NestedFlow Automation – Extensibility Of The Automation Framework

Creation Date: 12/8/2023

Tool Version: 1.4 and above

Table Of Contents

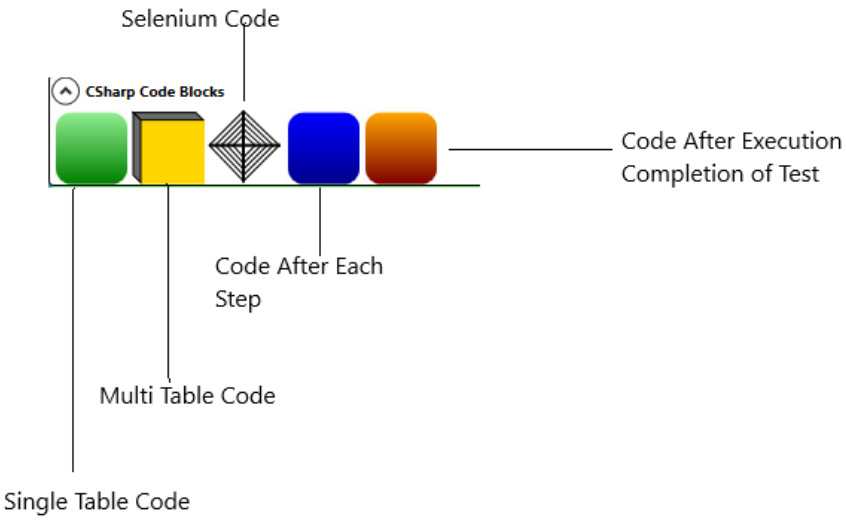
Introduction

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

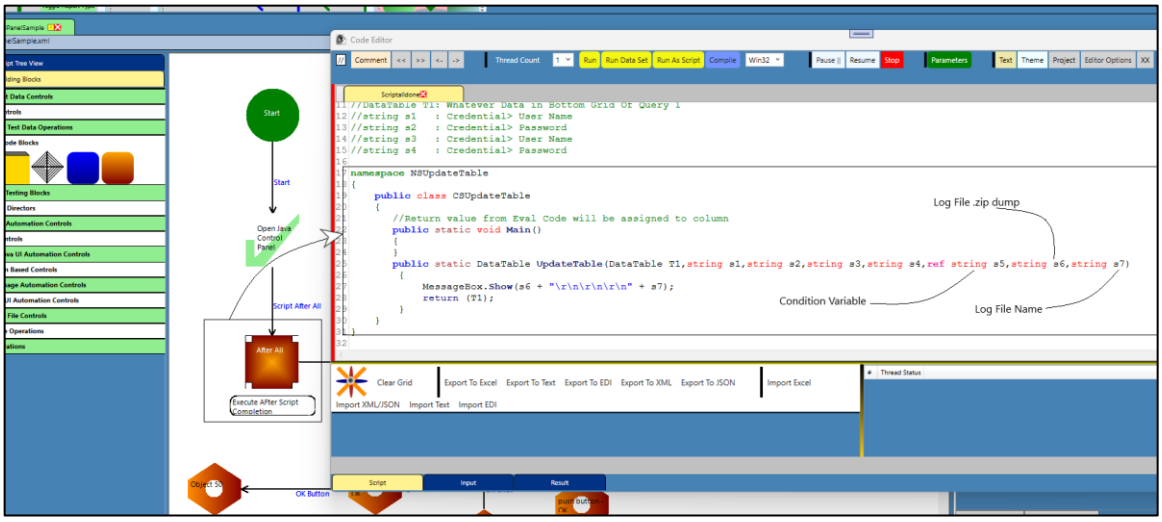
Type Of C# Codes

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

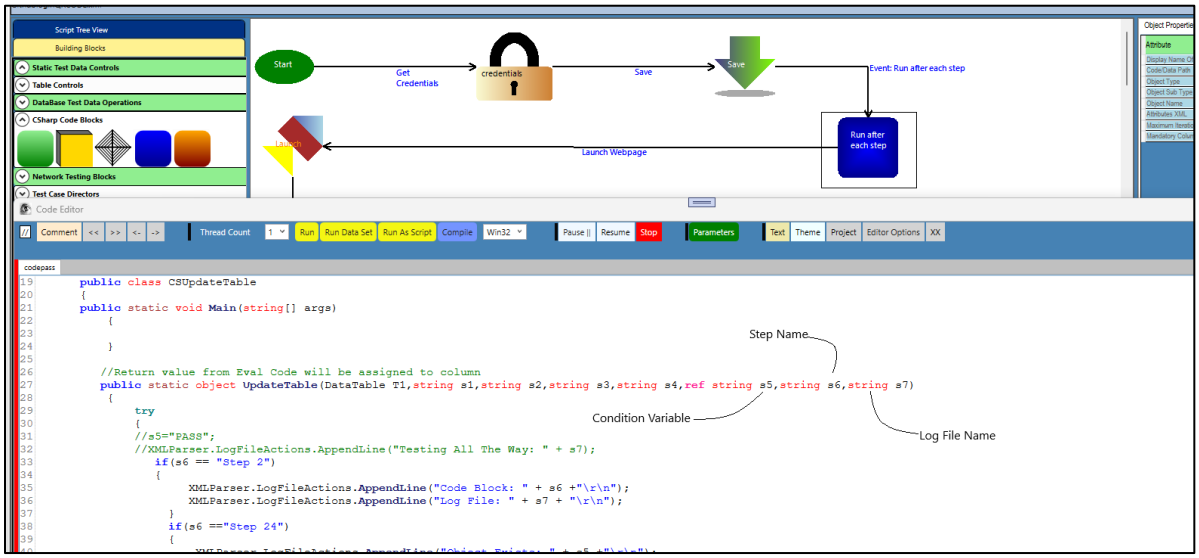
This code  executes only once (That is after the execution of all steps in a test). This code type can be used to perform one time activity after the script like Test management tool upload or file upload etc.



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

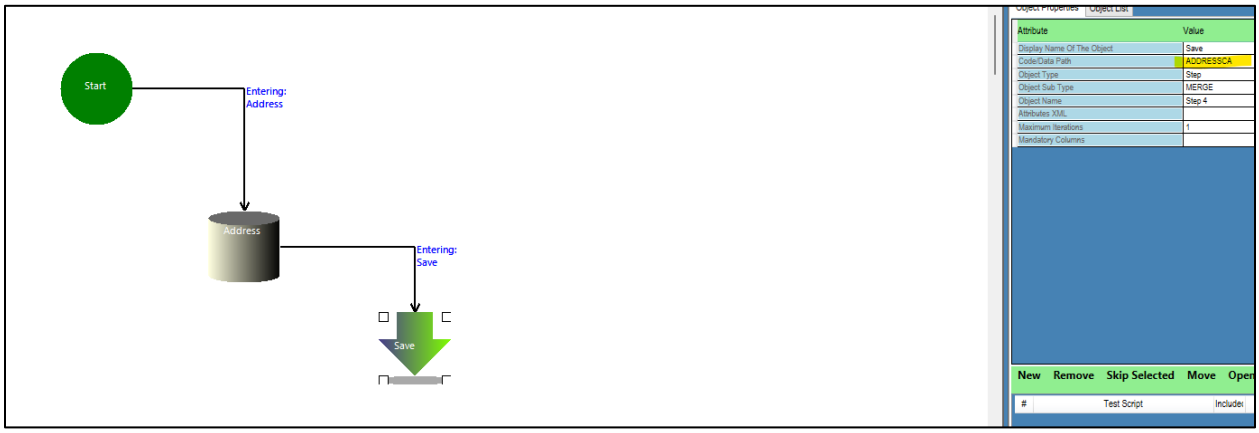
Code After Each Step

This code  executes after each step after its definition.



Code On Data Table

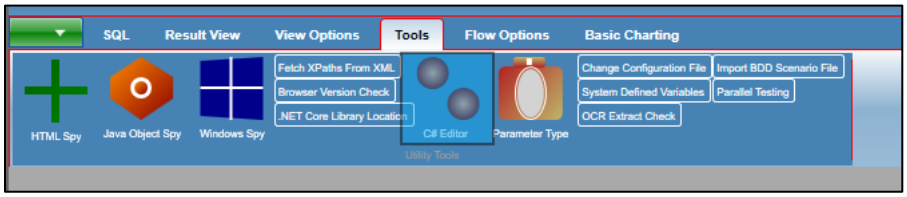
We can write custom C# code for modifying data. In this example we will add a 4th column and provide full address data by concatenating all other fields in it



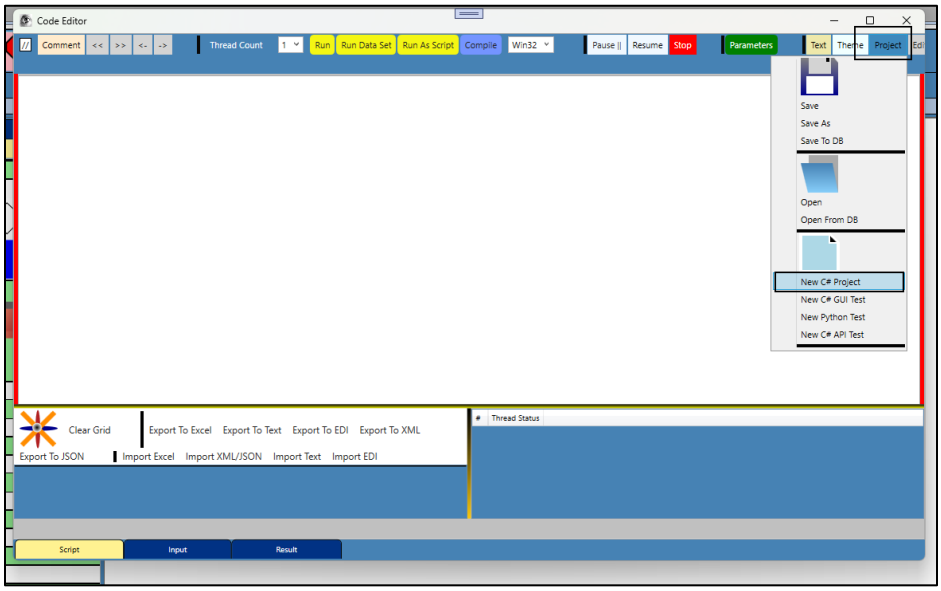
Let us save the Address data in a data table as shown above

Single Table Code

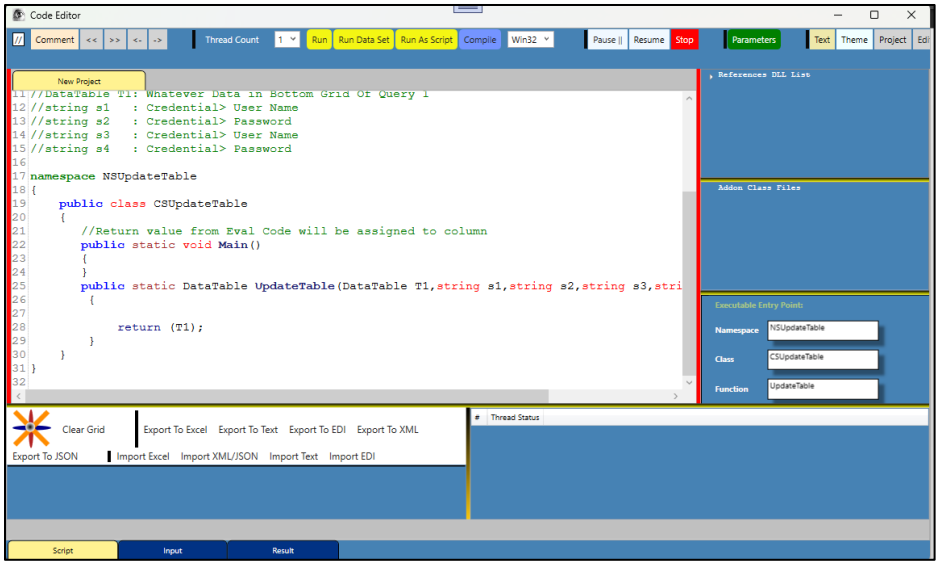
Navigate to **Tools → C# Editor**



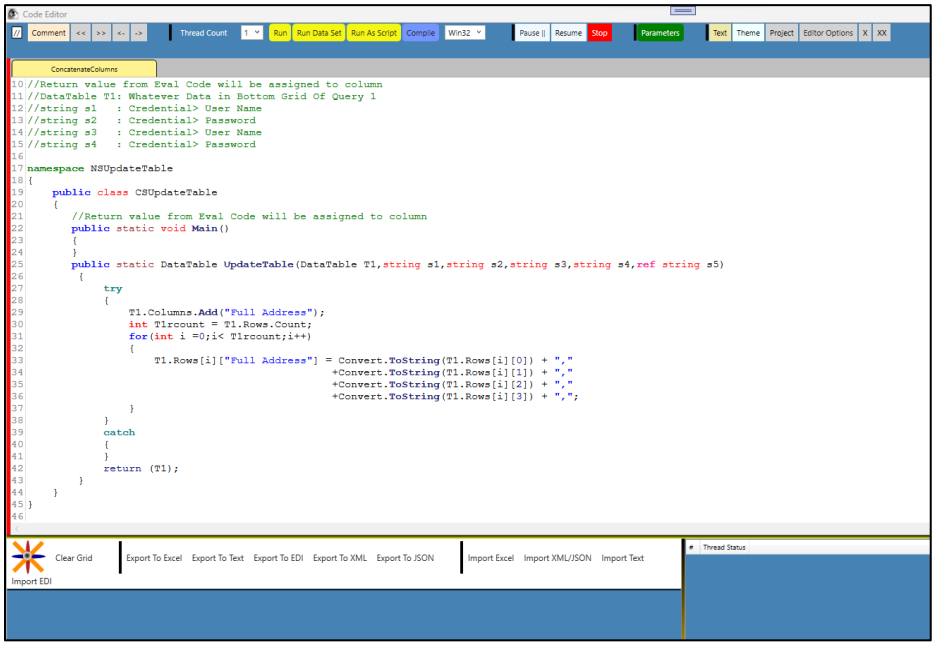
Navigate to **Project → New C# Project**



Code structure will be auto created

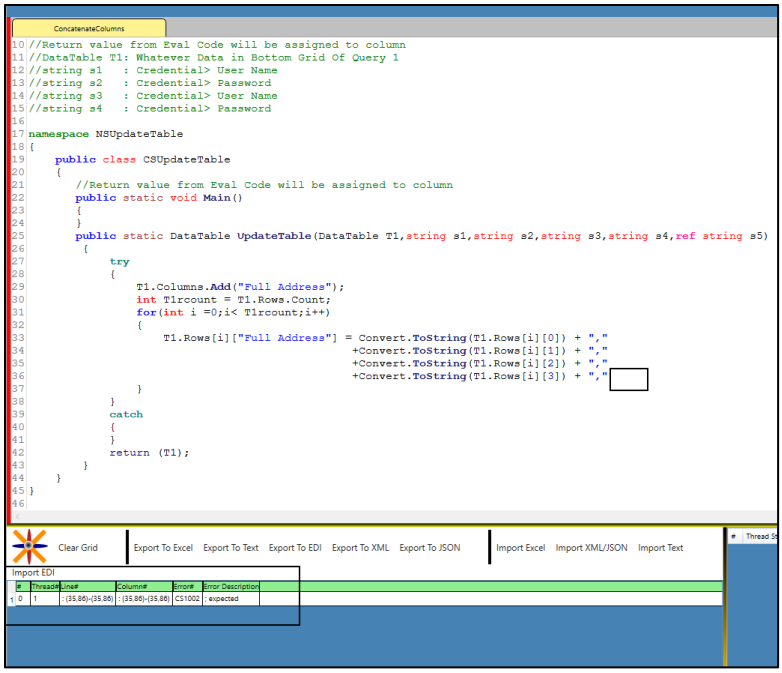


Either Save in file system or to database



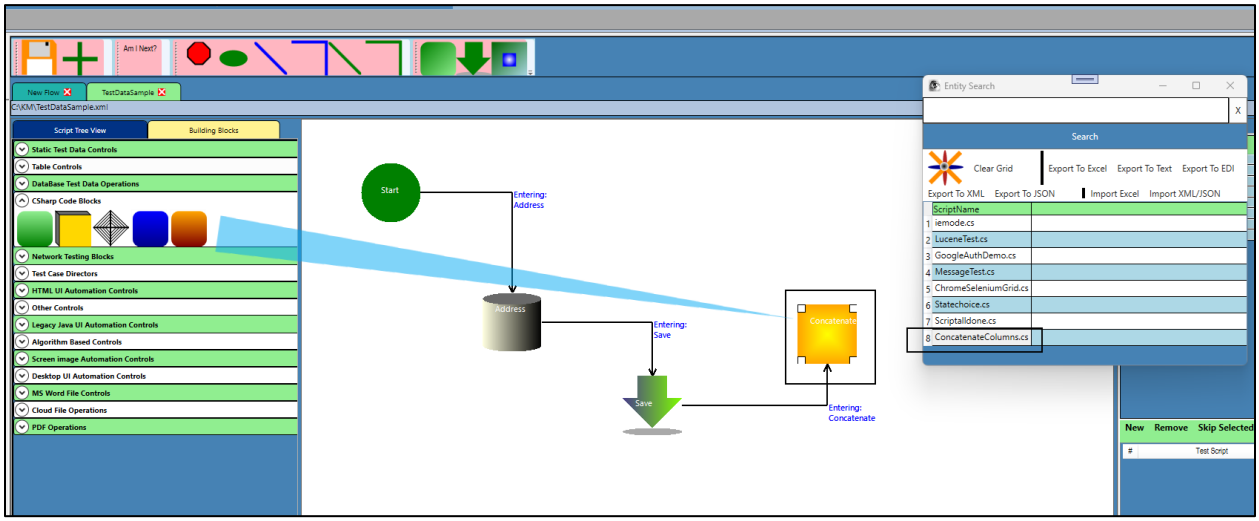
Click on **Compile** button

If errors exist, resolve them and recompile



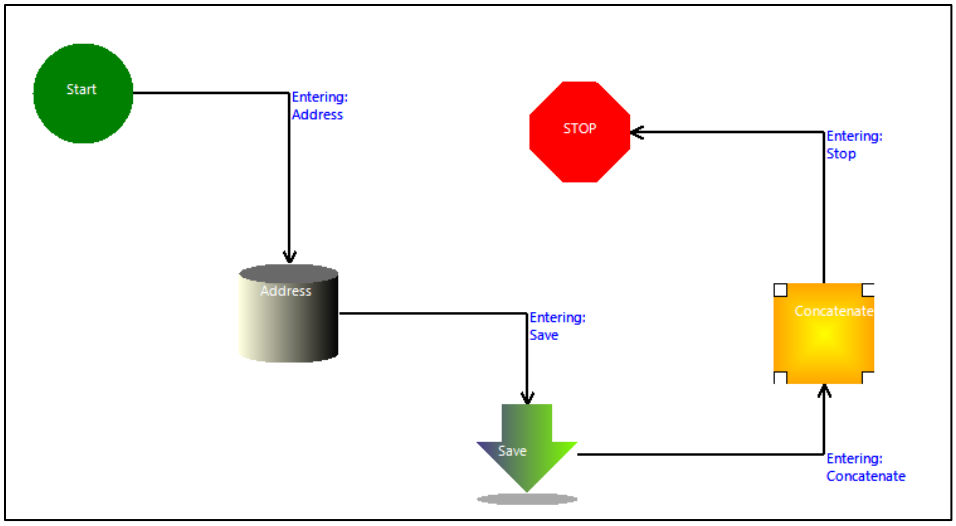
Close the form.

Add a Code block to the flow

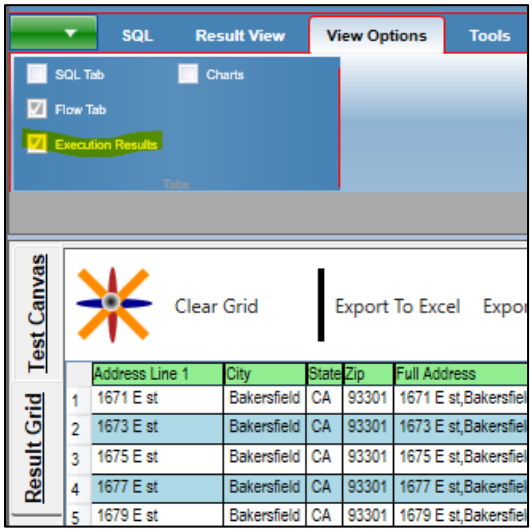


Right Click on the block and add the script either from file or from database (in this case used database)

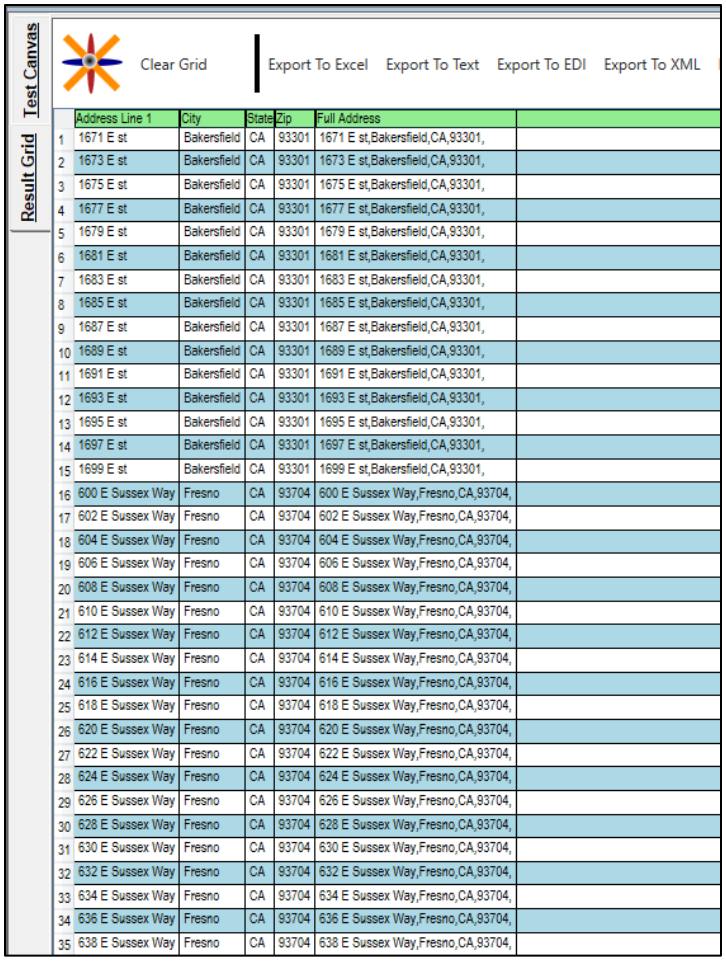
Complete the flow by adding Stop block:



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

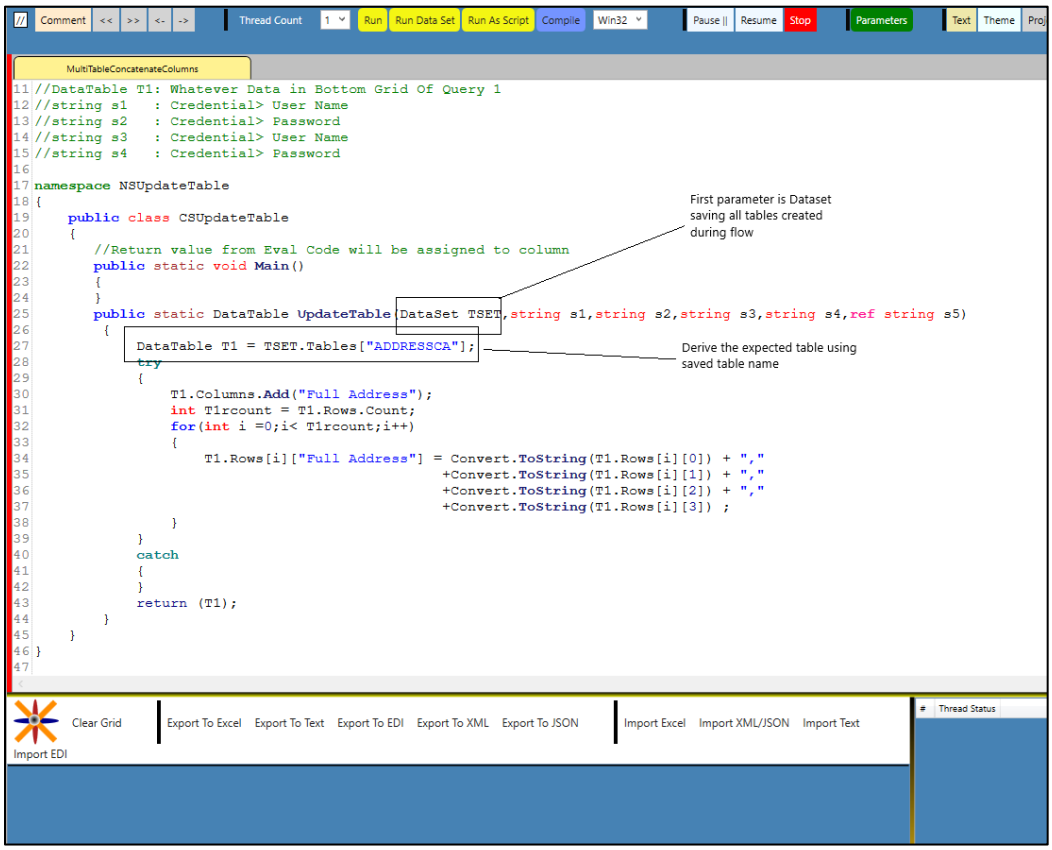


	Address Line 1	City	State	Zip	Full Address
1	1671 E st	Bakersfield	CA	93301	1671 E st,Bakersfiel
2	1673 E st	Bakersfield	CA	93301	1673 E st,Bakersfiel
3	1675 E st	Bakersfield	CA	93301	1675 E st,Bakersfiel
4	1677 E st	Bakersfield	CA	93301	1677 E st,Bakersfiel
5	1679 E st	Bakersfield	CA	93301	1679 E st,Bakersfiel

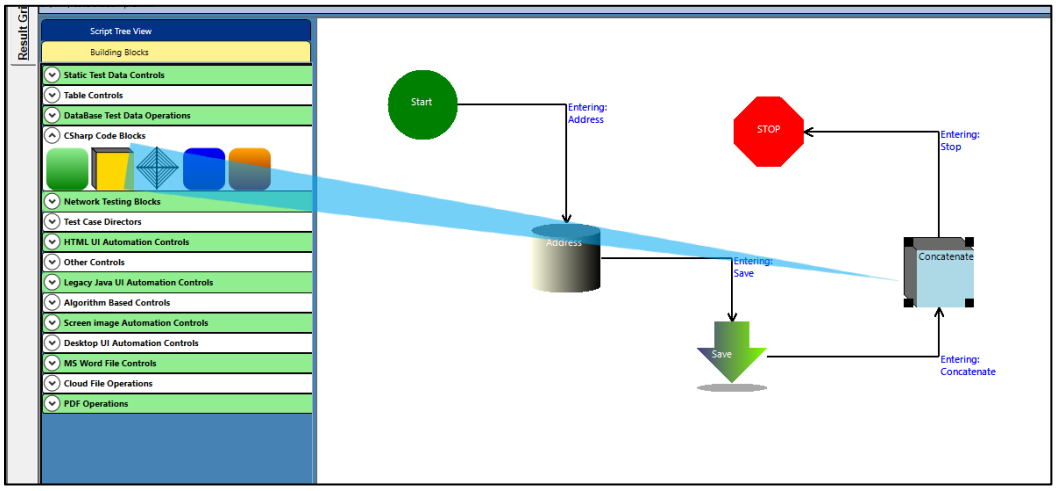


	Address Line 1	City	State	Zip	Full Address
1	1671 E st	Bakersfield	CA	93301	1671 E st,Bakersfield,CA,93301,
2	1673 E st	Bakersfield	CA	93301	1673 E st,Bakersfield,CA,93301,
3	1675 E st	Bakersfield	CA	93301	1675 E st,Bakersfield,CA,93301,
4	1677 E st	Bakersfield	CA	93301	1677 E st,Bakersfield,CA,93301,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



And linking this table to DataSet code block

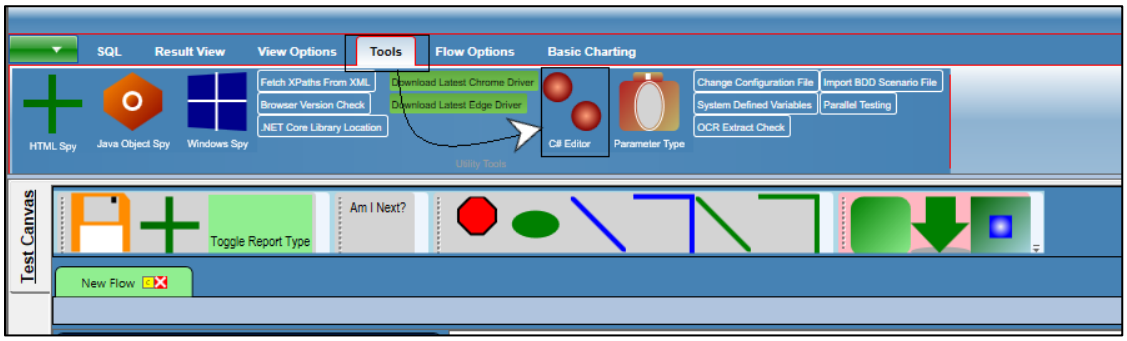


This kind of coding is useful when you need to access more than 1 data table at a time

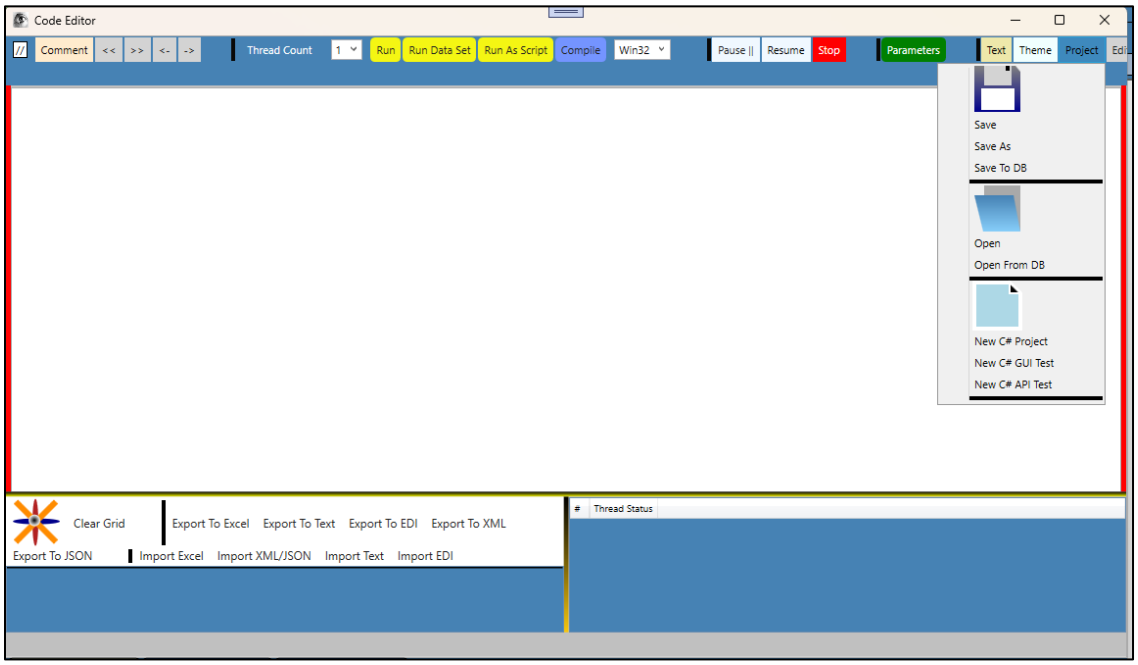
Web Automation Code

This code  is for manipulating the webdriver object

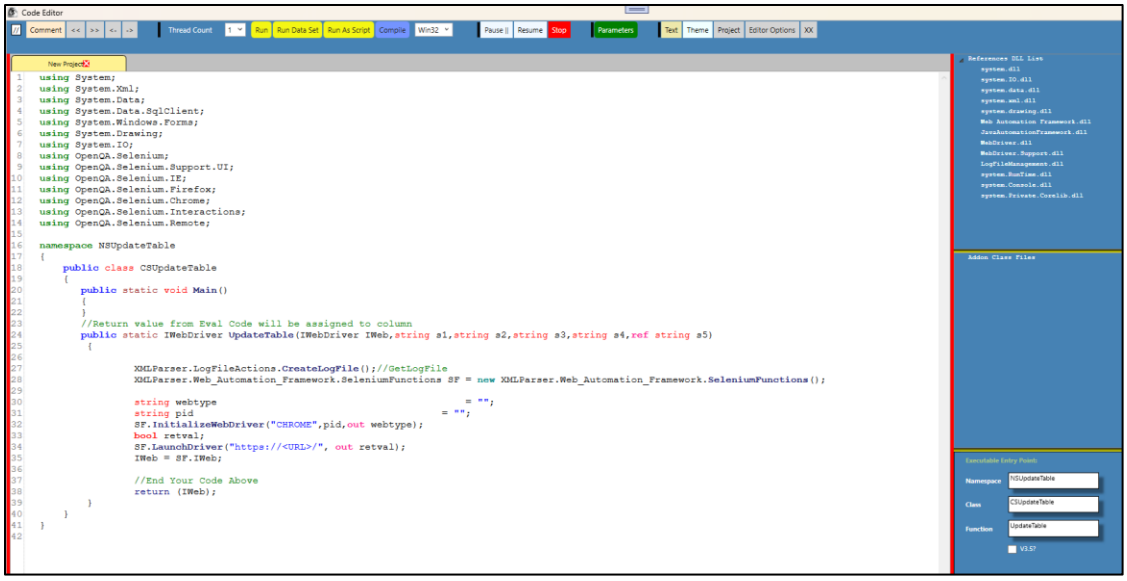
Navigate to **Tools → C# Editor**



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

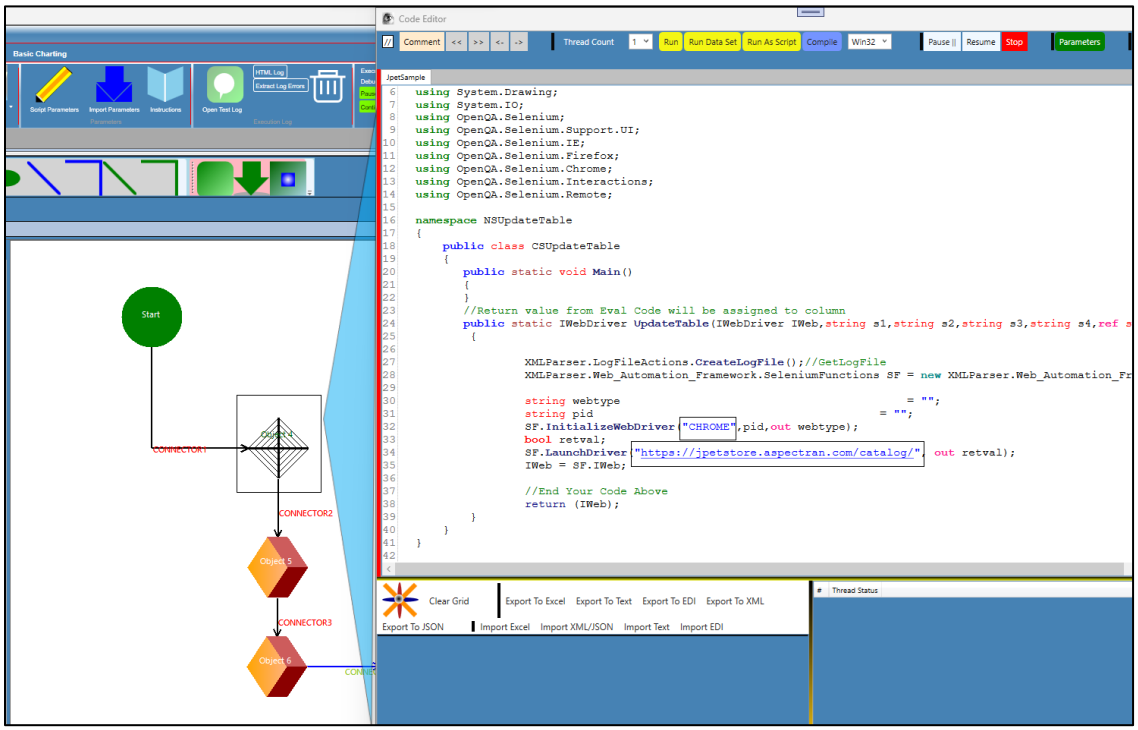


Following code will be created:

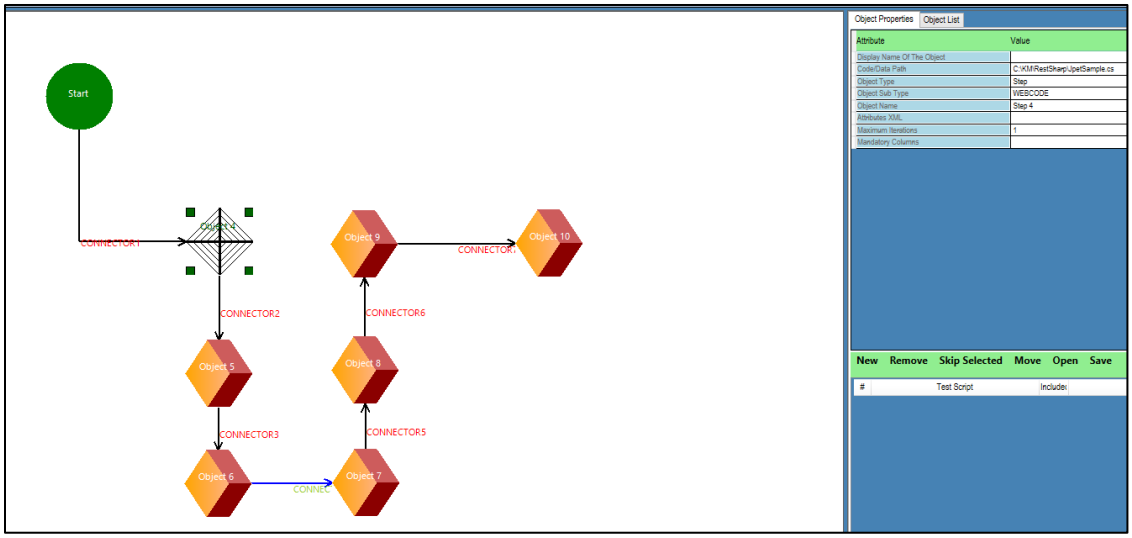


It takes, first parameter and return type as IWebDriver.

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.

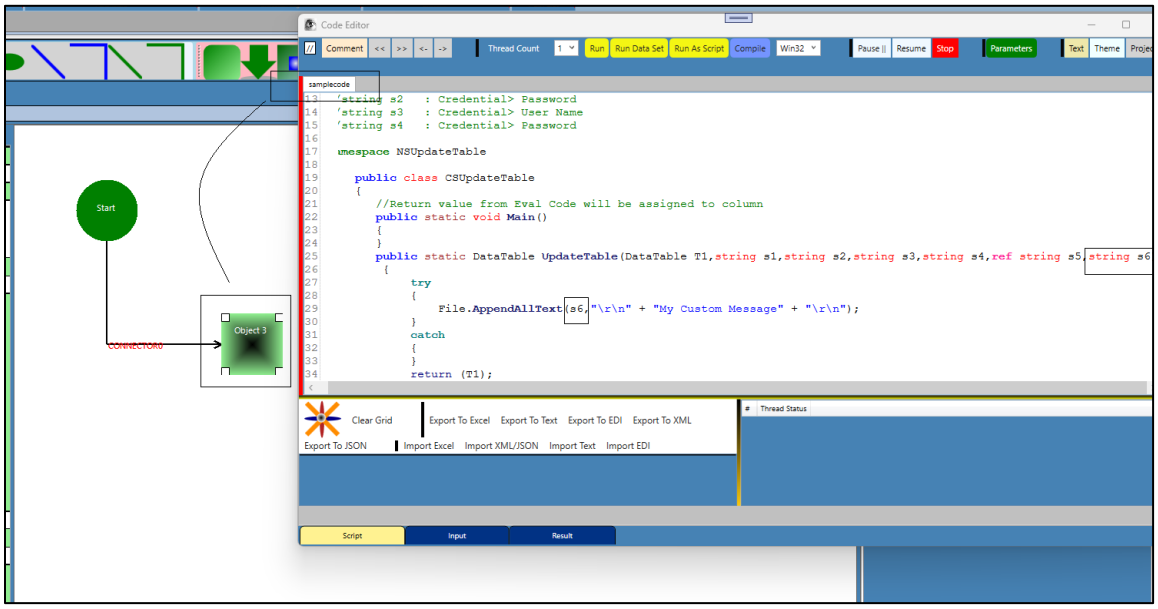


This code now can be assigned to a **WEBCODE** block used in place of usual Launch component and other components can be used just like earlier


Start
CONNECTOR1
Object 4
CONNECTOR2
Object 5
CONNECTOR3
Object 6
Object 7
CONNECTOR5
Object 8
CONNECTOR6
Object 9
CONNECTOR7
Object 10
Object Properties
Object List
Attribute
Value
Display Name Of The Object
Code/Data Path
C:\KM\RestSharp\JpetSample.cs
Object Type
Step
Object Sub Type
WEBCODE
Object Name
Step 4
Attributes XML
Maximum Iterations
1
Mandatory Columns
New
Remove
Skip Selected
Move
Open
Save
#
Test Script

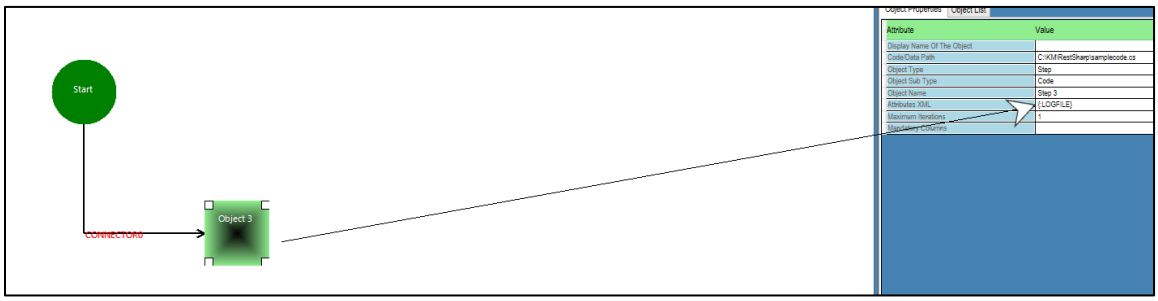
Using Additional Parameters

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.

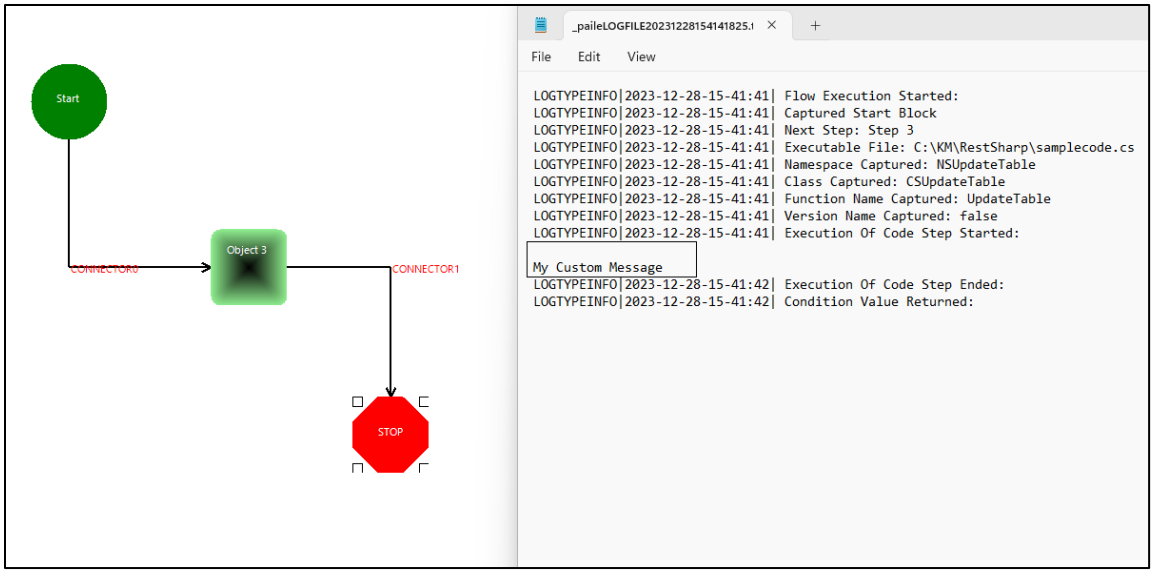


The extra parameter declared in the code needs to be added attributes in a comma separated way.

(In this case we are using only 1 parameter)

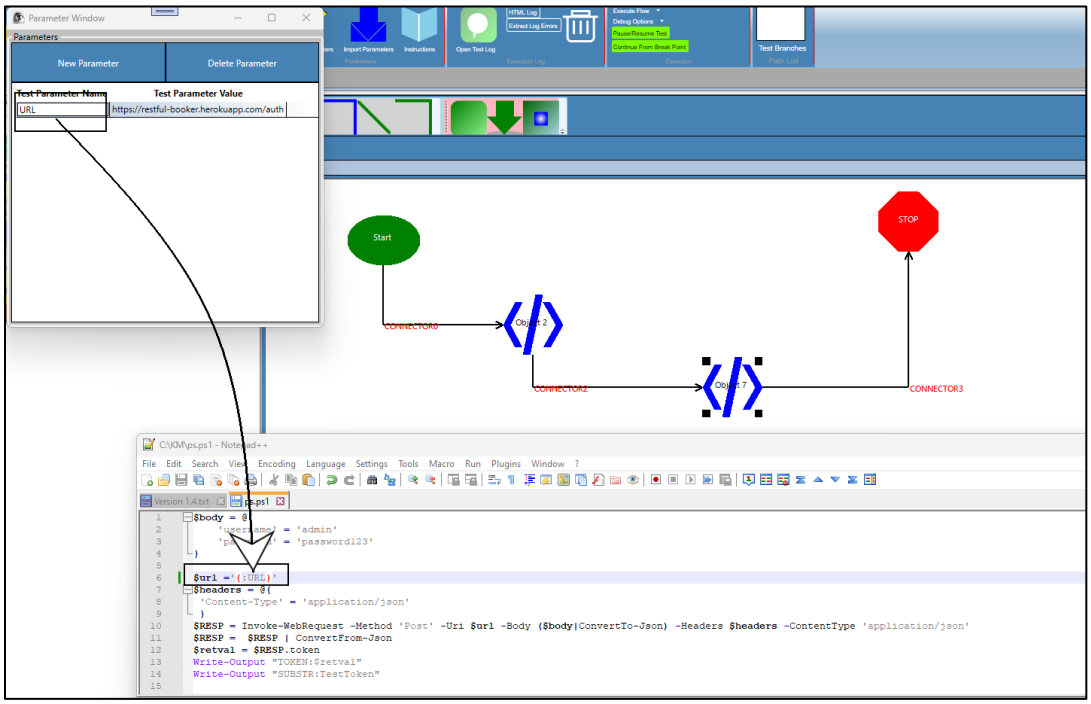


After the execution, Logfile will show:


Start
Object 3
CONNECTOR0
CONNECTOR1
STOP
_paileLOGFILE20231228154141825.l
File
Edit
View
LOGTYPEINFO|2023-12-28-15-41:41| Flow Execution Started:
LOGTYPEINFO|2023-12-28-15-41:41| Captured Start Block
LOGTYPEINFO|2023-12-28-15-41:41| Next Step: Step 3
LOGTYPEINFO|2023-12-28-15-41:41| Executable File: C:\KM\RestSharp\samplecode.cs
LOGTYPEINFO|2023-12-28-15-41:41| Namespace Captured: NSUpdateTable
LOGTYPEINFO|2023-12-28-15-41:41| Class Captured: CSUpdateTable
LOGTYPEINFO|2023-12-28-15-41:41| Function Name Captured: UpdateTable
LOGTYPEINFO|2023-12-28-15-41:41| Version Name Captured: false
LOGTYPEINFO|2023-12-28-15-41:41| Execution Of Code Step Started:
My Custom Message
LOGTYPEINFO|2023-12-28-15-41:42| Execution Of Code Step Ended:
LOGTYPEINFO|2023-12-28-15-41:42| Condition Value Returned:

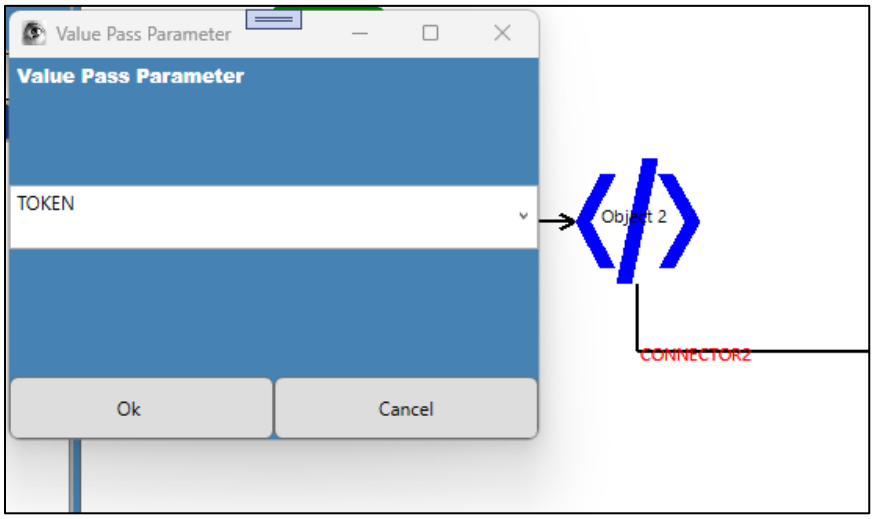
PowerShell Scripting

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.

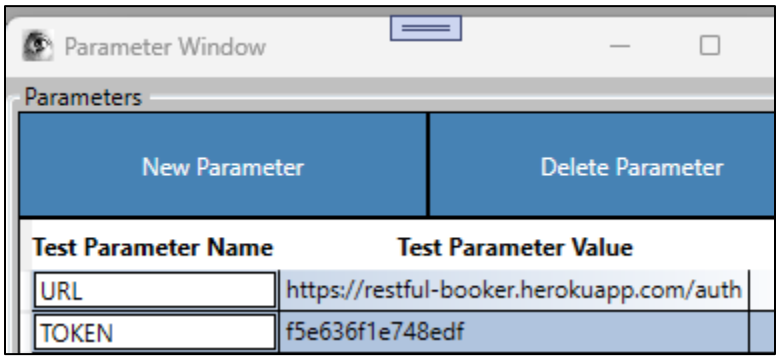
The first PowerShell component will execute the script and return TOKEN value (The pass parameter is **TOKEN** and the value Output by PowerShell which starts with **TOKEN**: (ParameterName:ParameterValue) will be assigned to this parameter



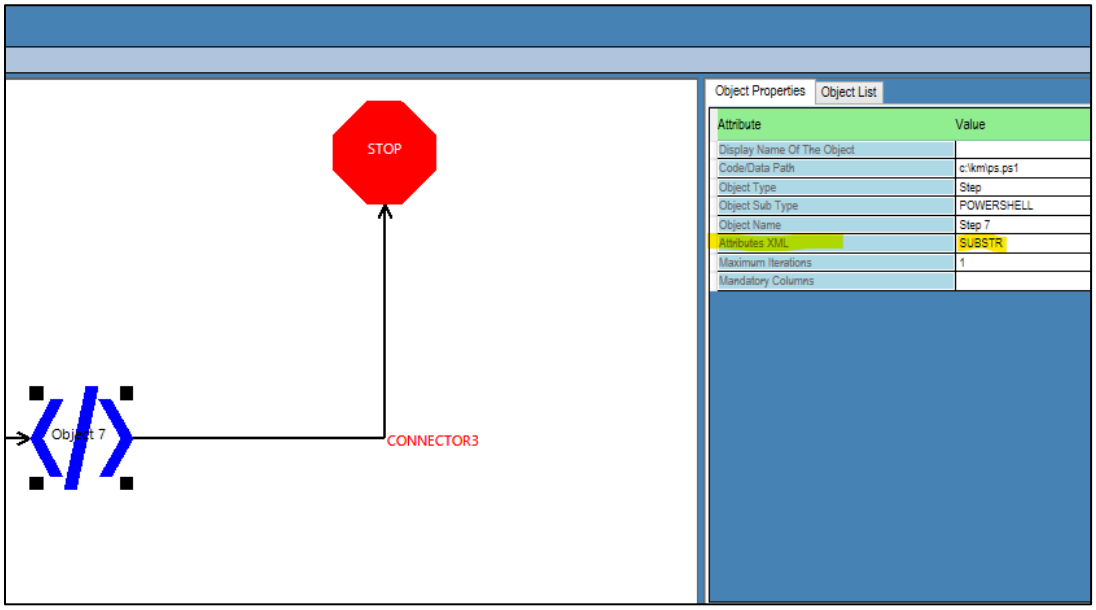
Log Shows:

```
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



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
```