NestedFlow Automation – Parallel Testing

Creation Date: 9/18/2023

Table Of Contents

Introduction

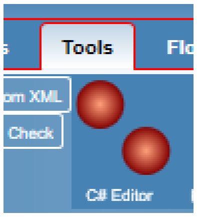
This document explains the parallel testing methods that can be done using the NestedFlow automation tool. Tool allows both sequential and parallel execution of test cases and allows parallel execution of C# code.

Parallel Execution Of C# code

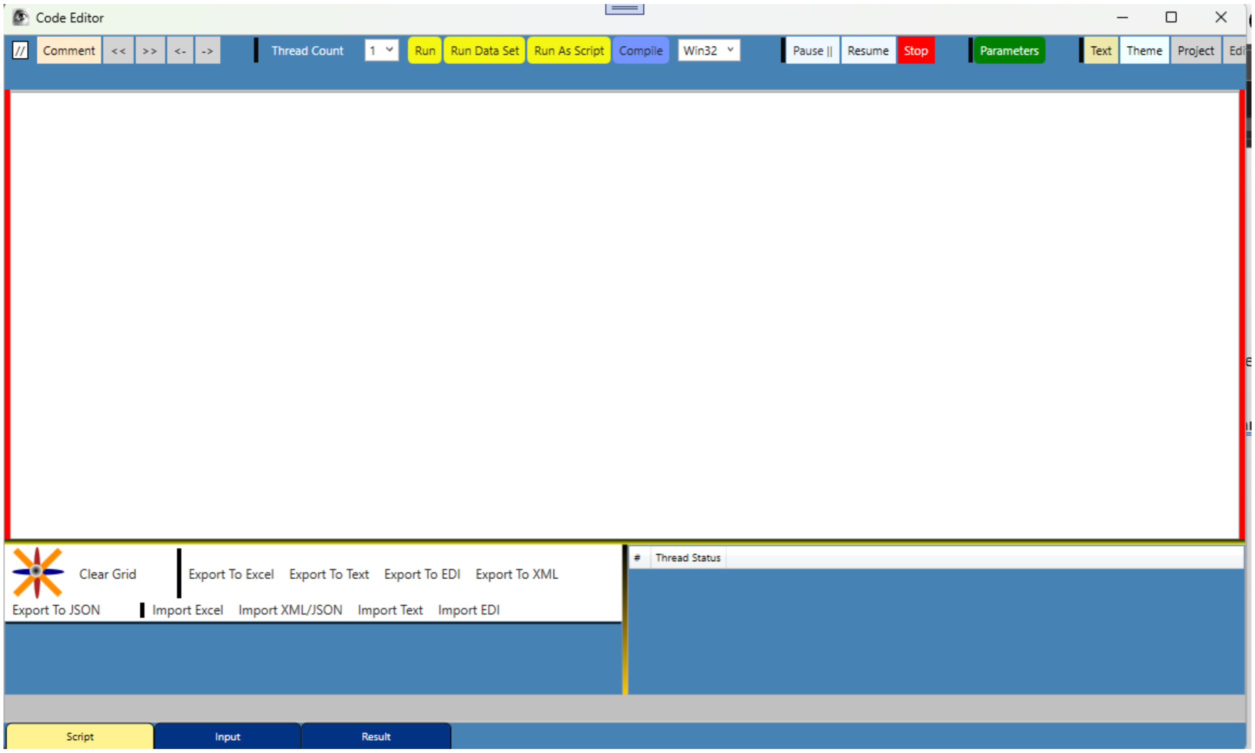
NestedFlow Automation tool allows parallel execution of c# code which can be used to execute time consuming tasks which can be done through code (e.g.: Push of lot of data using APIs)

In this below example we will demonstrate a simple HelloWorld example in single and multi thread mode.

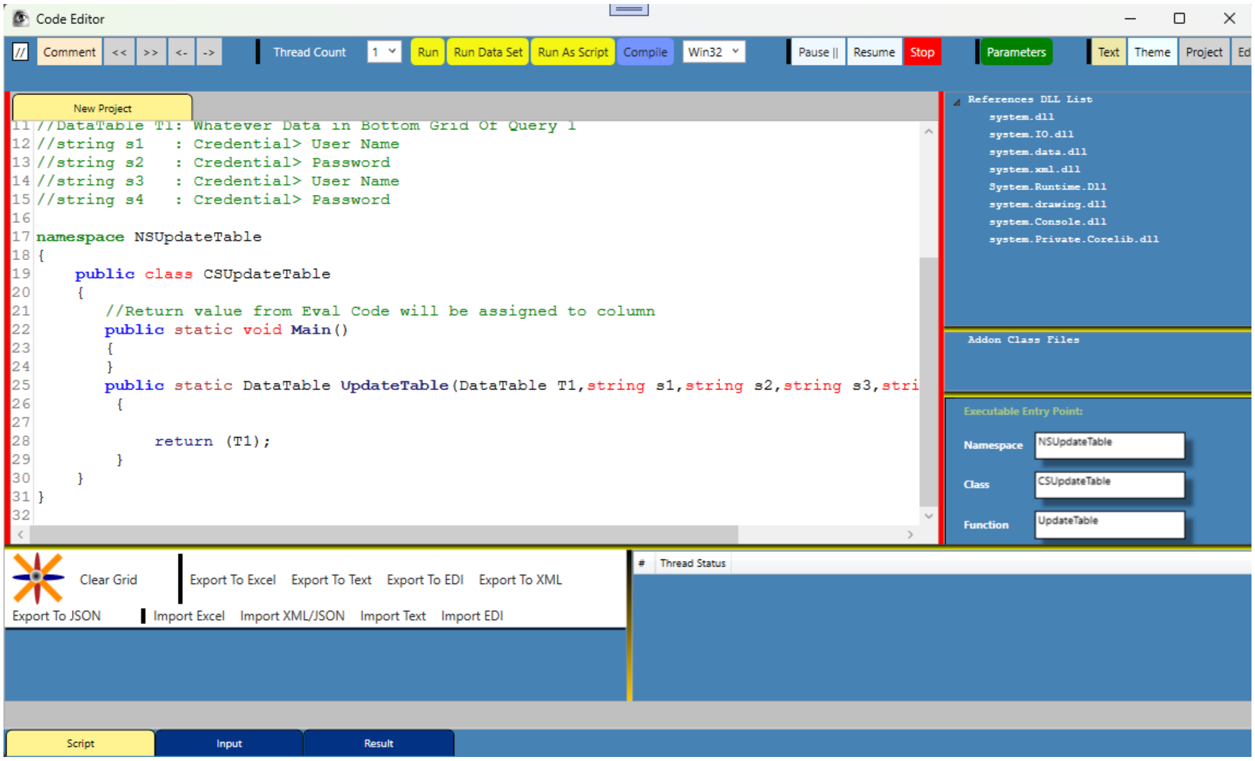
Click on C# Editor menu in the Tools ribbon



Code Editor will be opened:



Click on **Project → New C# Project** A barebone C# program structure will be created:



```
//DataTable T1: Whatever Data in Bottom Grid Of Query 1
//string s1    : Credential> User Name
//string s2    : Credential> Password
//string s3    : Credential> User Name
//string s4    : Credential> Password

namespace NSUpdateTable
{
    public class CSUpdateTable
    {
       //Return value from Eval Code will be assigned to column
       public static void Main()
       {
       }
       public static DataTable UpdateTable(DataTable T1,string s1,string s2,string s3,stri
        {

            return (T1);
        }
    }
}
```

References DLL List
- system.dll
- system.IO.dll
- system.data.dll
- system.xml.dll
- System.Runtime.Dll
- system.drawing.dll
- system.Console.dll
- system.Private.Corelib.dll

Addon Class Files

Executable Entry Point:

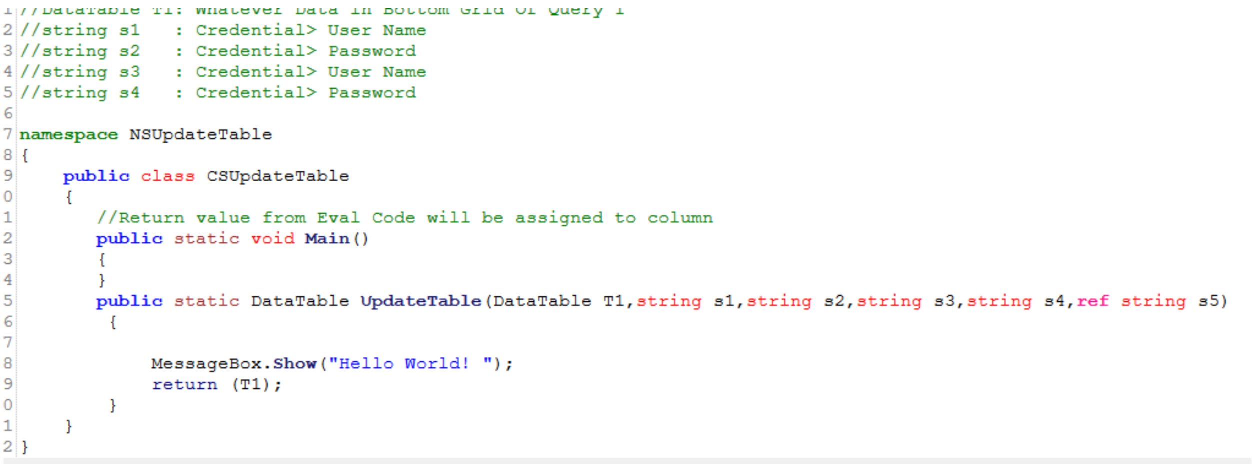
Namespace: NSUpdateTable

Class: CSUpdateTable

Function: UpdateTable

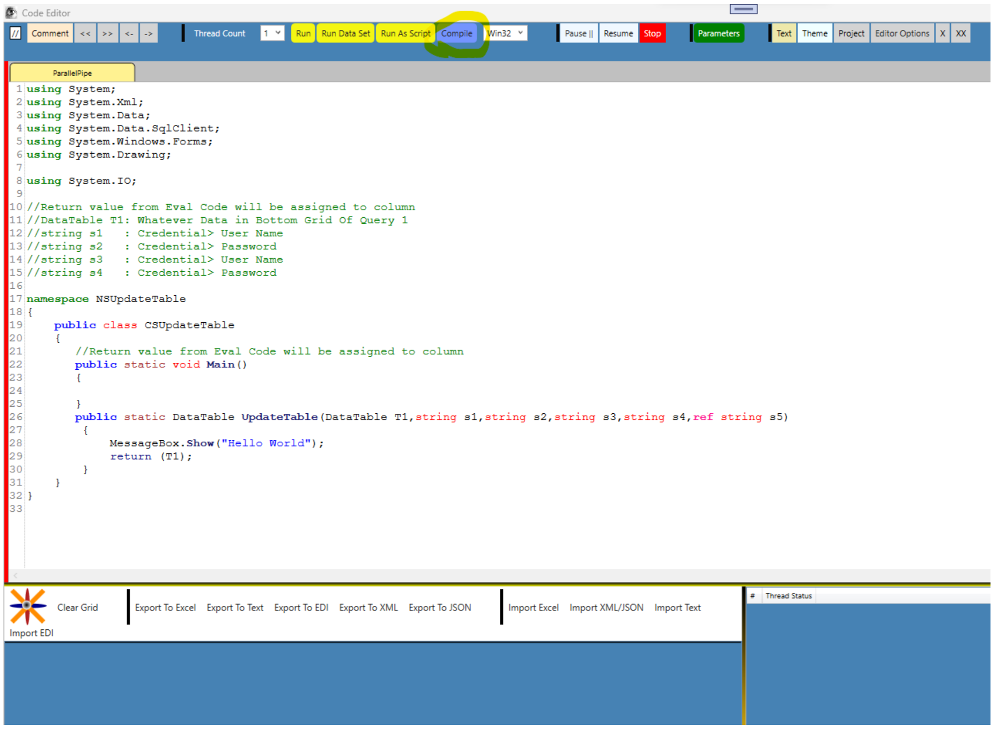
Look at the NameSpace, Class and Function which will be executed on the right-hand side auto collapsing panel. These will be pre populated according to the auto generated code. If you update the code, these values should be updated manually. Also (default libraries are already added. You can extend the code by adding more libraries)

We have completed a simple code as below:



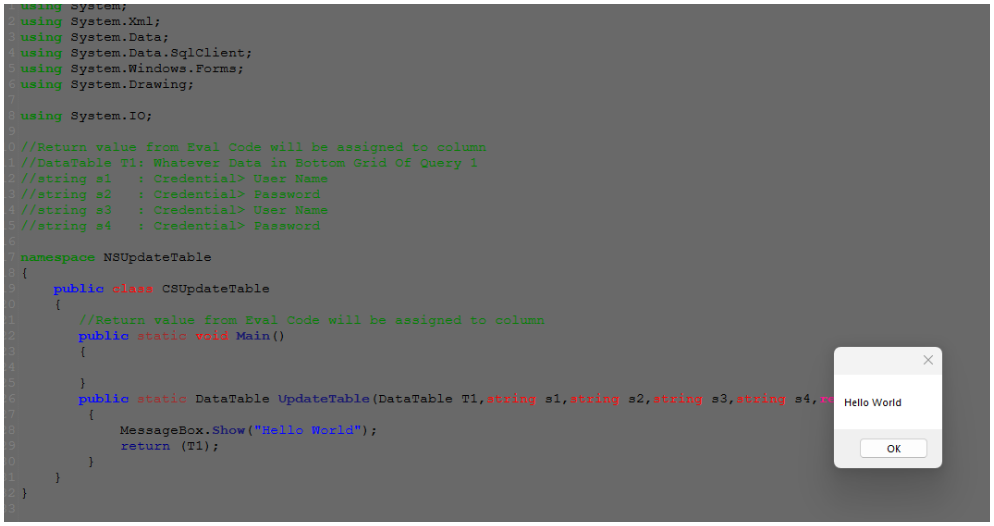
```
//DataTable T1: Whatever Data in Bottom Grid Of Query 1
//string s1    : Credential> User Name
//string s2    : Credential> Password
//string s3    : Credential> User Name
//string s4    : Credential> Password

namespace NSUpdateTable
{
    public class CSUpdateTable
    {
       //Return value from Eval Code will be assigned to column
       public static void Main()
       {
       }
       public static DataTable UpdateTable(DataTable T1,string s1,string s2,string s3,string s4,ref string s5)
        {

            MessageBox.Show("Hello World! ");
            return (T1);
        }
    }
}
```

Compile the code and make sure no errors (if errors exist please resolve them)



```
using System;
using System.Xml;
using System.Data;
using System.Data.SqlClient;
using System.Windows.Forms;
using System.Drawing;

using System.IO;

//Return value from Eval Code will be assigned to column
//DataTable T1: Whatever Data in Bottom Grid Of Query 1
//string s1   : Credential> User Name
//string s2   : Credential> Password
//string s3   : Credential> User Name
//string s4   : Credential> Password

namespace NSUpdateTable
{
    public class CSUpdateTable
    {
       //Return value from Eval Code will be assigned to column
       public static void Main()
       {

       }
       public static DataTable UpdateTable(DataTable T1,string s1,string s2,string s3,string s4,ref string s5)
        {
            MessageBox.Show("Hello World");
            return (T1);
        }
    }
}
```

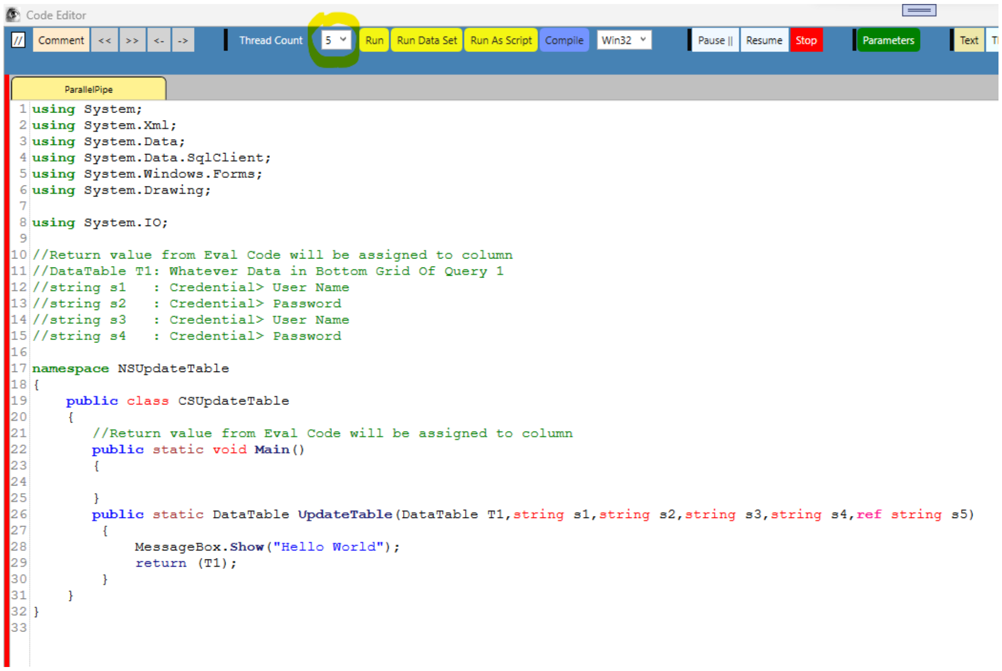
Click on Run button



```
using System;
using System.Xml;
using System.Data;
using System.Data.SqlClient;
using System.Windows.Forms;
using System.Drawing;

using System.IO;

//Return value from Eval Code will be assigned to column
//DataTable T1: Whatever Data in Bottom Grid Of Query 1
//string s1   : Credential> User Name
//string s2   : Credential> Password
//string s3   : Credential> User Name
//string s4   : Credential> Password

namespace NSUpdateTable
{
    public class CSUpdateTable
    {
       //Return value from Eval Code will be assigned to column
       public static void Main()
       {

       }
       public static DataTable UpdateTable(DataTable T1,string s1,string s2,string s3,string s4,
        {
            MessageBox.Show("Hello World");
            return (T1);
        }
    }
}
```

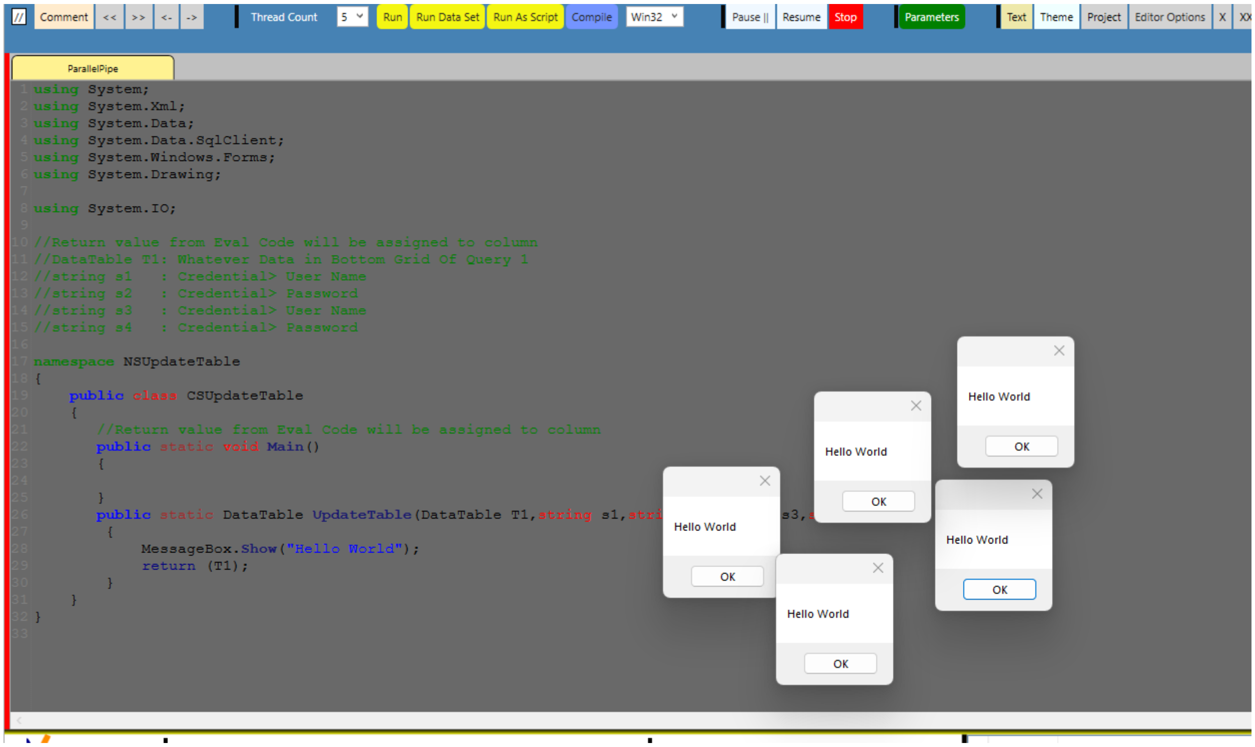
Hello World

Now make the Thread count =5 and click on **Run** button



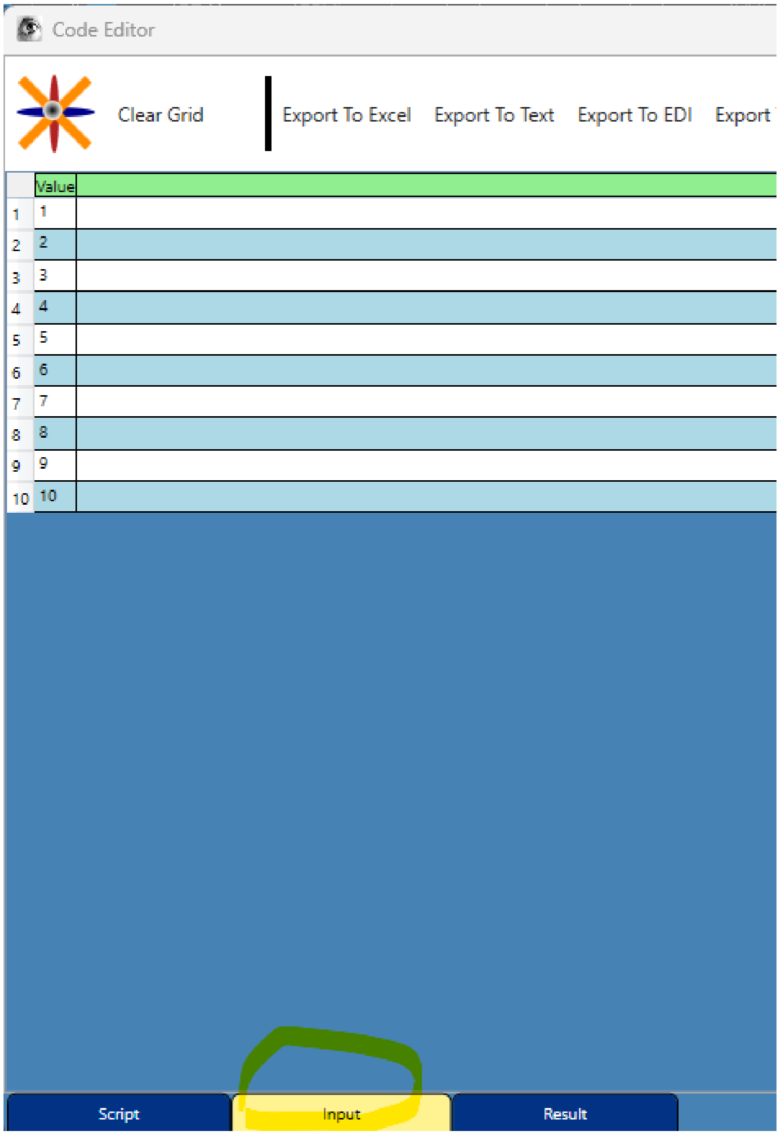
```
using System;
using System.Xml;
using System.Data;
using System.Data.SqlClient;
using System.Windows.Forms;
using System.Drawing;

using System.IO;

//Return value from Eval Code will be assigned to column
//DataTable T1: Whatever Data in Bottom Grid Of Query 1
//string s1   : Credential> User Name
//string s2   : Credential> Password
//string s3   : Credential> User Name
//string s4   : Credential> Password

namespace NSUpdateTable
{
    public class CSUpdateTable
    {
        //Return value from Eval Code will be assigned to column
        public static void Main()
        {

        }
        public static DataTable UpdateTable(DataTable T1,string s1,string s2,string s3,string s4,ref string s5)
         {
             MessageBox.Show("Hello World");
             return (T1);
         }
    }
}
```

You will see 5 messages popping



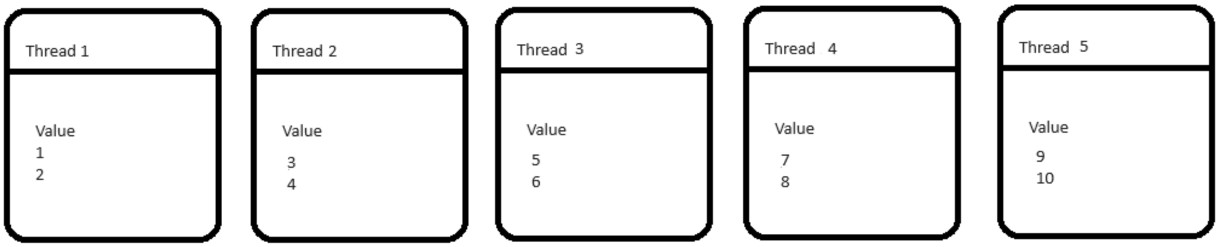
```
using System;
using System.Xml;
using System.Data;
using System.Data.SqlClient;
using System.Windows.Forms;
using System.Drawing;

using System.IO;

//Return value from Eval Code will be assigned to column
//DataTable T1: Whatever Data in Bottom Grid Of Query 1
//string s1   : Credential> User Name
//string s2   : Credential> Password
//string s3   : Credential> User Name
//string s4   : Credential> Password

namespace NSUpdateTable
{
    public class CSUpdateTable
    {
        //Return value from Eval Code will be assigned to column
        public static void Main()
        {

        }
        public static DataTable UpdateTable(DataTable T1,string s1,stri          s3,s
         {
             MessageBox.Show("Hello World");
             return (T1);
         }
    }
}
```

Navigate to Input Tab and load a data (In below scenario we are loading 10 rows)

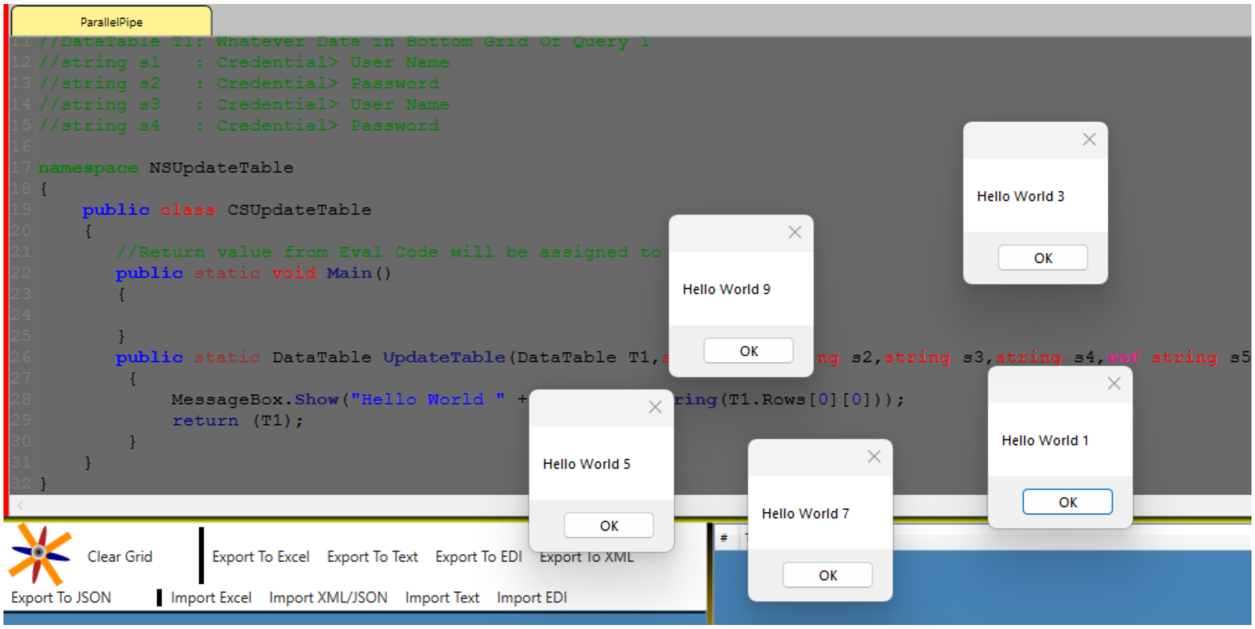


	Value
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

Navigate back to Script Tab. Now, if we again run it with 5 threads, the input data will be split in to 5 and distributed equally to each thread



Now displaying the first number along with Hello world. (You will see 1,3,5, 7, 9)



```
//DataTable T1: Whatever Data in Bottom Grid Of Query 1
//string s1    : Credential> User Name
//string s2    : Credential> Password
//string s3    : Credential> User Name
//string s4    : Credential> Password

namespace NSUpdateTable
{
    public class CSUpdateTable
    {
        //Return value from Eval Code will be assigned to
        public static void Main()
        {

        }
        public static DataTable UpdateTable(DataTable T1,s          ng s2,string s3,string s4,ref string s5
         {
             MessageBox.Show("Hello World " +           ring(T1.Rows[0][0]));
             return (T1);
         }
    }
}
```

Hello World 3

Hello World 9

Hello World 5

Hello World 7

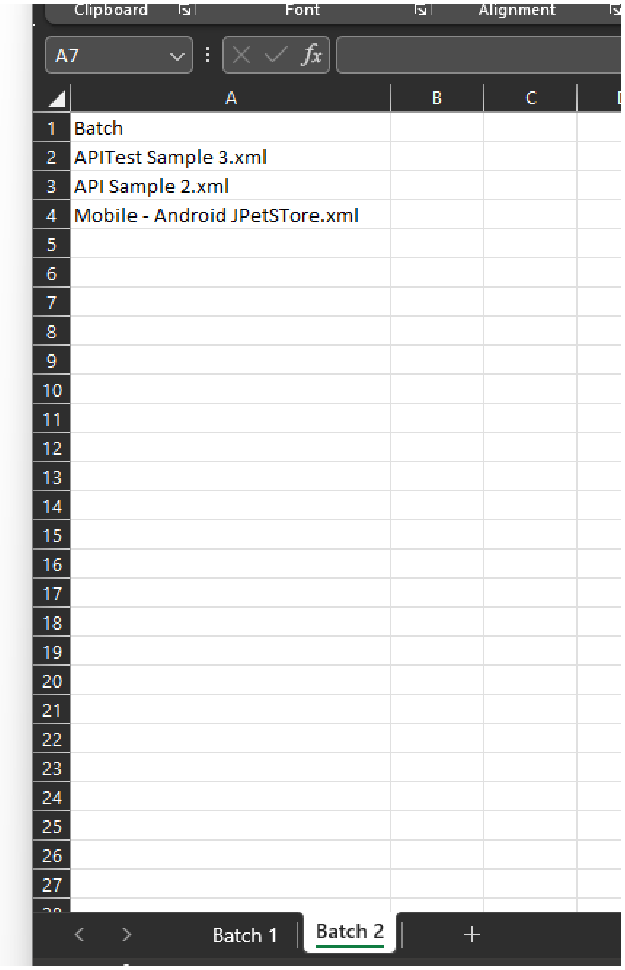
Hello World 1

Running Flow Codes In Parallel

NestedFlow Automation tool also allows us to run visual flow codes in parallel.

Note: Do not run visually intensive test cases in parallel. It might create focus issues some times. This option runs excellently for non-UI test cases (mainly batch jobs and API tests)

In An Excel, enter the batch of test cases need to run in parallel

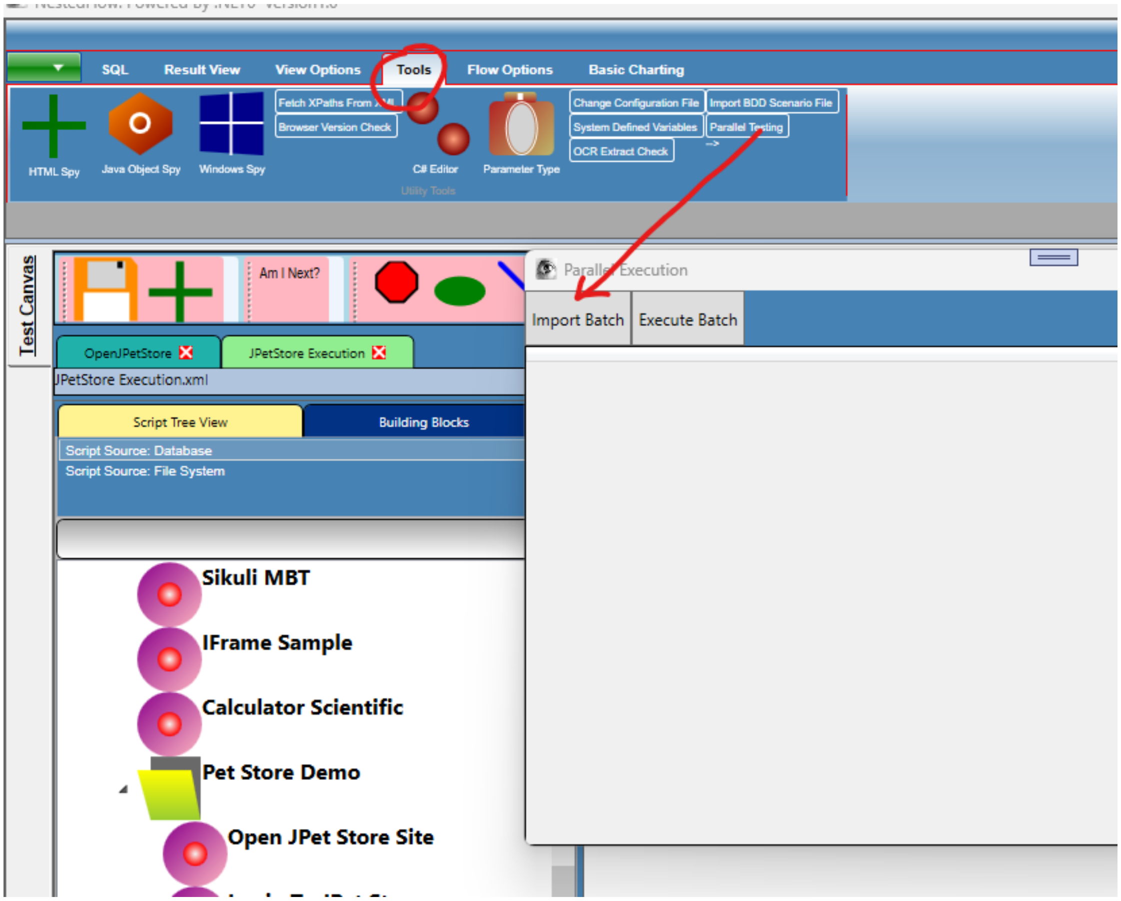


	A	B	C
1	Batch		
2	APITest Sample 3.xml		
3	API Sample 2.xml		
4	Mobile - Android JPetSTore.xml		

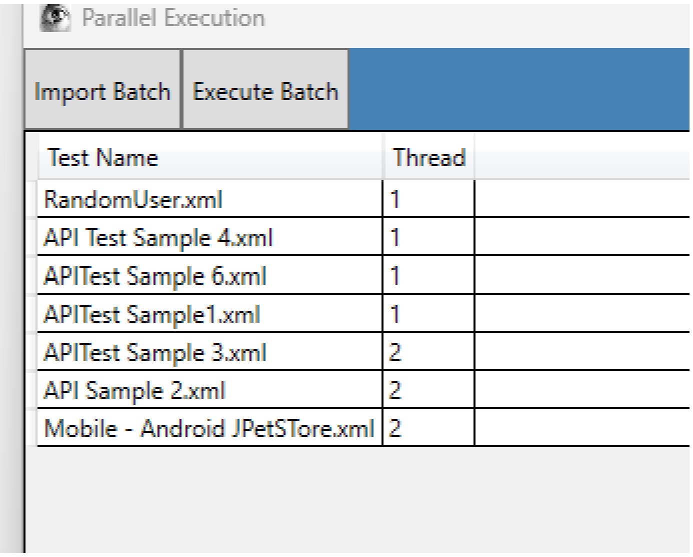
Each batch of test cases should be in a sheet starting with "Batch". The first row is considered as header name and all others are test case names. Each sheet will have only one column.

In this example I am using 2 sheets each containing test case names. Test cases in each batch will be run sequentially but all batches are run parallel. Maximum threads allowed are 10.

In Tools ribbon menu, click on **Parallel Testing** menu

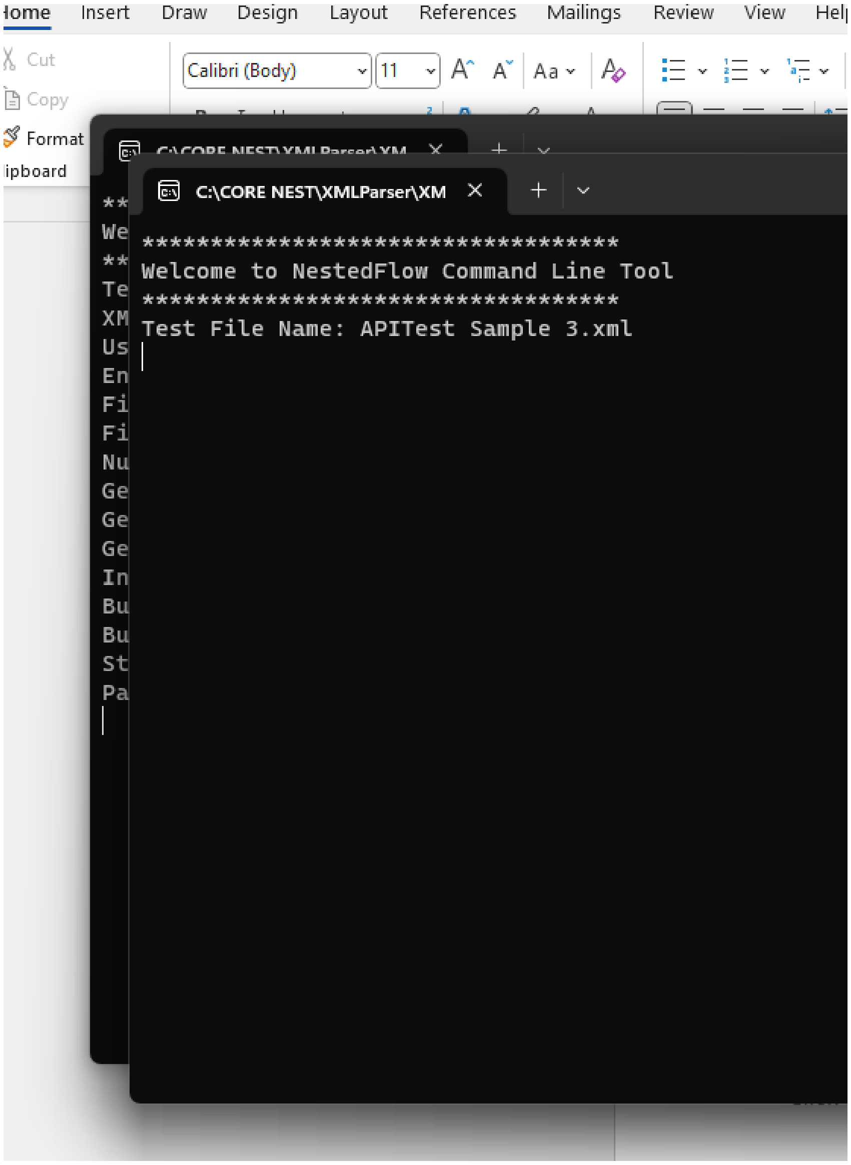


Click on **Import Batch** button and choose excel file. The batch details will be displayed



Test Name	Thread
RandomUser.xml	1
API Test Sample 4.xml	1
APITest Sample 6.xml	1
APITest Sample1.xml	1
APITest Sample 3.xml	2
API Sample 2.xml	2
Mobile - Android JPetSTore.xml	2

Click on Execute Batch to run Batches 1 and 2 parallelly.



```
**
We
**
Te
XM
Us
En
Fi
Fi
Nu
Ge
Ge
Ge
In
Bu
Bu
St
Pa
```

```
***********************************
Welcome to NestedFlow Command Line Tool
***********************************
Test File Name: APITest Sample 3.xml
```