Test Driven Development with NUnit for .NET - Getting Started

Boulder Colorado .NET User Group

http://www.BoulderUG.com/

6/27/2006 5:30 PM

Clark Anderson

- Migrated my engineering solution skills from Thermal Design Analysis of AeroSpace Controls
- Design and Development of User Interfaces and Databases.
- Following Test Driven Development, TDD, for several years.
- Primarily: JUnit, VBUnit and NUnit.

Unit Testing Frameworks

- JUnit, from http://www.junit.org, is designed for JAVA Unit Testing.
- VBUnit3, from http://www.vbunit.org, is designed for Visual Basic, Pre .NET. VBUnit3 benefited several VB6 projects.
- NUnit, from http://www.nunit.org, is a unit-testing framework for all of the .NET languages. Learning how to use NUnit with VB 2005.

Getting Started

- Installed Visual Studio 2005.
- Downloaded and installed NUnit-2.2.8 from http://www.nunit.org .
- Created Visual Studio 2005 and Nunit 2.2.8 development environment.
- With a lot of help from the folks at: nunit-users@lists.sourceforge.net https://lists.sourceforge.net/lists/listinfo/nunit-users

VS 2005 - Nunit Environment

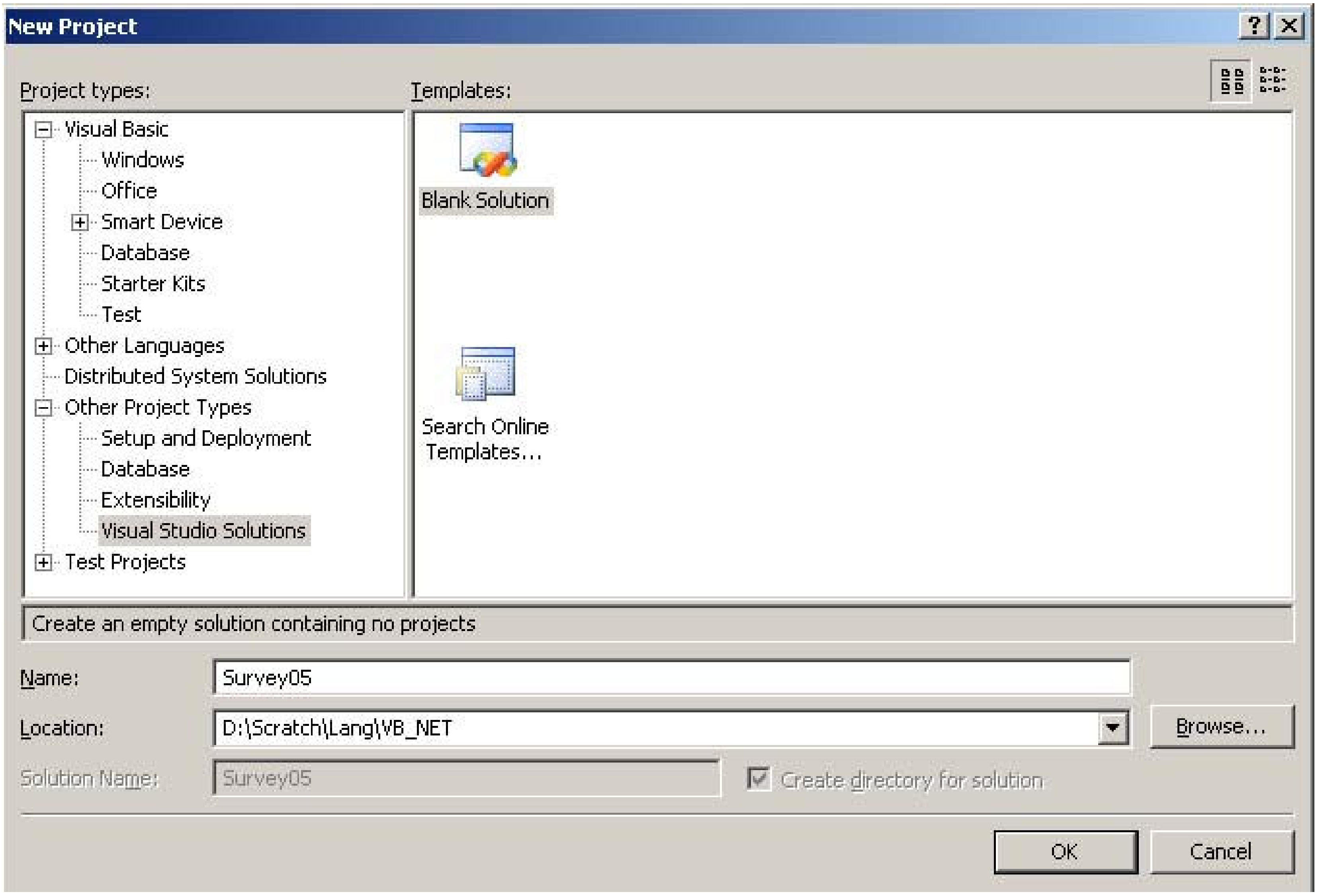
- Create a new Solution in Visual Studio.
- Within solution, create new Project for target product.
- Within solution, create new Class Library for unit testing code.
- Add references between new Projects.
- Add beginning NUnit code.
- Test, Develop, Test (TDD).

Create a new Solution in VS

- Select menu item:

  File/New Project…
- Other Project Types
- Visual studio Solutions
- Blank Solution
- Name: (e.g.: Survey05)
- Location: (e.g.: D:\Scratch\Lang\VB_NET)

Create a new Solution in VS



New Project

Project types:

- Visual Basic
  - Windows
  - Office
  - Smart Device
  - Database
  - Starter Kits
  - Test
- Other Languages
- Distributed System Solutions
- Other Project Types
  - Setup and Deployment
  - Database
  - Extensibility
  - Visual Studio Solutions
- Test Projects

Templates:

Create an empty solution containing no projects

Name:	Survey05
Location:	D:\Scratch\Lang\VB_NET
Solution Name:	Survey05

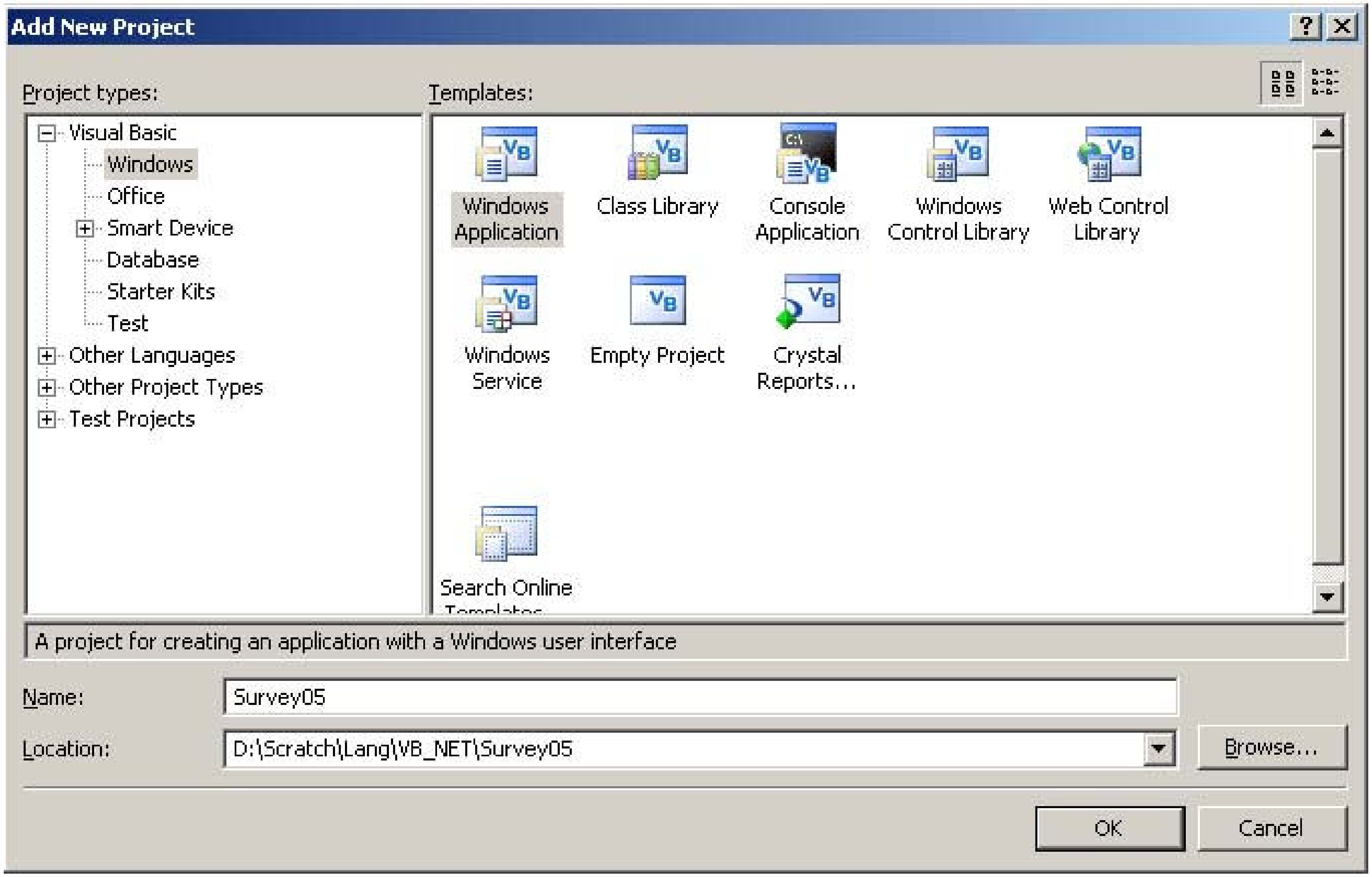
[✓] Create directory for solution

•Within solution, create new Project for your target product.

- Select menu item:

  File/**Add**/New Project...
- Visual Basic (or Visual C#)
- Windows
- Windows Application Template
- Name: Survey05
- Location: D:\Scratch\Lang\VB_NET\Survey05

Within solution, create new Project for your target product.



A project for creating an application with a Windows user interface

Name: Survey05

Location: D:\Scratch\Lang\VB_NET\Survey05

•In the product/target Project, add a code Module:

With the product/target Project selected in the solution Browser,

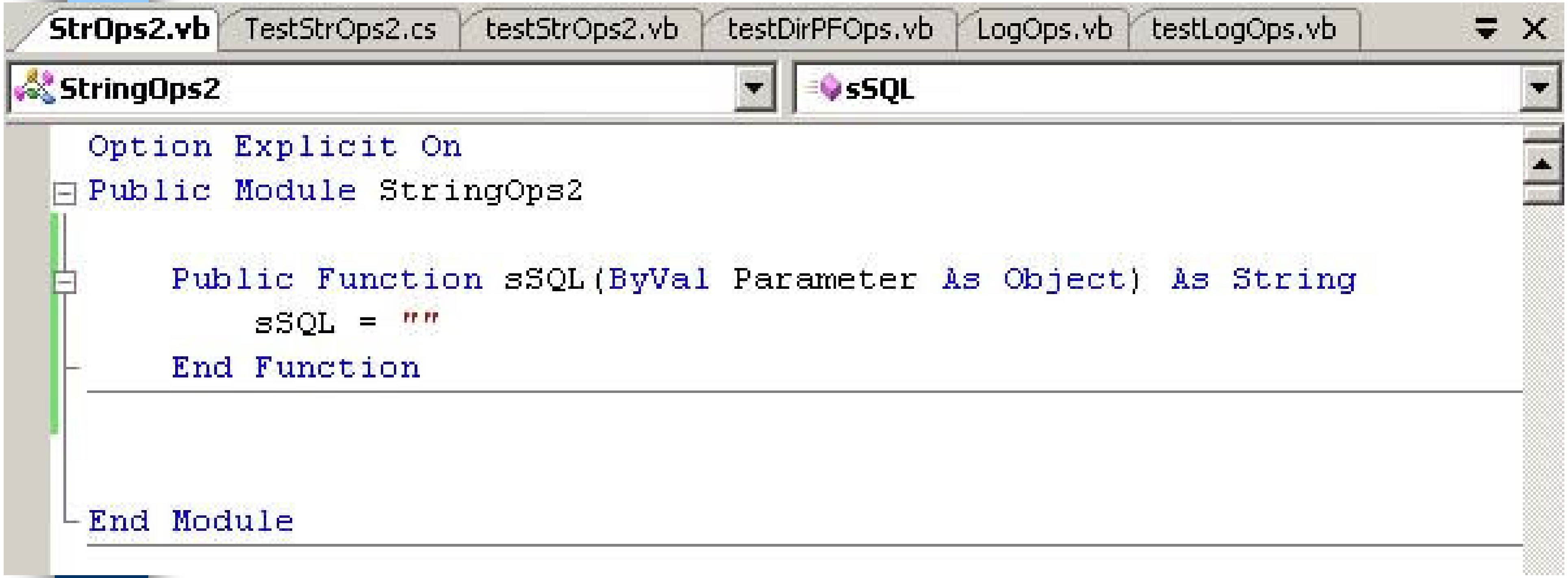
- Select menu item:

  Project/Add Module...
- Module Template
- Name: StrOps2
- Click [OK].

In code Module view/edit window:

Insert some Target/Product code(VB)!

(The entire Module must be Public.)



```
Option Explicit On
Public Module StringOps2

    Public Function sSQL(ByVal Parameter As Object) As String
        sSQL = ""
    End Function


End Module
```

In the product/target Project, set Build output path:

- Select menu item:

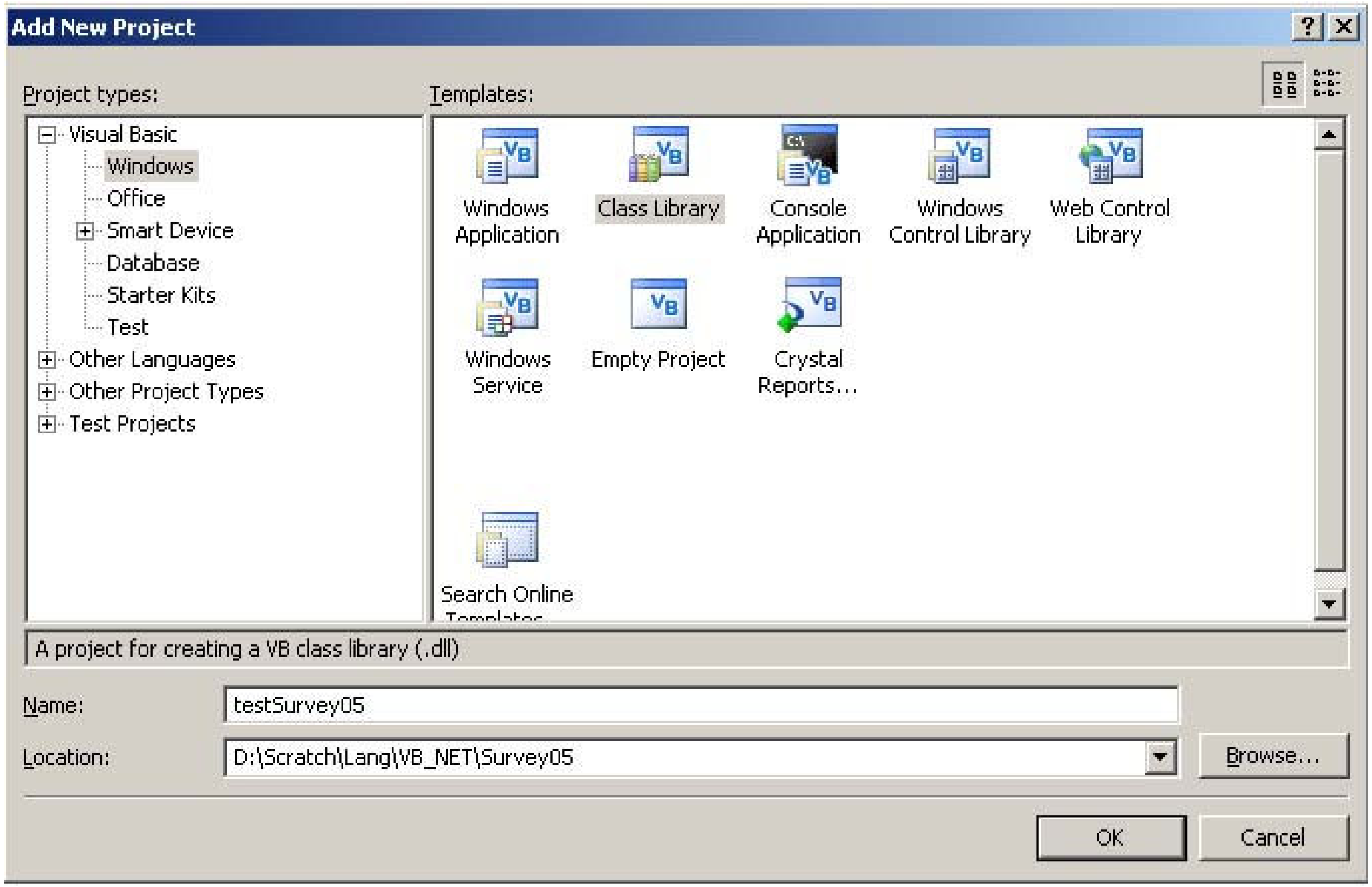
  Project/Survey05 Properties…
- Compile tab
- Build output path:
- Click [Browse…]
- Navigate back to the bin folder and select bin\Debug.
- Select Menu item: File / Save All

Within solution, create new Class Library for unit testing code.

- Select menu item:

  File/**Add**/New Project...
- Visual Basic (or Visual C#)
- Windows
- Class Library Template
- Name: testSurvey05
- Location: D:\Scratch\Lang\VB_NET\Survey05

Within solution, create new Class Library for unit testing code.



Add New Project

Project types:

- Visual Basic
  - Windows
  - Office
  - Smart Device
  - Database
  - Starter Kits
  - Test
- Other Languages
- Other Project Types
- Test Projects

Templates:

Windows Application, Class Library, Console Application, Windows Control Library, Web Control Library, Windows Service, Empty Project, Crystal Reports..., Search Online Templates

A project for creating a VB class library (.dll)

Name: testSurvey05

Location: D:\Scratch\Lang\VB_NET\Survey05

Browse...

OK Cancel

Add references between the new Projects: nunit.framework

- With the new test ClassLibrary selected in the solution Browser,
- Select menu item:

  Project/Add Reference…
- Click .NET tab
- Search for and Select nunit.framework
- Click [OK].

Add references between the new Projects: target/product project

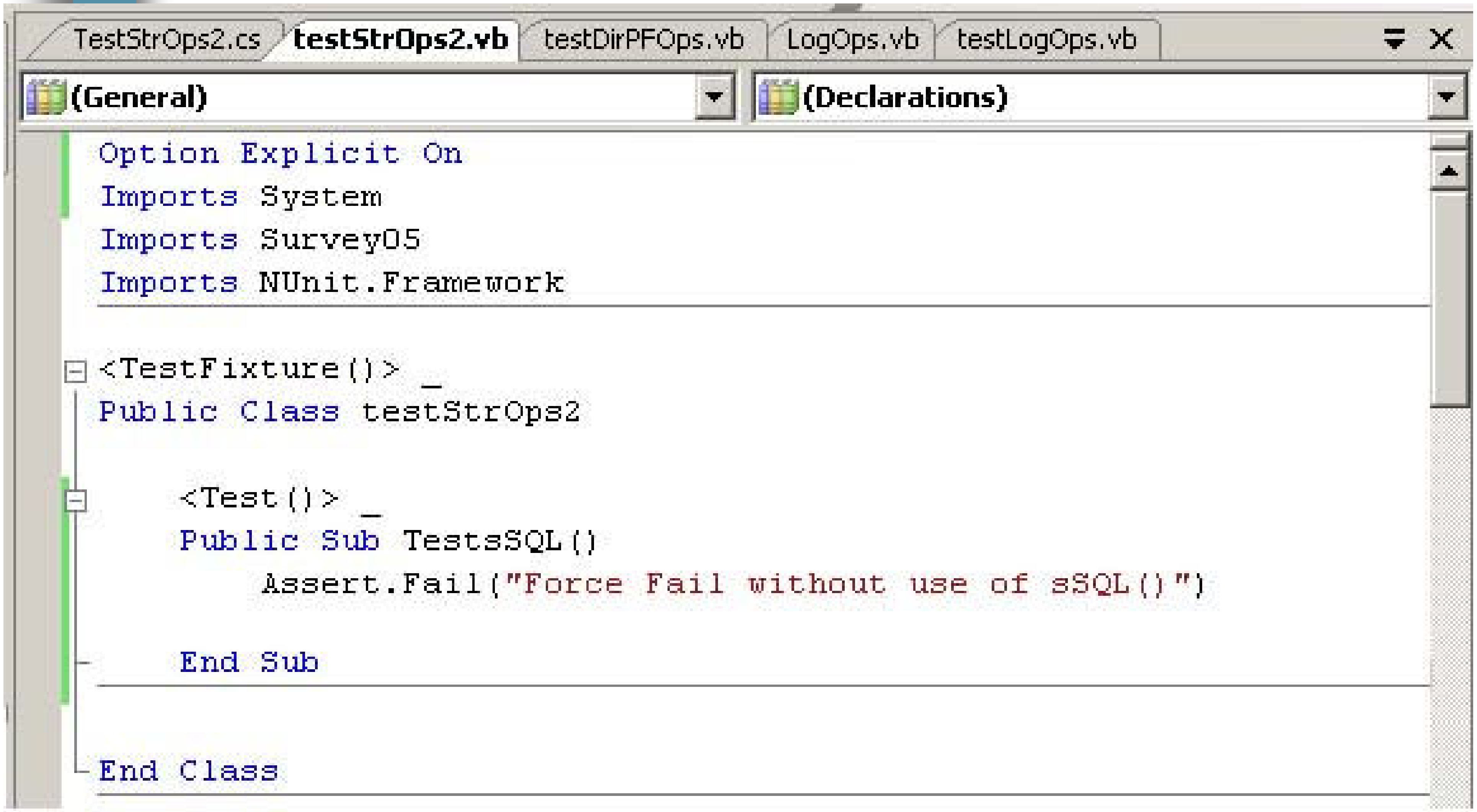
- With the new test ClassLibrary selected in the solution Browser,
- Select menu item:

  Project/Add Reference…
- Click Projects tab
- Select Survey05

  D:\Scratch\Lang\VB_NET\Survey05\Survey05
- Click [OK].

In Class Library view/edit window (VB):

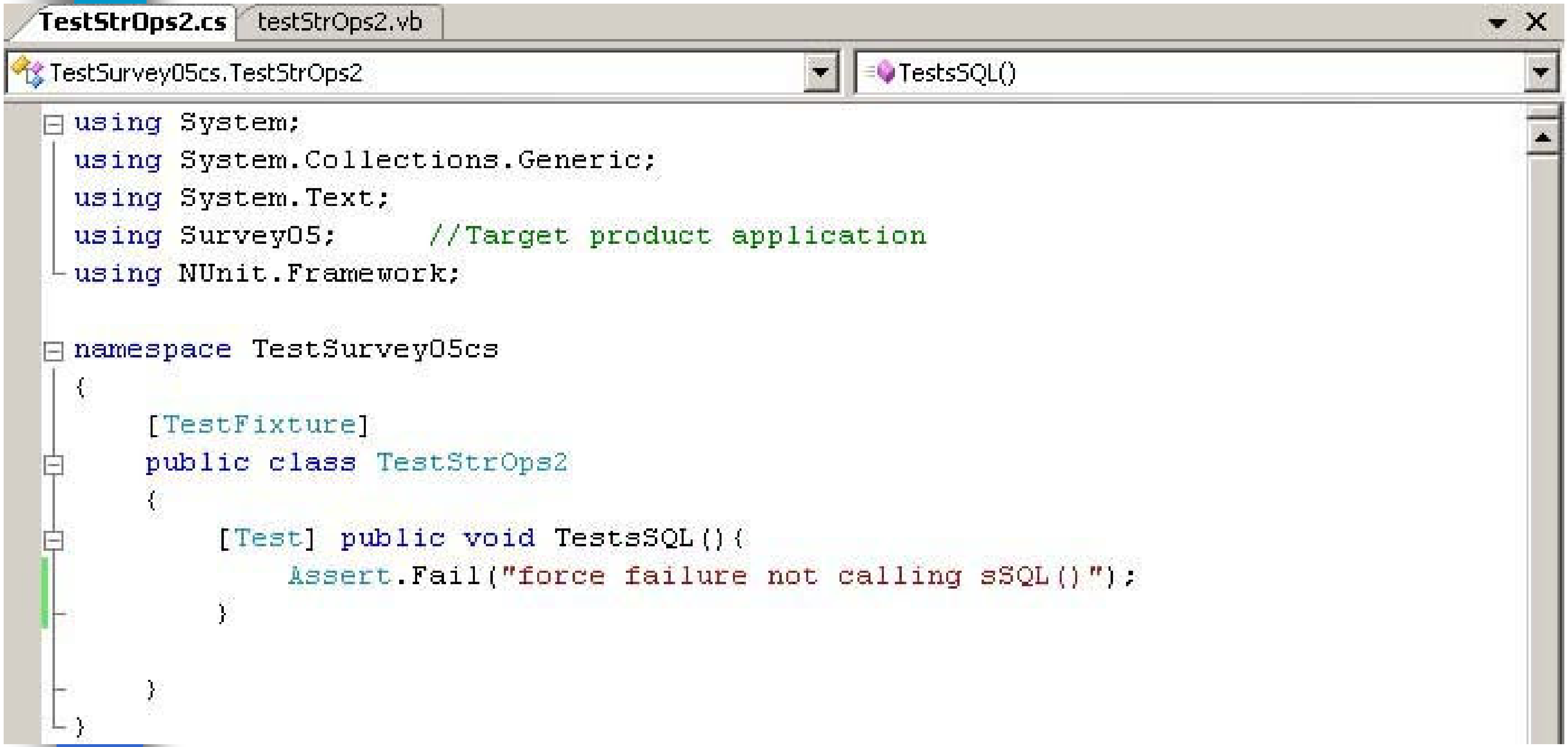
Insert some test code!



```
Option Explicit On
Imports System
Imports Survey05
Imports NUnit.Framework

<TestFixture()> _
Public Class testStrOps2

    <Test()> _
    Public Sub TestsSQL()
        Assert.Fail("Force Fail without use of sSQL()")

    End Sub


End Class
```

In Class Library view/edit window (C#):

Insert some test code!



```
using System;
using System.Collections.Generic;
using System.Text;
using Survey05;      //Target product application
using NUnit.Framework;

namespace TestSurvey05cs
{
    [TestFixture]
    public class TestStrOps2
    {
        [Test] public void TestsSQL(){
            Assert.Fail("force failure not calling sSQL()");
        }

    }
}
```

In the Class Library, set Build output path:

- Select menu item:

  Project/testSurvey05 Properties…
- Compile tab
- Build output path:
- Click [Browse…]
- Navigate back to the bin folder and select: bin\Debug.
- Select Menu item: File / Save All

We are finally going to Run NUnit!

- Select menu item: Build\Build Solution
- Select menu item: Tools\Nunit

The NUnit-GUI form should open!

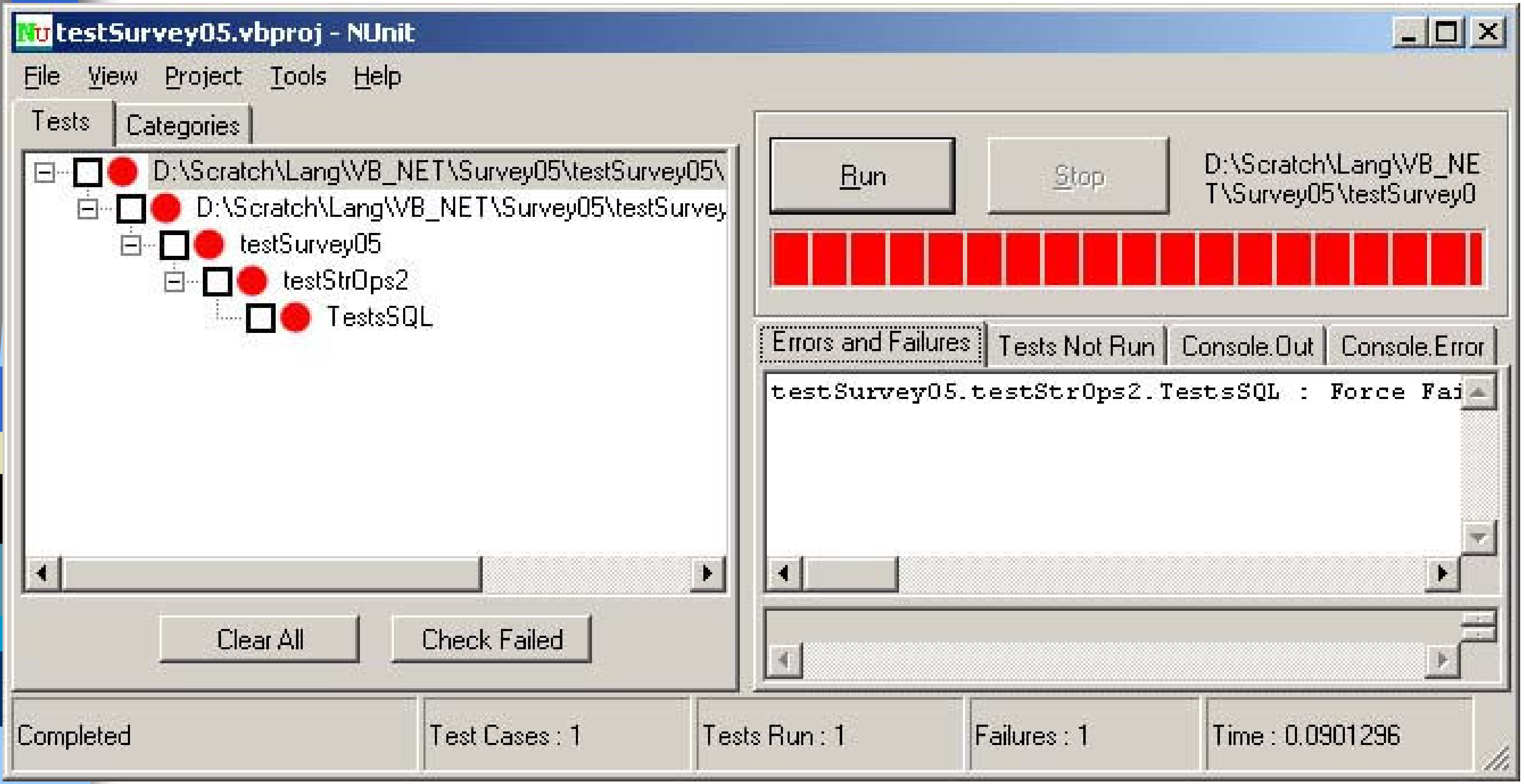
(Nunit menu item: Tools/Options/Test: [x] Enable VS support)

- Select NUnit menu item: File\Open…
- Navigate to D:\Scratch\Lang\VB_NET\Survey05\testSurvey05\testSurvey05.vbproj

The Tests tree should populate!

- Click [Run]

Nunit Results: Assert.Fail()

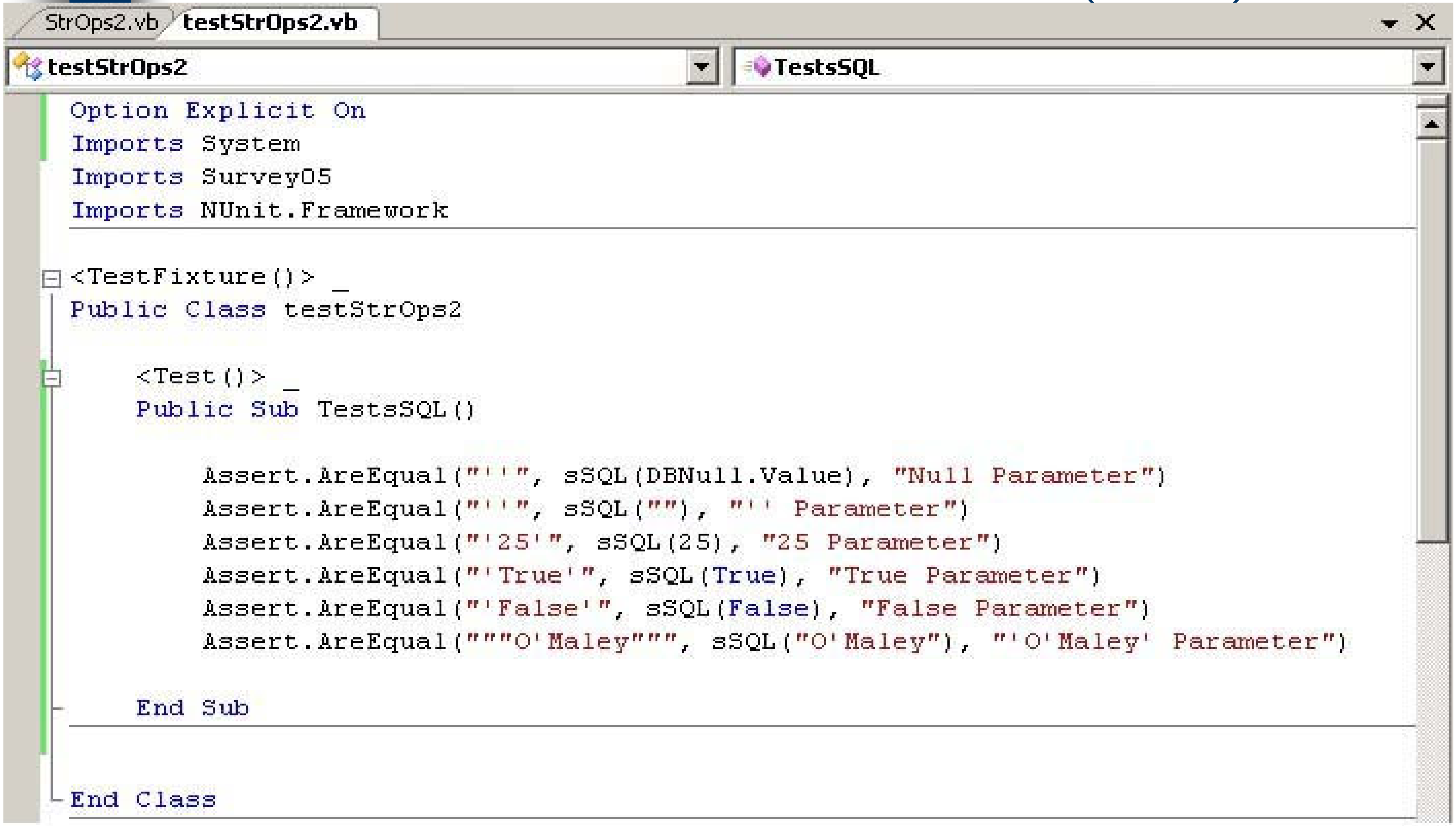

testSurvey05.vbproj - NUnit
File View Project Tools Help
Tests Categories
D:\Scratch\Lang\VB_NET\Survey05\testSurvey05\
D:\Scratch\Lang\VB_NET\Survey05\testSurvey
testSurvey05
testStrOps2
TestsSQL
Run
Stop
D:\Scratch\Lang\VB_NET\Survey05\testSurvey0
Errors and Failures
Tests Not Run
Console.Out
Console.Error
Clear All
Check Failed

```
testSurvey05.testStrOps2.TestsSQL : Force Fai
```

Completed | Test Cases : 1 | Tests Run : 1 | Failures : 1 | Time : 0.0901296

We know our test is working! The one line of test code causes the RED BAR.

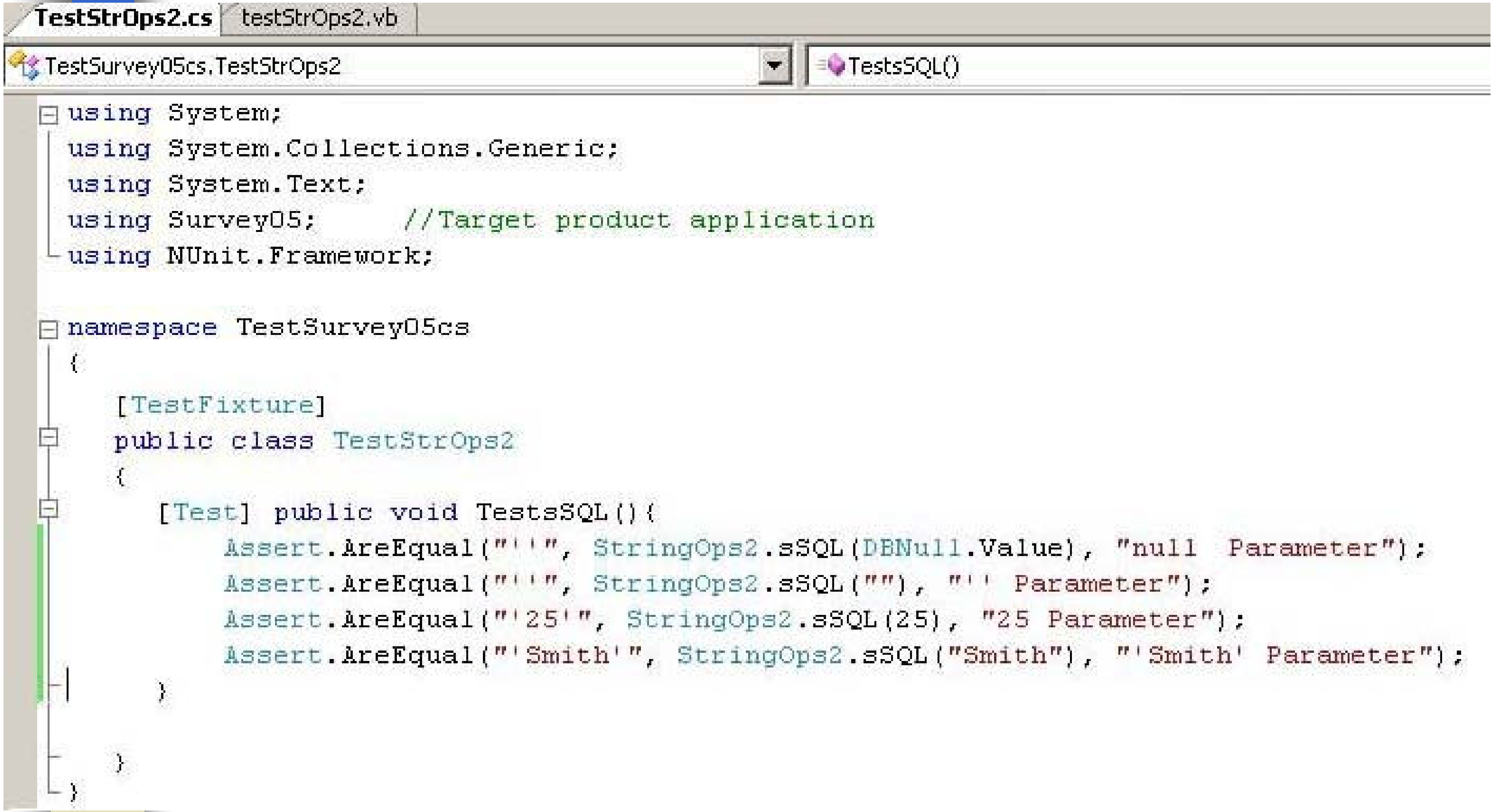
Add some real test code (VB):



```
Option Explicit On
Imports System
Imports Survey05
Imports NUnit.Framework

<TestFixture()> _
Public Class testStrOps2

    <Test()> _
    Public Sub TestsSQL()

        Assert.AreEqual("''", sSQL(DBNull.Value), "Null Parameter")
        Assert.AreEqual("''", sSQL(""), "'' Parameter")
        Assert.AreEqual("'25'", sSQL(25), "25 Parameter")
        Assert.AreEqual("'True'", sSQL(True), "True Parameter")
        Assert.AreEqual("'False'", sSQL(False), "False Parameter")
        Assert.AreEqual("""O'Maley""", sSQL("O'Maley"), "'O'Maley' Parameter")

    End Sub

End Class
```

File/Save All   Build/Rebuild Solution

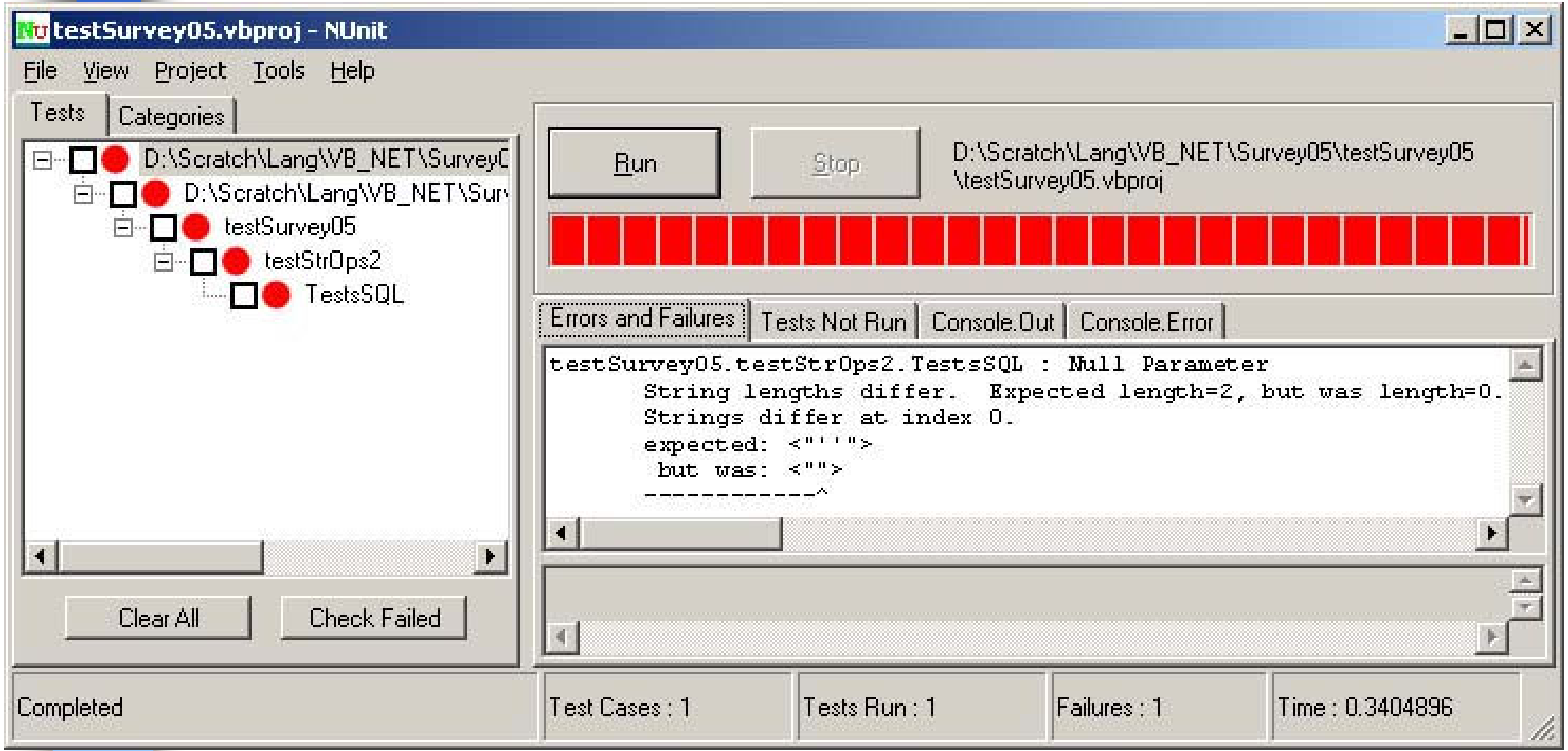
Add some real test code (C#):



```
using System;
using System.Collections.Generic;
using System.Text;
using Survey05;      //Target product application
using NUnit.Framework;

namespace TestSurvey05cs
{
    [TestFixture]
    public class TestStrOps2
    {
        [Test] public void TestsSQL(){
            Assert.AreEqual("''", StringOps2.sSQL(DBNull.Value), "null  Parameter");
            Assert.AreEqual("''", StringOps2.sSQL(""), "'' Parameter");
            Assert.AreEqual("'25'", StringOps2.sSQL(25), "25 Parameter");
            Assert.AreEqual("'Smith'", StringOps2.sSQL("Smith"), "'Smith' Parameter");
        }

    }
}
```

File/Save All   Build/Rebuild Solution

Nunit Results: Assert. ...

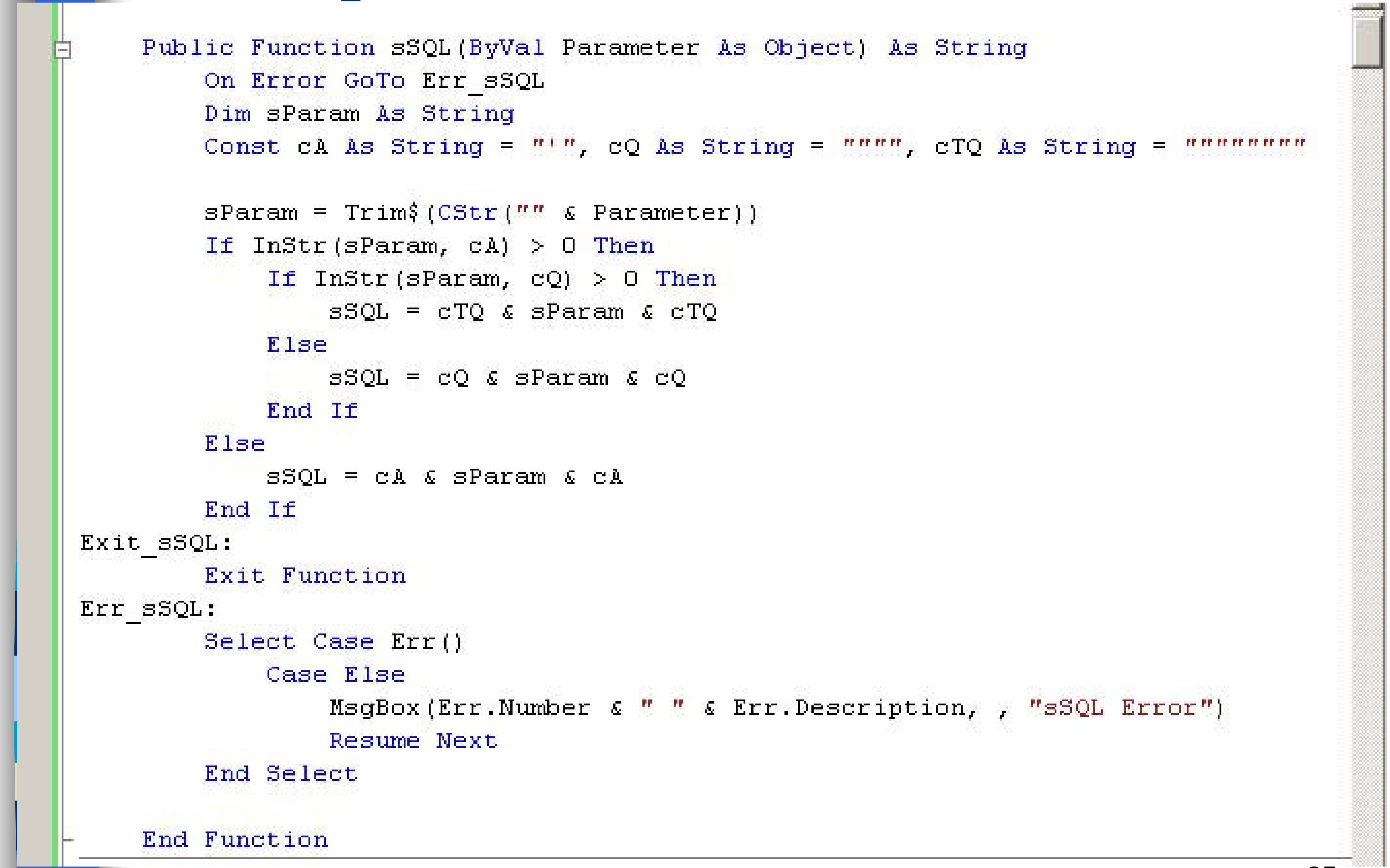

testSurvey05.vbproj - NUnit
File View Project Tools Help
Tests
Categories
D:\Scratch\Lang\VB_NET\SurveyC
D:\Scratch\Lang\VB_NET\Surv
testSurvey05
testStrOps2
TestsSQL
Run
Stop
D:\Scratch\Lang\VB_NET\Survey05\testSurvey05\testSurvey05.vbproj
Errors and Failures
Tests Not Run
Console.Out
Console.Error
Clear All
Check Failed

```
testSurvey05.testStrOps2.TestsSQL : Null Parameter
       String lengths differ.  Expected length=2, but was length=0.
       Strings differ at index 0.
       expected: <"''">
        but was: <"">
       ------------^
```

Completed | Test Cases : 1 | Tests Run : 1 | Failures : 1 | Time : 0.3404896

Oh Well! The test fails on the first Assert. The series of Assert statements are to Drive our Development.

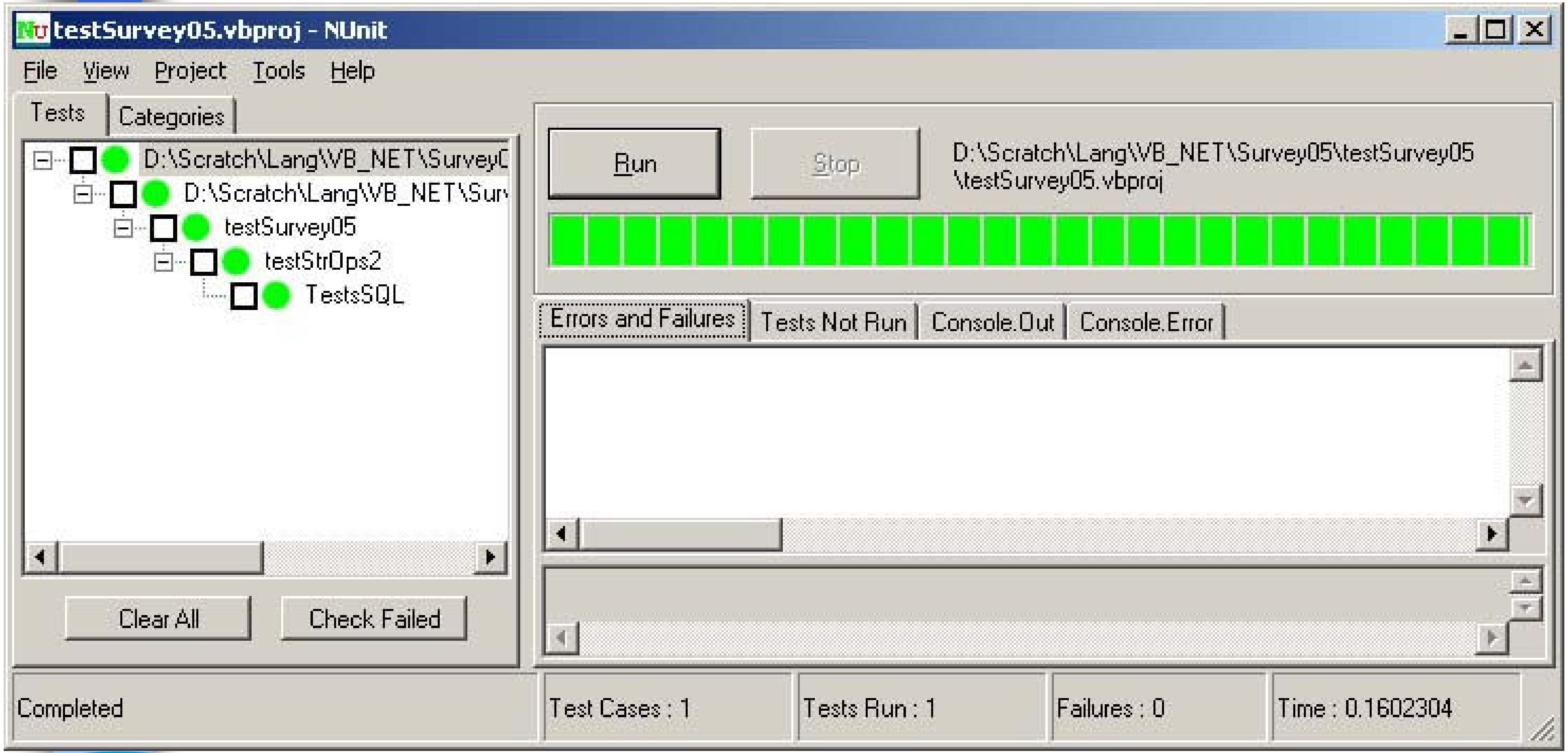
Develop Product code(VB) to Pass:



```
    Public Function sSQL(ByVal Parameter As Object) As String
        On Error GoTo Err_sSQL
        Dim sParam As String
        Const cA As String = "'", cQ As String = """", cTQ As String = """"""""

        sParam = Trim$(CStr("" & Parameter))
        If InStr(sParam, cA) > 0 Then
            If InStr(sParam, cQ) > 0 Then
                sSQL = cTQ & sParam & cTQ
            Else
                sSQL = cQ & sParam & cQ
            End If
        Else
            sSQL = cA & sParam & cA
        End If
Exit_sSQL:
        Exit Function
Err_sSQL:
        Select Case Err()
            Case Else
                MsgBox(Err.Number & " " & Err.Description, , "sSQL Error")
                Resume Next
        End Select

    End Function
```

File/Save All Build/Rebuild Solution

Nunit Results: Pass!


testSurvey05.vbproj - NUnit
File View Project Tools Help
Tests Categories
Run Stop
Errors and Failures Tests Not Run Console.Out Console.Error
Clear All Check Failed

D:\Scratch\Lang\VB_NET\Survey0
D:\Scratch\Lang\VB_NET\Surv
testSurvey05
testStrOps2
TestsSQL

D:\Scratch\Lang\VB_NET\Survey05\testSurvey05
\testSurvey05.vbproj

Completed | Test Cases : 1 | Tests Run : 1 | Failures : 0 | Time : 0.1602304

This is a tiny example of Test Driven Development.

Nunit - Tip of the Iceberg!

- There is a growing variety of Assert Methods:

  .AreEqual() .AreNotEqual()
  .AreNotSame() .AreSame().Contains()
  .Fail().Greater() .Ignore()
  .IsAssignableFrom() .IsEmpty()
  .IsFalse() .IsInstanceOfType()
  .IsNaN() .IsNotAssignableFrom()
  .IsNotEmpty() .IsNotInstanceOfType()
  .IsNotNull() .IsNull() .IsTrue() .Less()
  .ReferenceEquals()

Nunit - Benefits!

- As your product development continues, NUnit can immediately flag code if it ‘breaks’.
- Creating NUnit test
  - Before development is ideal!
  - Before major changes is quite beneficial.
  - Any time for critical code blocks can be lifesaver!

Clark Anderson

Computer Programmer - Analytical Engineer

Connecting People to their Data

User Interface Design and Development

Database Design and Development

Data Conversions and Reports

anderci@indra.com

http://www.indra.com/~anderci

http://www.nunit.org

http://www.testdriven.net/ (.NET 2.0 Beta)
(Looks promising for better integration!)

https://lists.sourceforge.net/lists/listinfo/nunit-users