skipping target "GenerateTargetFrameworkMonikerAttribute" because all output files are up-to-date with respect to the input files.
1>CoreCompile:
1>  G:\vs2019_rel\MSBuild\Current\Bin\Roslyn\csc.exe /noconfig /nowarn:1701,1702,2008 /nostdlib+ /platform:AnyCPU /errorreport:prompt /warn:4 /define:TRACE /errorendlocation /preferreduilang:en-US /highentropyva+ /reference:"C:\Users\Christopher Susie\Source\Scratch\
1>  Using shared compilation with compiler from directory: G:\vs2019_rel\MSBuild\Current\Bin\Roslyn
1>C:\Users\Christopher Susie\Source\Scratch\DotNetVaultQuickStart\DotNetVaultQuickStart\Program.cs(16,27,16,46): error DotNetVault_UsingMandatory: Value returned by expression [strVault.SpinLock] is not guarded by using construct.
1>
1>Build FAILED.
1>
1>"C:\Users\Christopher Susie\Source\Scratch\DotNetVaultQuickStart\DotNetVaultQuickStart\DotNetVaultQuickStart.csproj" (Rebuild;BuiltProjectOutputGroup;BuiltProjectOutputGroupDependencies;DebugSymbolsProjectOutputGroup;DebugSymbolsProjectOutputGroupDependencies;Docum
1>(CoreCompile target) ->
1>  C:\Users\Christopher Susie\Source\Scratch\DotNetVaultQuickStart\DotNetVaultQuickStart\Program.cs(16,27,16,46): error DotNetVault_UsingMandatory: Value returned by expression [strVault.SpinLock] is not guarded by using construct.
1>
1>    0 Warning(s)
1>    1 Error(s)
1>
1>Time Elapsed 00:00:01.68
========== Rebuild All: 0 succeeded, 1 failed, 0 skipped ==========
```

- The error is that you must **guard** the return value from a *Lock()* or *SpinLock()* method (or any other method whose return value you choose to annotate with the *UsingMandatory* attribute) with a using statement or declaration. Failure to ensure that the lock is promptly released would cause a serious error in your program … it would timeout whenever in the future you attempted to obtain the lock.

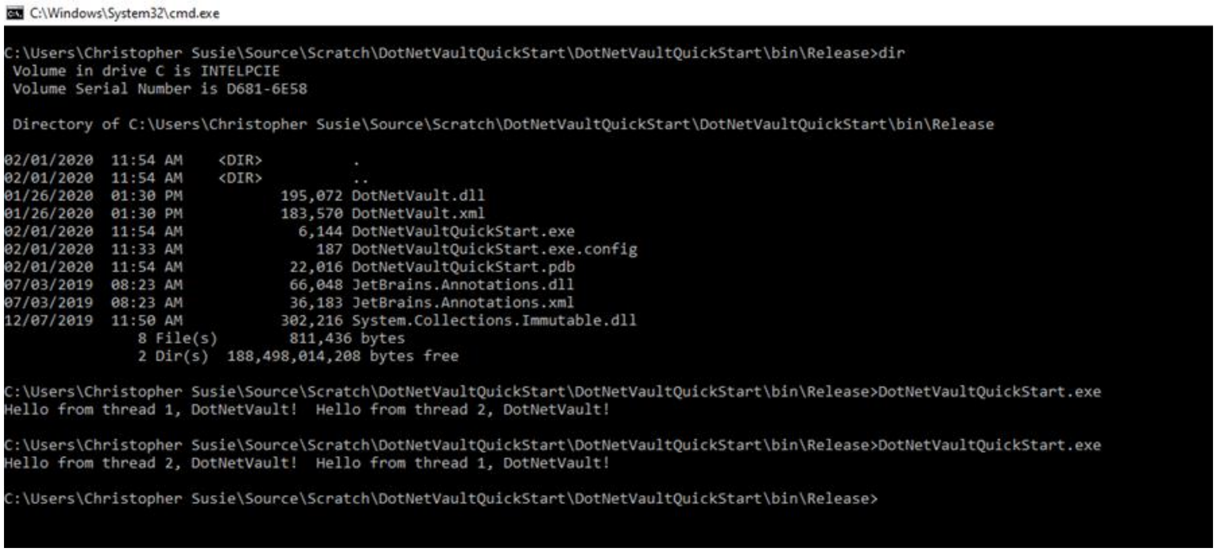
- To fix the error, on line 16, change "var lck =…" to "using var lck =…" as shown then Build again. This time, the build should succeed as shown:



```
u
u
u

namespace DotNetVaultQuickStart
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            var strVault = new BasicVault<string>(initialValue: string.Empty);

            Thread t1 = new Thread(start: () =>
            {
                Thread.SpinWait(iterations: 50000);
                using var lck :LockedVaultObject<BasicVault<string>,string> = strVault.SpinLock();
                lck.Value += "Hello from thread 1, DotNetVault!  ";
            });
            Thread t2 = new Thread(start: () =>
            {
                using var lck :LockedVaultObject<BasicVault<string>,string> = strVault.SpinLock();
                lck.Value += "Hello from thread 2, DotNetVault!  ";
            });
```

```
1>CoreCompile:
1>  G:\vs2019_rel\MSBuild\Current\Bin\Roslyn\csc.exe /noconfig /nowarn:1701,1702,2008 /nostdlib+ /platform:AnyCPU /errorreport:prompt /warn:4 /define:TRACE /errorendlocation /preferreduilang:en-US /highentropyv
1>  Using shared compilation with compiler from directory: G:\vs2019_rel\MSBuild\Current\Bin\Roslyn
1>_CopyFilesMarkedCopyLocal:
1>  Copying file from "G:\vs2019_rel\MSBuild\Current\Bin\System.Collections.Immutable.dll" to "C:\Users\Christopher Susie\Source\Scratch\DotNetVaultQuickStart\DotNetVaultQuickStart\bin\Release\System.Collection
1>  Copying file from "C:\Users\Christopher Susie\Source\Scratch\DotNetVaultQuickStart\packages\DotNetVault.0.1.4.1-beta\lib\netstandard2.0\DotNetVault.dll" to "C:\Users\Christopher Susie\Source\Scratch\DotNetV
1>  Copying file from "C:\Users\Christopher Susie\Source\Scratch\DotNetVaultQuickStart\packages\JetBrains.Annotations.2019.1.3\lib\net20\JetBrains.Annotations.dll" to "C:\Users\Christopher Susie\Source\Scratch\
1>  Copying file from "C:\Users\Christopher Susie\Source\Scratch\DotNetVaultQuickStart\packages\DotNetVault.0.1.4.1-beta\lib\netstandard2.0\DotNetVault.xml" to "C:\Users\Christopher Susie\Source\Scratch\DotNetV
1>  Copying file from "C:\Users\Christopher Susie\Source\Scratch\DotNetVaultQuickStart\packages\JetBrains.Annotations.2019.1.3\lib\net20\JetBrains.Annotations.xml" to "C:\Users\Christopher Susie\Source\Scratch\
1>  Creating "C:\Users\Christopher Susie\Source\Scratch\DotNetVaultQuickStart\DotNetVaultQuickStart\obj\Release\DotNetVaultQuickStart.csproj.CopyComplete" because "AlwaysCreate" was specified.
1>_CopyAppConfigFile:
1>  Copying file from "C:\Users\Christopher Susie\Source\Scratch\DotNetVaultQuickStart\DotNetVaultQuickStart\App.config" to "C:\Users\Christopher Susie\Source\Scratch\DotNetVaultQuickStart\DotNetVaultQuickStart
1>CopyFilesToOutputDirectory:
1>  Copying file from "C:\Users\Christopher Susie\Source\Scratch\DotNetVaultQuickStart\DotNetVaultQuickStart\obj\Release\DotNetVaultQuickStart.exe" to "C:\Users\Christopher Susie\Source\Scratch\DotNetVaultQuick
1>  DotNetVaultQuickStart -> C:\Users\Christopher Susie\Source\Scratch\DotNetVaultQuickStart\DotNetVaultQuickStart\bin\Release\DotNetVaultQuickStart.exe
1>  Copying file from "C:\Users\Christopher Susie\Source\Scratch\DotNetVaultQuickStart\DotNetVaultQuickStart\obj\Release\DotNetVaultQuickStart.pdb" to "C:\Users\Christopher Susie\Source\Scratch\DotNetVaultQuick
1>
1>Build succeeded.
1>     0 Warning(s)
1>     0 Error(s)
1>
1>Time Elapsed 00:00:00.20
========== Rebuild All: 1 succeeded, 0 failed, 0 skipped ==========
```

- Now you should be able to run the application as shown:



```
C:\Users\Christopher Susie\Source\Scratch\DotNetVaultQuickStart\DotNetVaultQuickStart\bin\Release>dir
 Volume in drive C is INTELPCIE
 Volume Serial Number is D681-6E58

 Directory of C:\Users\Christopher Susie\Source\Scratch\DotNetVaultQuickStart\DotNetVaultQuickStart\bin\Release

02/01/2020  11:54 AM    <DIR>          .
02/01/2020  11:54 AM    <DIR>          ..
01/26/2020  01:30 PM           195,072 DotNetVault.dll
01/26/2020  01:30 PM           183,570 DotNetVault.xml
02/01/2020  11:54 AM             6,144 DotNetVaultQuickStart.exe
02/01/2020  11:33 AM               187 DotNetVaultQuickStart.exe.config
02/01/2020  11:54 AM            22,016 DotNetVaultQuickStart.pdb
07/03/2019  08:23 AM            66,048 JetBrains.Annotations.dll
07/03/2019  08:23 AM            36,183 JetBrains.Annotations.xml
12/07/2019  11:50 AM           302,216 System.Collections.Immutable.dll
               8 File(s)        811,436 bytes
               2 Dir(s)  188,498,014,208 bytes free

C:\Users\Christopher Susie\Source\Scratch\DotNetVaultQuickStart\DotNetVaultQuickStart\bin\Release>DotNetVaultQuickStart.exe
Hello from thread 1, DotNetVault!  Hello from thread 2, DotNetVault!

C:\Users\Christopher Susie\Source\Scratch\DotNetVaultQuickStart\DotNetVaultQuickStart\bin\Release>DotNetVaultQuickStart.exe
Hello from thread 2, DotNetVault!  Hello from thread 1, DotNetVault!

C:\Users\Christopher Susie\Source\Scratch\DotNetVaultQuickStart\DotNetVaultQuickStart\bin\Release>
```

6. Congratulations, you have successfully built an application using DotNetVault. Next, we will look at a QuickStart application that shows the very basics of what can be done with this library and its integrated

static analyzer.