

NestedFlow Automation – Extensibility Of The Automation Framework

Creation Date: 12/8/2023

Tool Version: 1.4 and above

Table Of Contents

Introduction	1
Type Of C# Codes	2
Code After Execution Completion Of Test.....	2
Code After Each Step.....	3
Code On Data Table.....	4
Single Table Code	4
Multi Table Code	9
Web Automation Code.....	10
Using Additional Parameters.....	13
PowerShell Scripting	15

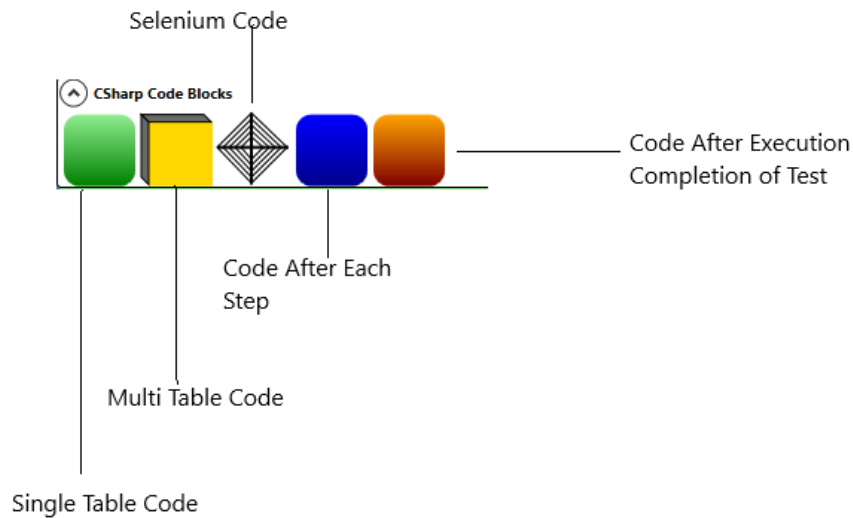
Introduction

This document explains how the NestedFlow Automation Framework can be extended beyond the default script less components using the C# code. Below items will be covered as part of this documentation

- Displaying log messages from code
- Utilization of powershell coding

Type Of C# Codes

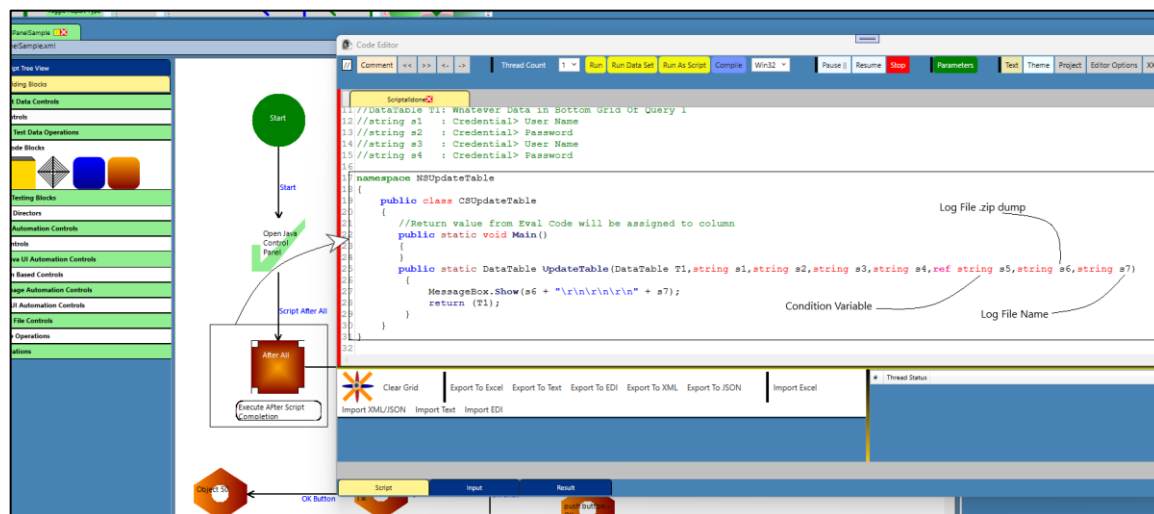
NestedFlow Automation tool allows dynamic C# coding to extend the tool usability. The main intention of the feature is to manipulate data tables. But we can use any library which is part of the tool to extend the features and capabilities of the framework. Below are the type of C# codes available in the tool



Code After Execution Completion Of Test




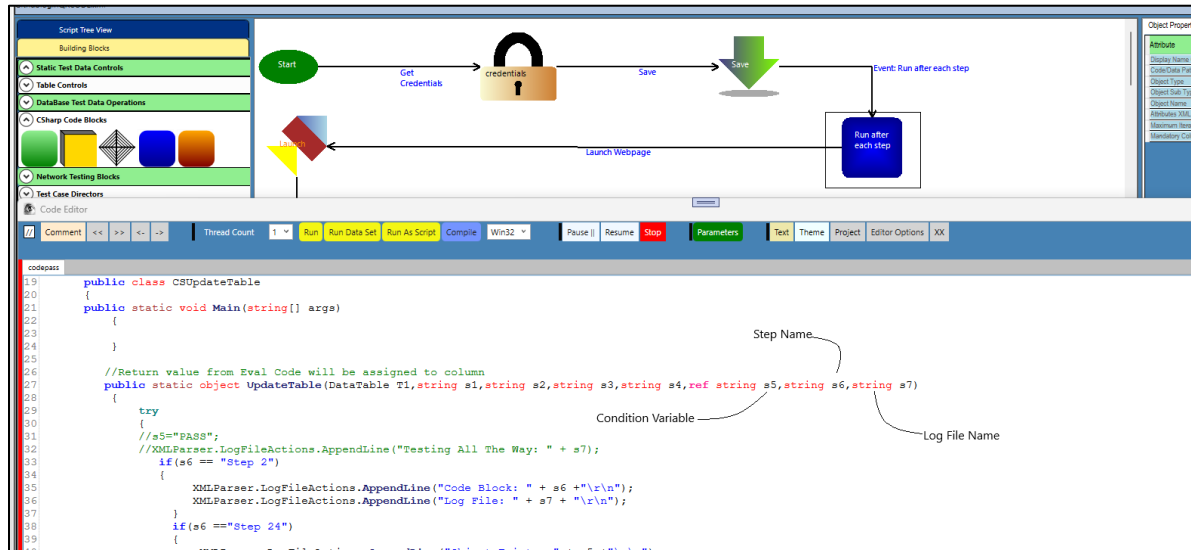
This code executes only once (That is after the execution of all steps in a test). This code type can be used to perform one time activity after the script like Test management tool upload or file upload etc.



It takes 7 parameters. Last 2 are pertaining the log file.

Code After Each Step

This code  executes after each step after its definition.

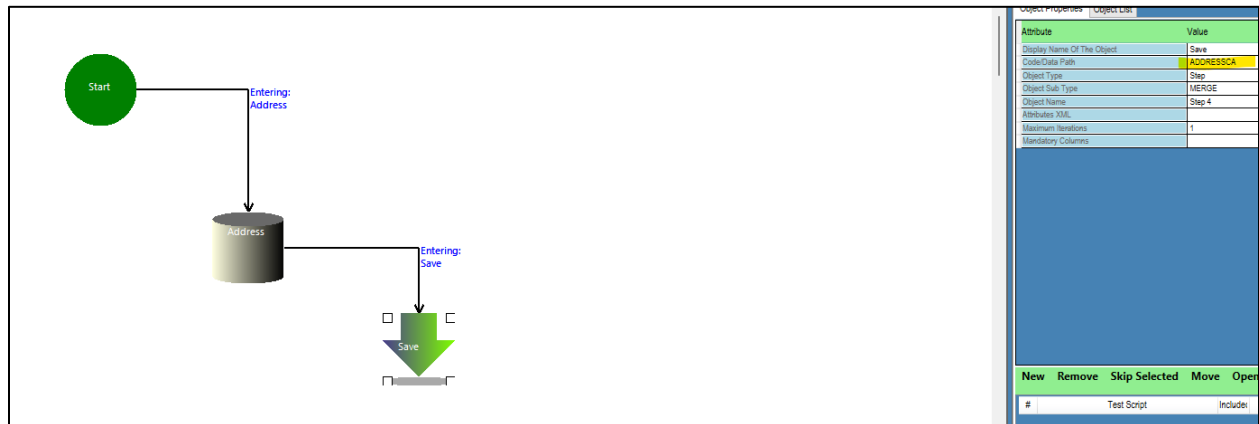


The screenshot displays a test automation tool interface. The top section shows a test flow diagram with the following steps: Start (green circle) → Get Credentials (lock icon) → Save (green arrow) → Event: Run after each step (blue box) → Launch Webpage (yellow diamond). The bottom section is a code editor showing the implementation of the 'Run after each step' event. The code is in C# and defines a class `CSUpdateTable` with a `Main` method. The `Main` method uses `XMLParser.LogFileActions.AppendLine` to log test results. Annotations point to variables in the code: 'Step Name' points to `s7`, 'Condition Variable' points to `s6`, and 'Log File Name' points to `s5`.

```
19 public class CSUpdateTable
20 {
21     public static void Main(string[] args)
22     {
23     }
24 }
25
26 //Return value from Eval Code will be assigned to column
27 public static object UpdateTable(DataTable T1, string s1, string s2, string s3, string s4, ref string s5, string s6, string s7)
28 {
29     try
30     {
31         //s5="PASS";
32         //XMLParser.LogFileActions.AppendLine("Testing All The Way: " + s7);
33         if(s6 == "Step 2")
34         {
35             XMLParser.LogFileActions.AppendLine("Code Block: " + s6 + "\r\n");
36             XMLParser.LogFileActions.AppendLine("Log File: " + s7 + "\r\n");
37         }
38         if(s6 == "Step 24")
39         {
40             XMLParser.LogFileActions.AppendLine("Code Block: " + s6 + "\r\n");
41         }
42     }
43     catch { }
44 }
```

Code On Data Table

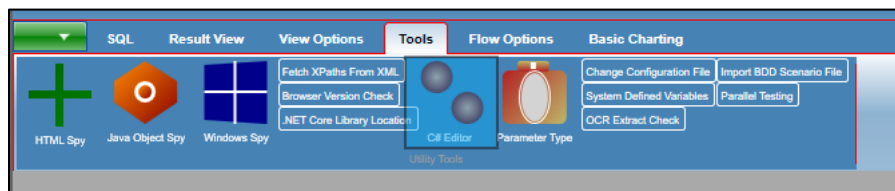
We can write custom C# code for modifying data. In this example we will add a 4th column and provide full address data by concatenating all other fields in it



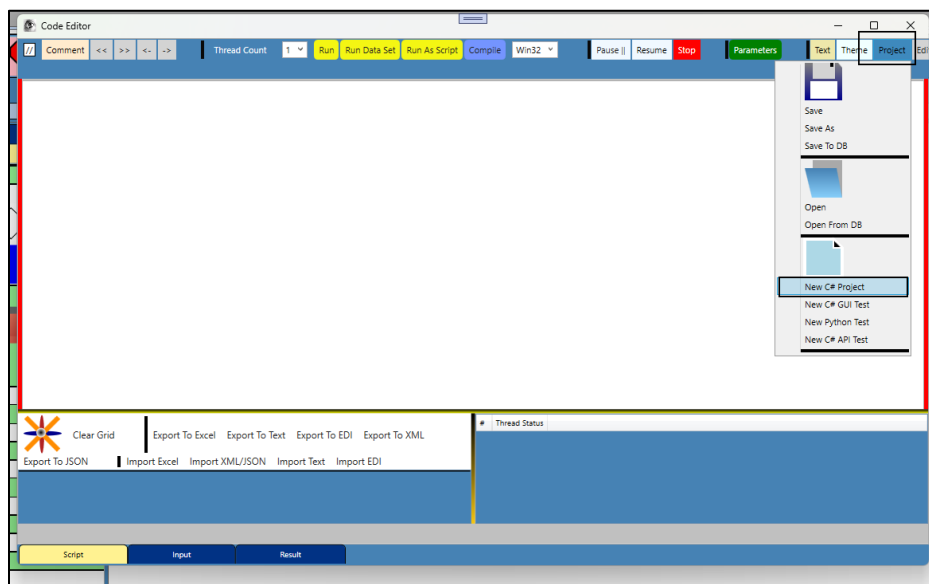
Let us save the Address data in a data table as shown above

Single Table Code

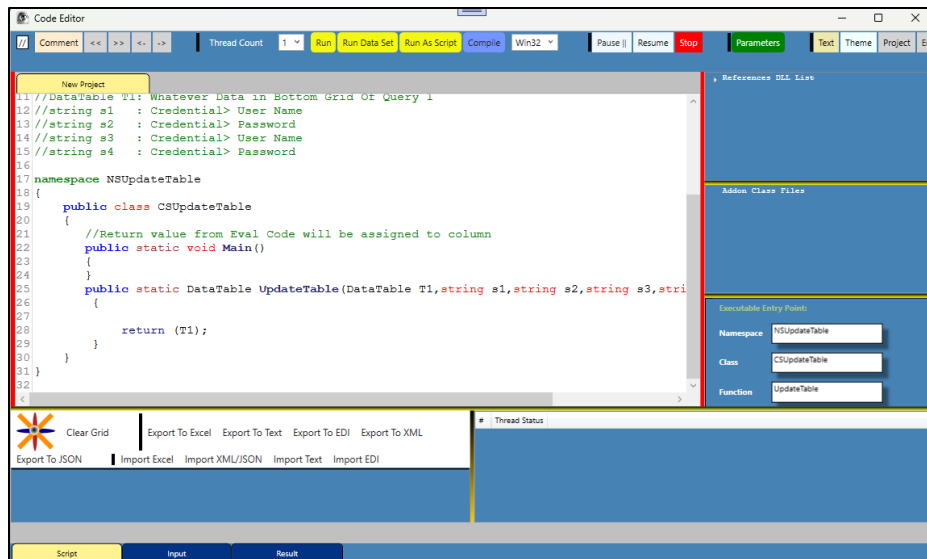
Navigate to **Tools** → **C# Editor**



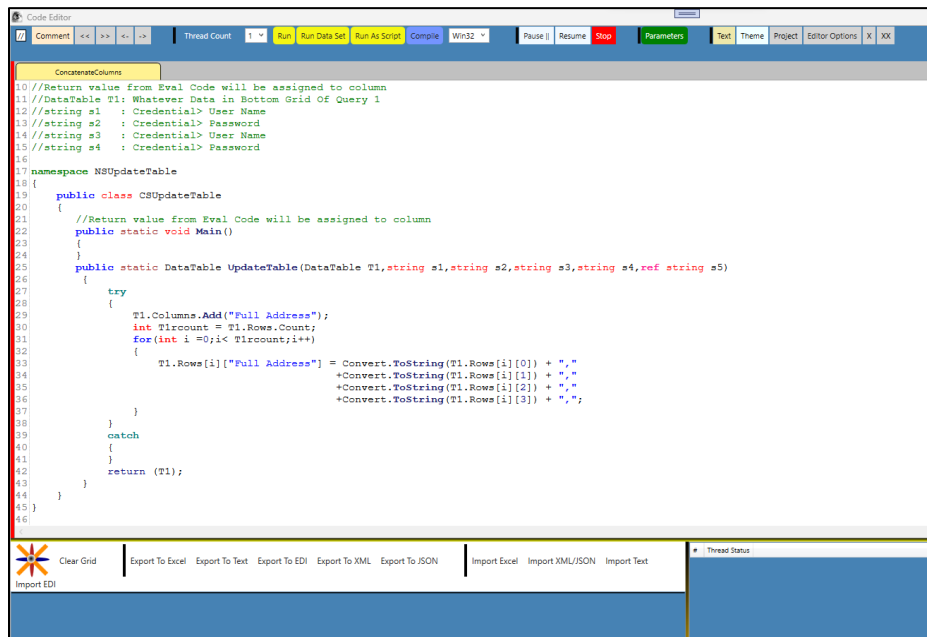
Navigate to **Project** → **New C# Project**



Code structure will be auto created



Either Save in file system or to database



Click on **Compile** button

If errors exist, resolve them and recompile

```

10 //Return value from Eval Code will be assigned to column
11 //DataTable T1: Whatever Data in Bottom Grid Of Query 1
12 //string s1 : Credential> User Name
13 //string s2 : Credential> Password
14 //string s3 : Credential> User Name
15 //string s4 : Credential> Password
16
17 namespace NSUpdateTable
18 {
19     public class CSUpdateTable
20     {
21         //Return value from Eval Code will be assigned to column
22         public static void Main()
23         {
24             public static DataTable UpdateTable(DataTable T1,string s1,string s2,string s3,string s4,ref string s5)
25             {
26                 try
27                 {
28                     T1.Columns.Add("Full Address");
29                     int Tircount = T1.Rows.Count;
30                     for(int i = 0;i< Tircount;i++)
31                     {
32                         T1.Rows[i]["Full Address"] = Convert.ToString(T1.Rows[i][0]) + ","
33                         +Convert.ToString(T1.Rows[i][1]) + ","
34                         +Convert.ToString(T1.Rows[i][2]) + ","
35                         +Convert.ToString(T1.Rows[i][3]) + ","
36                     }
37                 }
38                 catch
39                 {
40                 }
41                 return (T1);
42             }
43         }
44     }
45 }
46

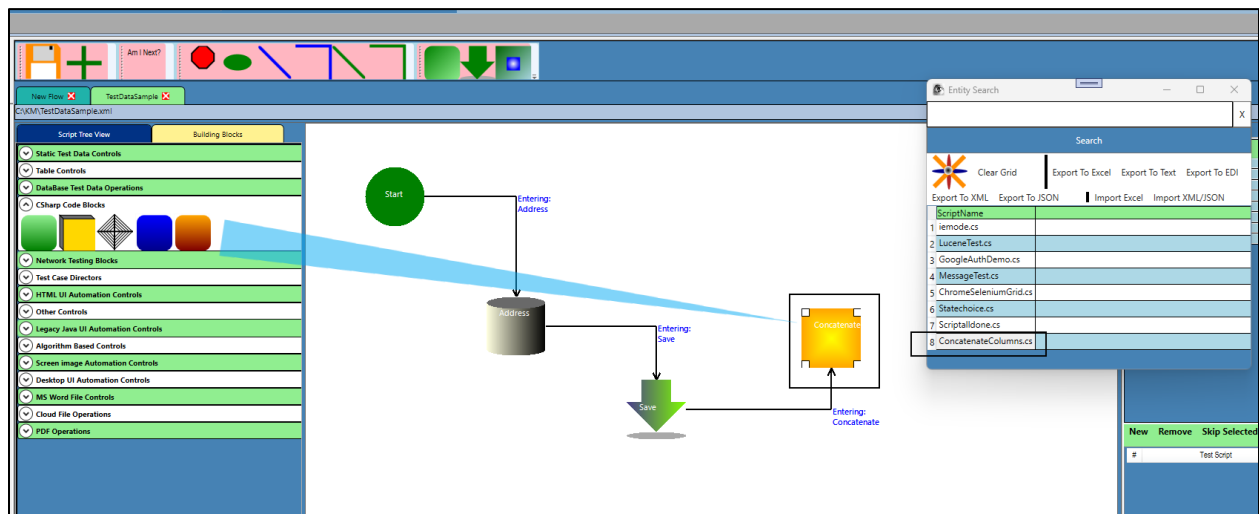
```

Clear Grid | Export To Excel | Export To Text | Export To EDI | Export To XML | Export To JSON | Import Excel | Import XML/JSON | Import Text

#	Filename	Column	Error	Error Description
1	0	1	(35.85)-(35.85)	(35.85)-(35.85) CS1002 : expected

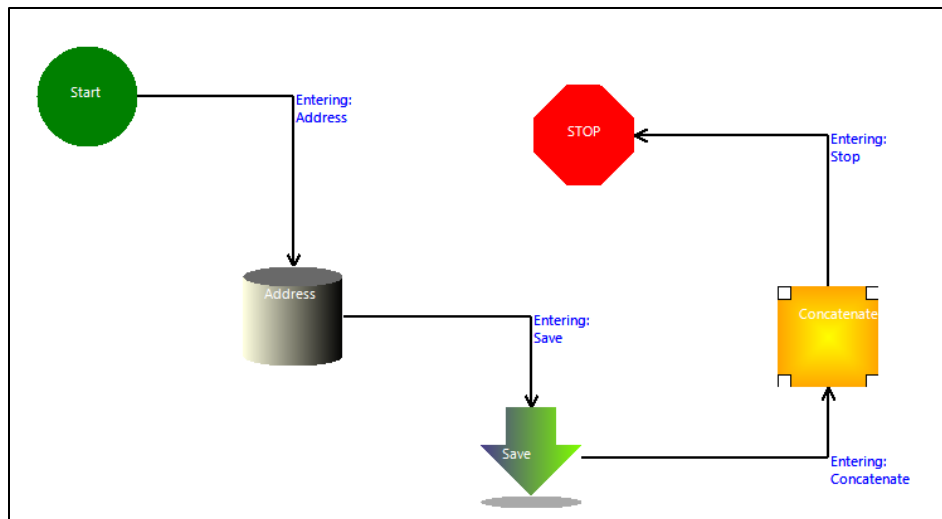
Close the form.

Add a Code block to the flow



Right Click on the block and add the script either from file or from database (in this case used database)

Complete the flow by adding Stop block:



The result: (Result Grid can be enabled by ribbon menu **View Options** → **Execution Results**)

▼

SQL

Result View

View Options

Tools


☐ SQL Tab
 ☐ Charts

☒ Flow Tab

☒ Execution Results

Table

Test Canvas




Clear Grid

Export To Excel

Export

	Address Line 1	City	State	Zip	Full Address
1	1671 E st	Bakersfield	CA	93301	1671 E st,Bakersfield
2	1673 E st	Bakersfield	CA	93301	1673 E st,Bakersfield
3	1675 E st	Bakersfield	CA	93301	1675 E st,Bakersfield
4	1677 E st	Bakersfield	CA	93301	1677 E st,Bakersfield
5	1679 E st	Bakersfield	CA	93301	1679 E st,Bakersfield

Test Canvas



Clear Grid

Export To Excel

Export To Text

Export To EDI

Export To XML

Result Grid

	Address Line 1	City	State	Zip	Full Address	
1	1671 E st	Bakersfield	CA	93301	1671 E st,Bakersfield,CA,93301,	
2	1673 E st	Bakersfield	CA	93301	1673 E st,Bakersfield,CA,93301,	
3	1675 E st	Bakersfield	CA	93301	1675 E st,Bakersfield,CA,93301,	
4	1677 E st	Bakersfield	CA	93301	1677 E st,Bakersfield,CA,93301,	
5	1679 E st	Bakersfield	CA	93301	1679 E st,Bakersfield,CA,93301,	
6	1681 E st	Bakersfield	CA	93301	1681 E st,Bakersfield,CA,93301,	
7	1683 E st	Bakersfield	CA	93301	1683 E st,Bakersfield,CA,93301,	
8	1685 E st	Bakersfield	CA	93301	1685 E st,Bakersfield,CA,93301,	
9	1687 E st	Bakersfield	CA	93301	1687 E st,Bakersfield,CA,93301,	
10	1689 E st	Bakersfield	CA	93301	1689 E st,Bakersfield,CA,93301,	
11	1691 E st	Bakersfield	CA	93301	1691 E st,Bakersfield,CA,93301,	
12	1693 E st	Bakersfield	CA	93301	1693 E st,Bakersfield,CA,93301,	
13	1695 E st	Bakersfield	CA	93301	1695 E st,Bakersfield,CA,93301,	
14	1697 E st	Bakersfield	CA	93301	1697 E st,Bakersfield,CA,93301,	
15	1699 E st	Bakersfield	CA	93301	1699 E st,Bakersfield,CA,93301,	
16	600 E Sussex Way	Fresno	CA	93704	600 E Sussex Way,Fresno,CA,93704,	
17	602 E Sussex Way	Fresno	CA	93704	602 E Sussex Way,Fresno,CA,93704,	
18	604 E Sussex Way	Fresno	CA	93704	604 E Sussex Way,Fresno,CA,93704,	
19	606 E Sussex Way	Fresno	CA	93704	606 E Sussex Way,Fresno,CA,93704,	
20	608 E Sussex Way	Fresno	CA	93704	608 E Sussex Way,Fresno,CA,93704,	
21	610 E Sussex Way	Fresno	CA	93704	610 E Sussex Way,Fresno,CA,93704,	
22	612 E Sussex Way	Fresno	CA	93704	612 E Sussex Way,Fresno,CA,93704,	
23	614 E Sussex Way	Fresno	CA	93704	614 E Sussex Way,Fresno,CA,93704,	
24	616 E Sussex Way	Fresno	CA	93704	616 E Sussex Way,Fresno,CA,93704,	
25	618 E Sussex Way	Fresno	CA	93704	618 E Sussex Way,Fresno,CA,93704,	
26	620 E Sussex Way	Fresno	CA	93704	620 E Sussex Way,Fresno,CA,93704,	
27	622 E Sussex Way	Fresno	CA	93704	622 E Sussex Way,Fresno,CA,93704,	
28	624 E Sussex Way	Fresno	CA	93704	624 E Sussex Way,Fresno,CA,93704,	
29	626 E Sussex Way	Fresno	CA	93704	626 E Sussex Way,Fresno,CA,93704,	
30	628 E Sussex Way	Fresno	CA	93704	628 E Sussex Way,Fresno,CA,93704,	
31	630 E Sussex Way	Fresno	CA	93704	630 E Sussex Way,Fresno,CA,93704,	
32	632 E Sussex Way	Fresno	CA	93704	632 E Sussex Way,Fresno,CA,93704,	
33	634 E Sussex Way	Fresno	CA	93704	634 E Sussex Way,Fresno,CA,93704,	
34	636 E Sussex Way	Fresno	CA	93704	636 E Sussex Way,Fresno,CA,93704,	
35	638 E Sussex Way	Fresno	CA	93704	638 E Sussex Way,Fresno,CA,93704,	

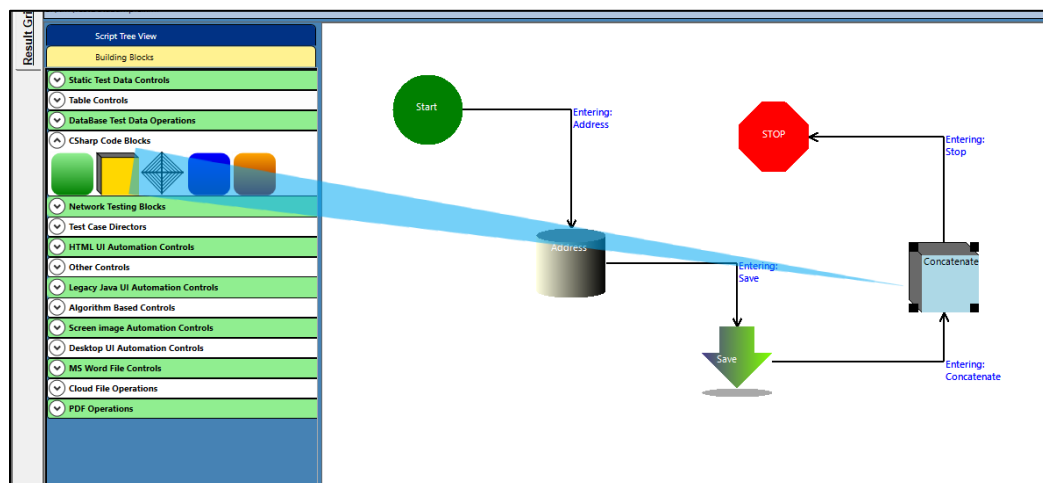
Multi Table Code

```
11//DataTable T1: Whatever Data in Bottom Grid Of Query 1
12//string s1 : Credential> User Name
13//string s2 : Credential> Password
14//string s3 : Credential> User Name
15//string s4 : Credential> Password
16
17namespace NSUpdateTable
18{
19    public class CSUpdateTable
20    {
21        //Return value from Eval Code will be assigned to column
22        public static void Main()
23        {
24        }
25        public static DataTable UpdateTable(DataSet TSET, string s1, string s2, string s3, string s4, ref string s5)
26        {
27            DataTable T1 = TSET.Tables["ADDRESSCA"];
28            try
29            {
30                T1.Columns.Add("Full Address");
31                int Tircount = T1.Rows.Count;
32                for(int i =0;i< Tircount;i++)
33                {
34                    T1.Rows[i]["Full Address"] = Convert.ToString(T1.Rows[i][0]) + ", "
35                                                +Convert.ToString(T1.Rows[i][1]) + ", "
36                                                +Convert.ToString(T1.Rows[i][2]) + ", "
37                                                +Convert.ToString(T1.Rows[i][3]) ;
38                }
39            }
40            catch
41            {
42            }
43            return (T1);
44        }
45    }
46}
47
```

First parameter is DataSet saving all tables created during flow

Derive the expected table using saved table name

And linking this table to DataSet code block



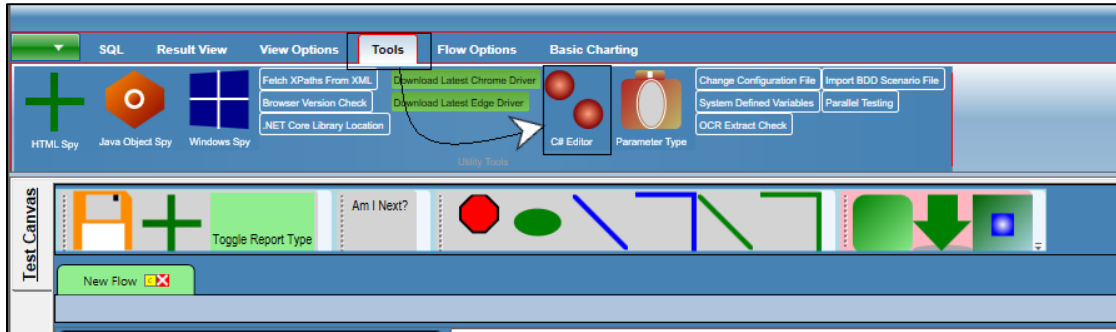
This kind of coding is useful when you need to access more than 1 data table at a time

Web Automation Code

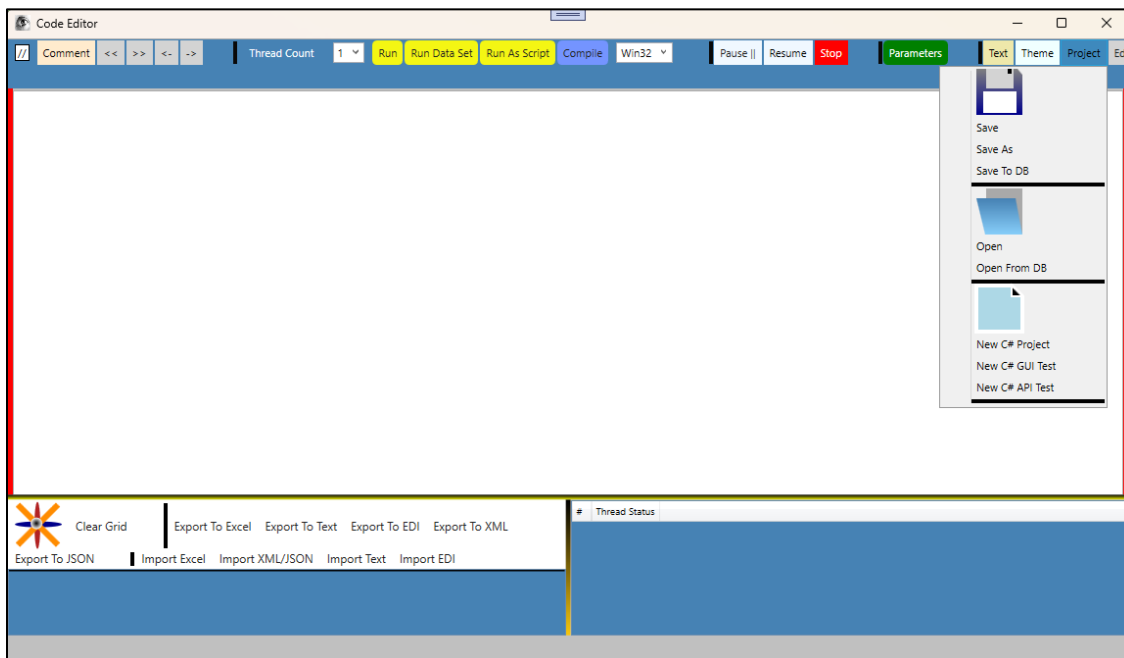


This code is for manipulating the webdriver object

Navigate to **Tools** → **C# Editor**



Navigate to **Project** → **New C# GUI Test**



Following code will be created:

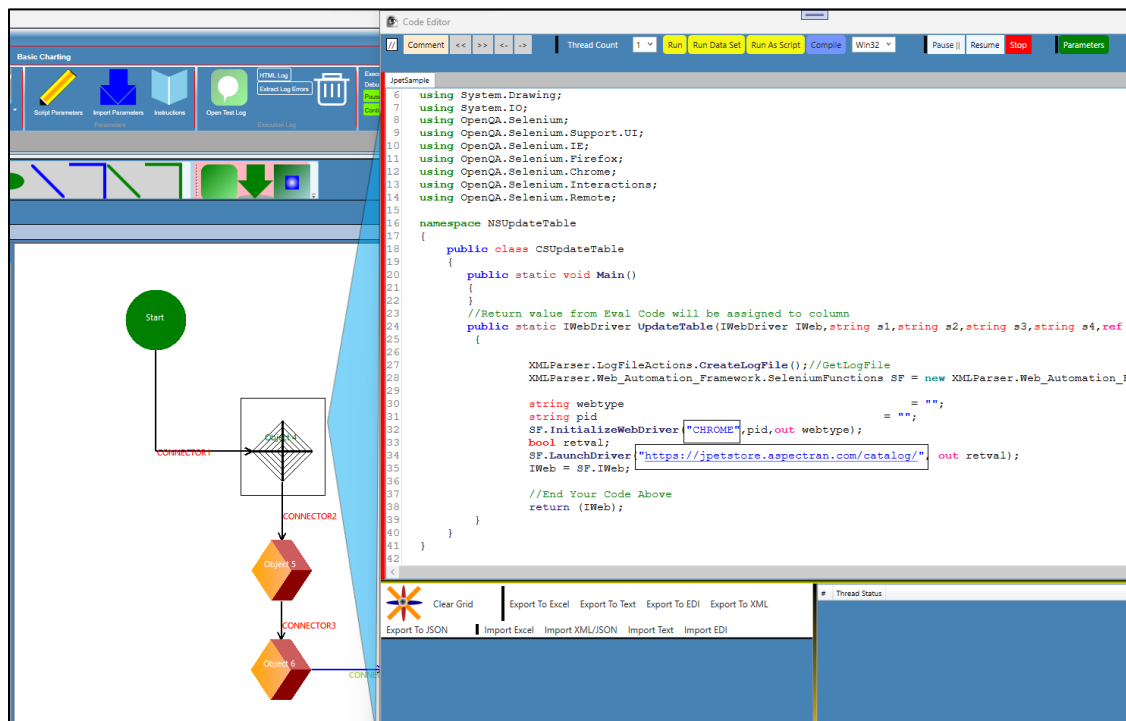
```

1 using System;
2 using System.Xml;
3 using System.Data;
4 using System.Data.SqlClient;
5 using System.Windows.Forms;
6 using System.Drawing;
7 using System.IO;
8 using OpenQA.Selenium;
9 using OpenQA.Selenium.Support.UI;
10 using OpenQA.Selenium.IE;
11 using OpenQA.Selenium.Firefox;
12 using OpenQA.Selenium.Chrome;
13 using OpenQA.Selenium.Interactions;
14 using OpenQA.Selenium.Remote;
15
16 namespace NSUpdateTable
17 {
18     public class CSUpdateTable
19     {
20         public static void Main()
21         {
22         }
23         //Return value from Eval Code will be assigned to column
24         public static IWebDriver UpdateTable(IWebDriver IWeb,string s1,string s2,string s3,string s4,ref string s5)
25         {
26
27             XMLParser.LogFileActions.CreateLogFile();//GetLogFile
28             XMLParser.Web_Automation_Framework.SeleniumFunctions SF = new XMLParser.Web_Automation_Framework.SeleniumFunctions();
29
30             string webtype = "";
31             string pid = "";
32             SF.InitializeWebDriver("CHROME",pid,out webtype);
33             bool retval;
34             SF.LaunchDriver("https://CURL/", out retval);
35             IWeb = SF.IWeb;
36
37             //End Your Code Above
38             return (IWeb);
39         }
40     }
41 }
42

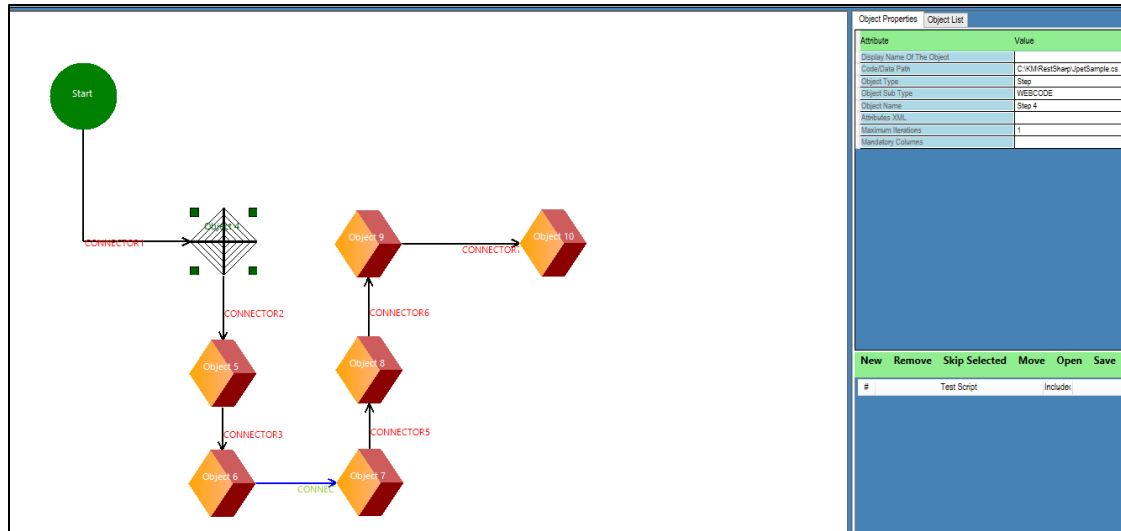
```

It takes, first parameter and return type as IWebDriver.

This code can be updated for browser and webpage and also can have custom code to perform actions which may be easier to achieve with code than blocks. In the current example we will just open jpet store on chrome and save the script.

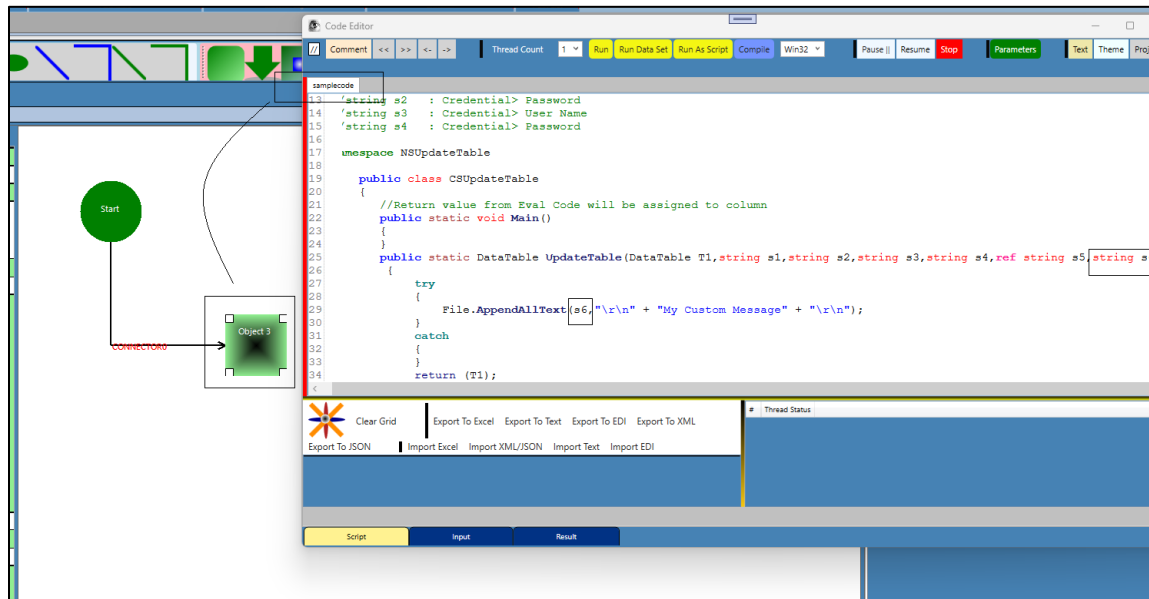


This code now can be assigned to a **WEBCODE** block used in place of usual Launch component and other components can be used just like earlier



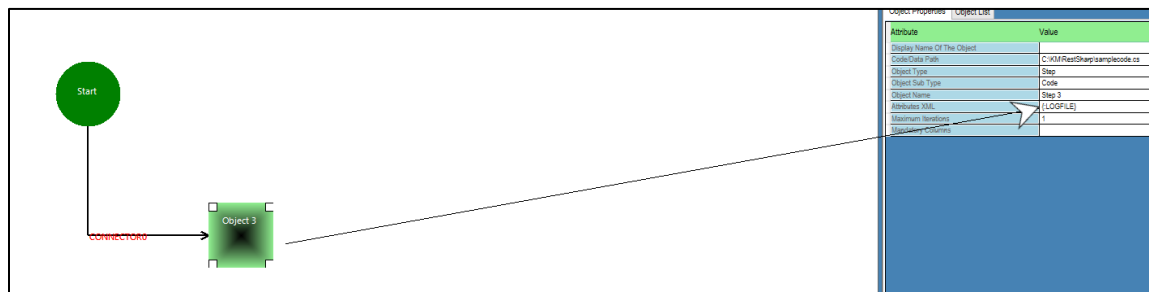
Using Additional Parameters

Single Table code, Multi Table code and Web code can be used with additional parameters. In the below example we will write a code that will have log messages inside the code.

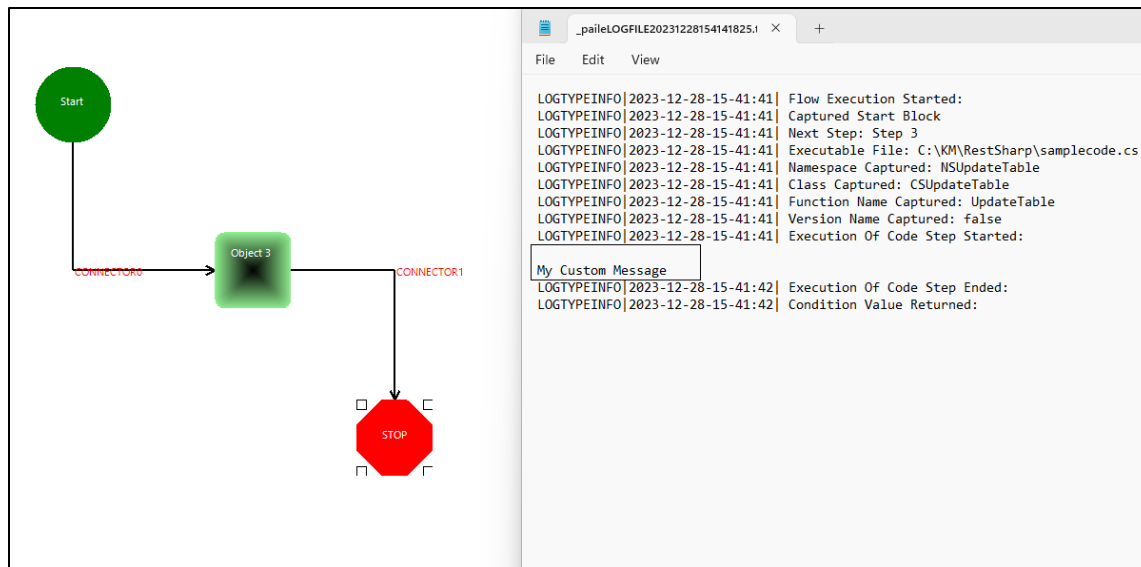


The extra parameter declared in the code needs to be added attributes in a comma separated way.

(In this case we are using only 1 parameter)

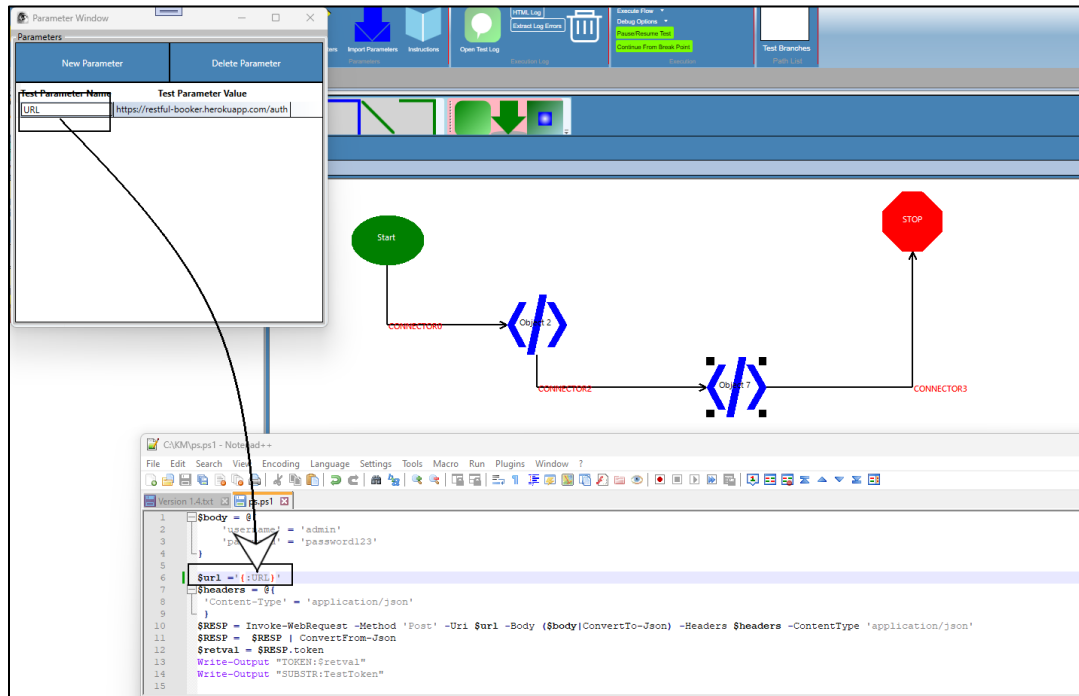


After the execution, Logfile will show:



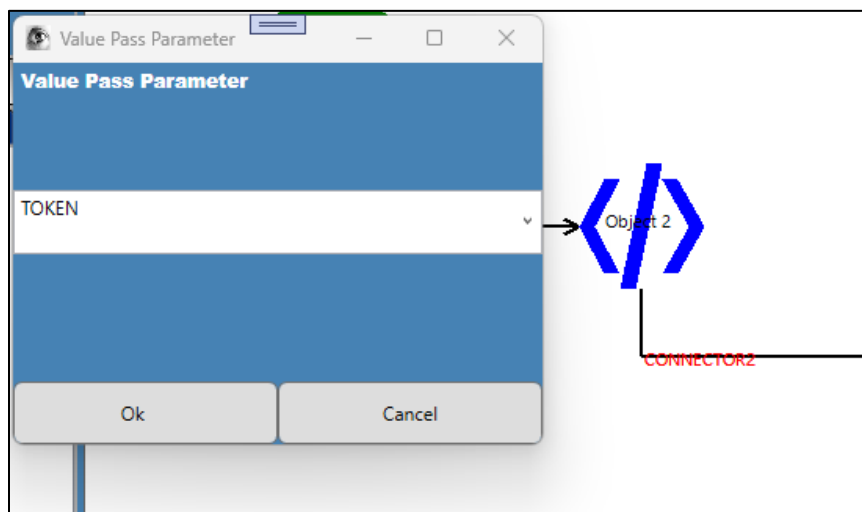
PowerShell Scripting

NestedFlow Automation tool allows scripting in PowerShell. Tool also allows capturing of the output from the PowerShell. In the below example we will execute a PowerShell script and get the responses back on to the flow.



The Test flow has a URL parameter which is used in the PowerShell script as shown above.

The first PowerShell component will execute the script and return TOKEN value (The pass parameter is **TOKEN** and the value Output by PowerShell which starts with **TOKEN:** (ParameterName:ParameterValue) will be assigned to this parameter



Log Shows:

```
LOGTYPEINFO|2023-12-29-14-07:10| Flow Execution Started: C:\KM\powershell.xml
LOGTYPEINFO|2023-12-29-14-07:10| Captured Start Block
LOGTYPEINFO|2023-12-29-14-07:10| Next Step: Step 2
LOGTYPEINFO|2023-12-29-14-07:10| Execute Powershell Script
LOGTYPEINFO|2023-12-29-14-07:10| $body = @{
    'username' = 'admin'
    'password' = 'password123'
}

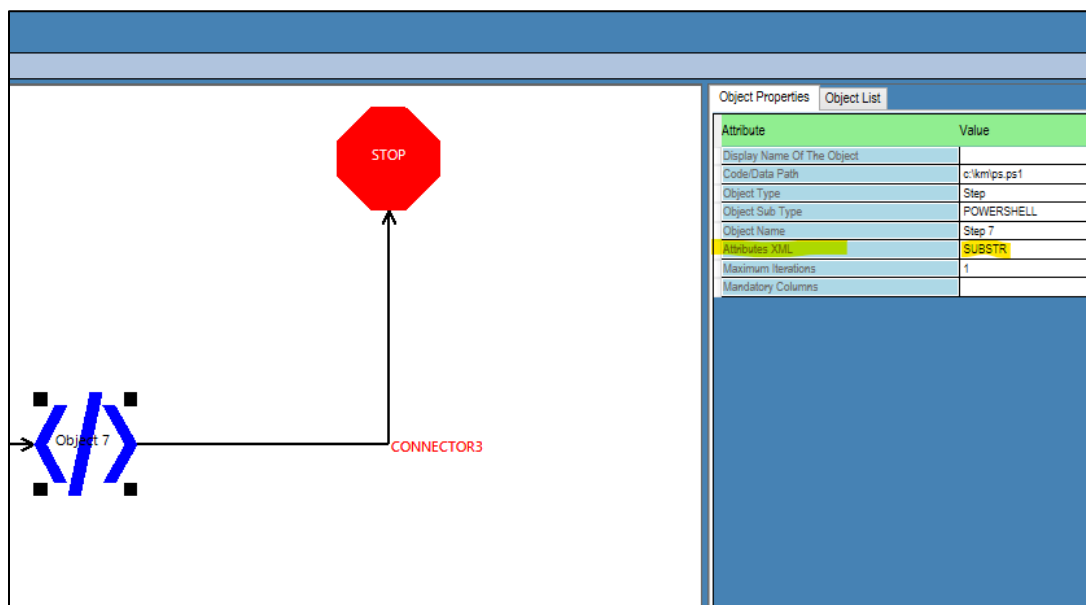
$url = 'https://restful-booker.herokuapp.com/auth'
$headers = @{
    'Content-Type' = 'application/json'
}
$RESP = Invoke-WebRequest -Method 'Post' -Uri $url -Body ($body|ConvertTo-Json) -Headers $headers -ContentType 'application/json'
$RESP = $RESP | ConvertFrom-Json
$retval = $RESP.token
Write-Output "TOKEN:$retval"
Write-Output "SUBSTR:TestToken"

LOGTYPEINFO|2023-12-29-14-07:10| TOKEN:f5e636f1e748edf
```

And the parameter returned is as shown below

Parameters	
<div>New ParameterDelete Parameter</div>	
Test Parameter Name	Test Parameter Value
URL	https://restful-booker.herokuapp.com/auth
TOKEN	f5e636f1e748edf

The 2nd PowerShell component has attribute **SUBSTR**







In this case, the script will not be executed, but Variable SUBSTR from previous execution will be returned.

```

    }
    $RESP = Invoke-WebRequest -Method 'Post' -Uri $url
    $RESP = $RESP | ConvertFrom-Json
    $retval = $RESP.token
    Write-Output "TOKEN:$retval"
    Write-Output "SUBSTR:TestToken"

```

-  2023-12-29-14-07:10 TOKEN:f5e636f1e748edf
-  2023-12-29-14-07:10 SUBSTR:TestToken
-  2023-12-29-14-07:10 Next Step: Step 7
-  2023-12-29-14-07:10 Execute Powershell Script

Parameters	
<div>New Parameter</div> <div>Delete Parameter</div>	
Test Parameter Name	Test Parameter Value
URL	https://restful-booker.herokuapp.com/auth
TOKEN	f5e636f1e748edf
SUBSTR	TestToken