

Process Studio Activity Guide

(Release 7.4.1)

© 2023, AutomationEdge Technologies Inc. All rights reserved

Information used in this document is subject to change without notice. Companies, names, and data used in the examples herein are fictitious unless otherwise noted. No part of this document can be reproduced or transmitted in any format, by any means, electronic or mechanical, for any purpose without permission of AutomationEdge Technologies.

Contents

I	Introduction	5
II	Workspace for GUI Automation	7
1	Introduction to GUI Automation for Web.....	7
1.1	Supported Browsers for GUI Automation/ GUI Spy.....	9
1.2	A note on 'Web Browser Name' in all the steps used in GUI automation for web	10
1.3	Closing Web Driver Instances	11
2	Project 1: Stock Value	12
2.1	Building your first GUI workflow for web: HelloWeb	12
2.2	HelloWeb with GUI Spy	12
3	Project 2: Multiple Stock Values	37
3.1	Making HelloWeb Generic.....	37
3.2	Loop in Workflows: Components and Features.....	38
3.3	Building your second GUI Automation for web workflow: HelloWebGeneric	39
3.4	Working with Parameters.....	51
3.5	Parameterising HelloWebGeneric	53
4	Project 3: Modularity.....	57
4.1	Building third RPA workflow with Child workflow: HelloHierarchy	57
5	Project 4: Tables & Lists	67
5.1	Workflow: HelloWebLoopTable	67
III	Publishing Workflows on AE	90
6	Publishing Workflows on AE Overview.....	90
6.1	Introduction	90
6.2	Parameters considerations	90
6.3	Publishing Workflows: Flow	92
7	Publish HelloWeb to AutomationEdge	93
7.1	Publish HelloWeb to Development Instance	93
7.2	Setup, Execute HelloWeb on Development Instance.....	115
7.3	Workflows: Export from Development Instance.....	123
7.4	Workflows: Import to UAT Instance	126
7.5	Workflows: Export from UAT Instance.....	132
7.6	Workflows: Import to Production Instance	135
IV	Workspace for Data Processing	141
8	Project 5: Sales Revenue	141
8.1	Building your first data processing workflow: HelloWorld	141
8.2	Building your first process.....	154
9	Project 6: Payroll.....	178
9.1	Build your Second project – Payroll for SalaryProcessing	178
9.2	Compute Salary Workflow	185

9.3	Send Mail Workflow	200
9.4	Employee Main Process	205
10	Project 7: Scripting	209
10.1	Building your first scripting workflow HelloJavaScript	209
V	Workspace for Tutorials	217
11	Project 8: GUI Spy for Web.....	217
12	Project 8: GUI Spy Recorder	218
13	Project 8: GUI Spy for Windows.....	224
13.1	Sample workflow with GUI Spy for Windows.....	224
14	Project 8: Desktop Spy	242
14.1	Sample Workflow with Desktop Spy.....	242
15	Project 8: Surface Spy	261
15.1	Sample Workflow with Surface Spy.....	261
16	Project 9: Try-Catch.....	281
16.1	Augmenting HelloWorld workflow with Try-Catch.....	282
16.2	Try-Catch with Loop Steps	295
16.3	Nested Try Catch	297
16.4	Error Handling Hop	299
17	Project 10: Debugging	300
17.1	Create a workflow and setup Debugging features	300
17.2	Set Breakpoints	303
17.3	Debug	310
17.4	Debug All Rows	319
17.5	Debug Project	321
18	Project 11: Git Repository Integration.....	326
18.1	Repository.....	326
18.2	Commit.....	332
18.3	Push.....	333
18.4	Pull.....	334
18.5	Branches.....	335
18.6	Merge Scenarios.....	339
18.7	Visual Diff.....	348
19	Project 12: SVN Repository Integration	353
19.1	Download and Install Subversion (SVN) Client	353
19.2	Create Project and Checkout SVN repository	354
19.3	Open SVN repository in Process Studio.....	361
Contact	377

I Introduction

AutomationEdge Process Studio is one of the three major components of AutomationEdge (henceforth called AE) platform. Other two being AutomationEdge Agent and AutomationEdge Server. Process Studio is a Java based desktop tool meant for creating or modelling business processes. It offers drag-and-drop capabilities that almost eliminates need to write a program or a script while building business processes. AutomationEdge Process Studio has a rich library of plugins to access, prepare, and blend data from disparate data sources and enterprise applications. It also provides an ability for customers to write their own plugins to extend this library.

A business process can be modelled in Process Studio either as a workflow or a process. A workflow is about moving and transforming rows whereas a process is more about high level flow control: executing workflows, sending mails on failure, transferring files via FTP etc. Another key difference is that all the steps in a workflow execute in parallel, but the steps in a process execute in order.

Course Design:

The course consists of two guides,

- AutomationEdge_R7.4.1_ProcessStudio_User_Guide
- AutomationEdge_R7.4.1_ProcessStudio_Activity_Guide

1. AutomationEdge_R7.4.1_ProcessStudio_User_Guide

The guide provides a comprehensive overview of all the components of Process Studio. Also use this guide for Getting started with Process Studio by launching Process Studio and synchronization with AutomationEdge.

2. AutomationEdge_R7.4.1_ProcessStudio_Activity_Guide

The guide provides hands on exercises. This includes designing and building four Workspaces with eleven projects as listed below,

1. GUI Automation

GUI Automation has five projects with workflows covering various concepts of design and artefacts for GUI RPA for Web.

- Single Stock Value
- Multiple Stock Values
- Modularity
- Tables & Lists

2. Publishing Workflows to AutomationEdge

Publish Process Studio projects to AutomationEdge server. This section discusses the lifecycle of publishing projects as AutomationEdge workflows. The sample project is from the GUI Automation workspace,

- Single Stock Value

3. Data Processing

Data Processing has three projects with workflows covering various concepts of design and artefacts for data processing with Process Studio.

- Sales Revenue
- Salary Processing

4. Tutorials

The Tutorials workspace has five projects and covers common tools/features of Process Studio used across workflows/processes.

- Spy (GUI Spy for Web/Windows, GUI Spy Recording, Desktop and Surface Spy)
- Try-Catch
- Debugging
- Git/SVN

The Process Studio - Exercise Projects are also bundled with AutomationEdge Product documentation.

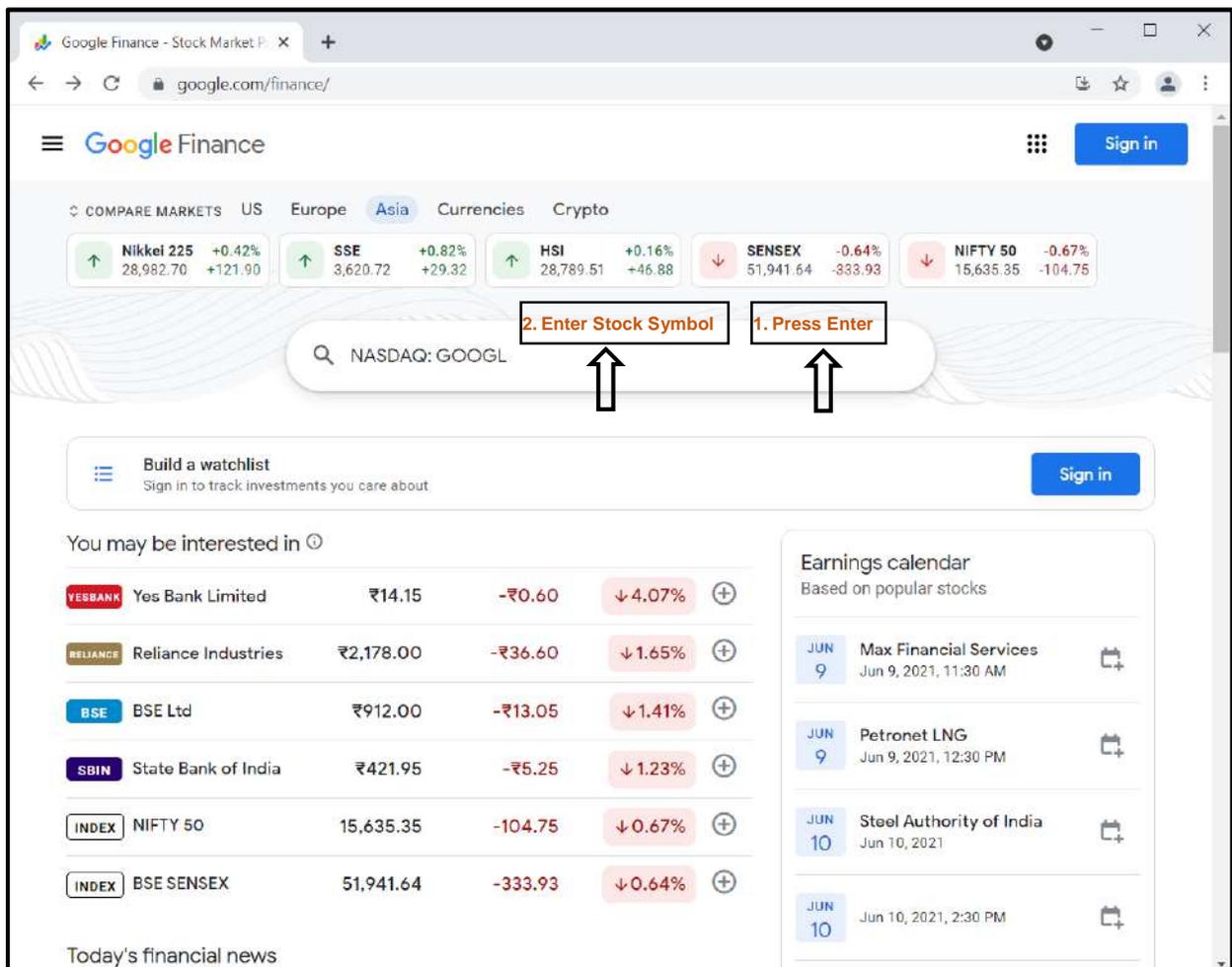
II Workspace for GUI Automation

(Robotic Process Automation(RPA) GUI Automation for Web)

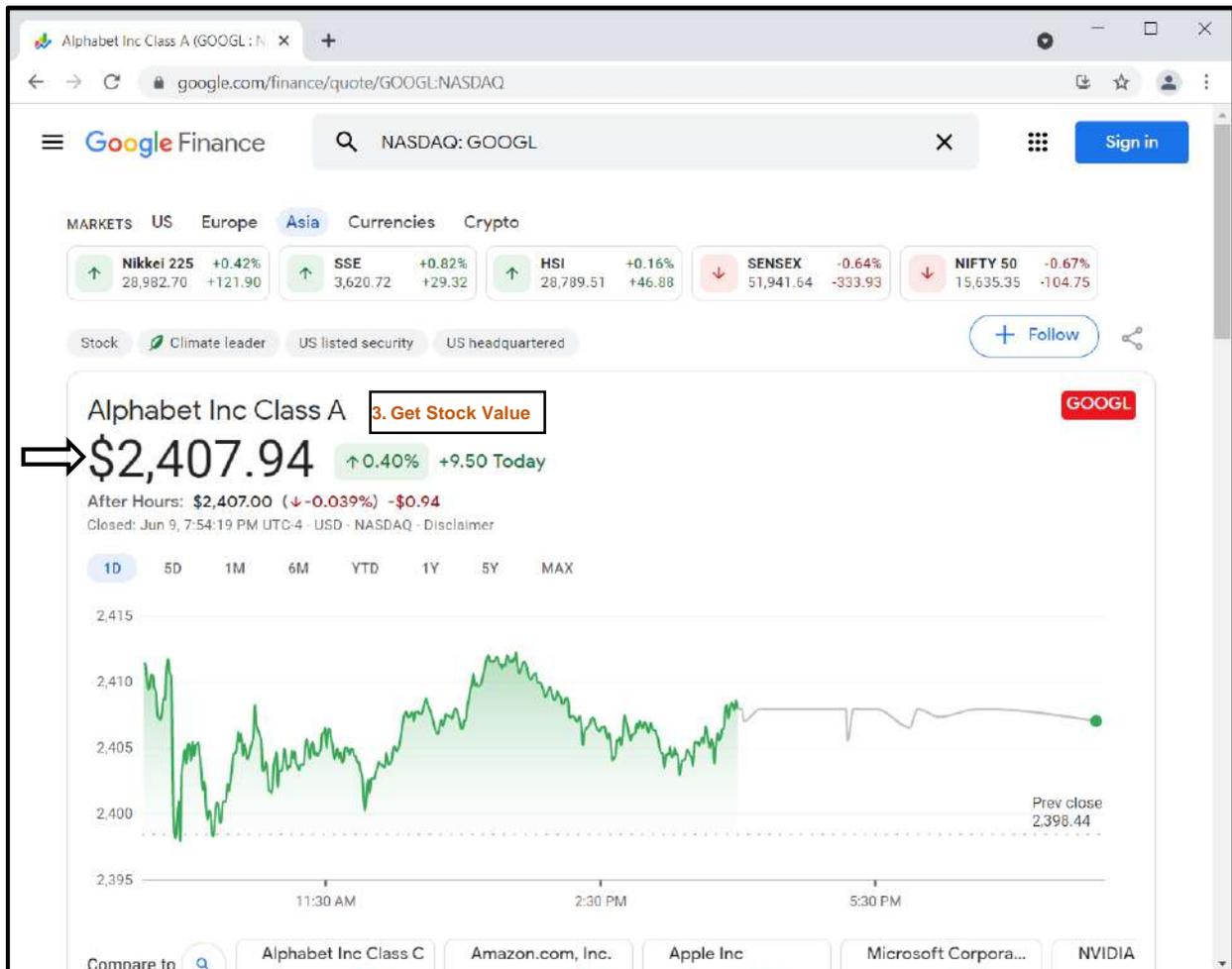
1 Introduction to GUI Automation for Web

When we automate web applications, we actually launch a web browser and mimic operations on it as would be carried out by a human being. For example, this is what we do to get current stock price of a company on <https://www.google.com/finance> site. We enter value of stock symbol in the search text box, we press enter and get the price of the stock from the search result.

1. On the <https://www.google.com/finance> page enter the stock symbol and press enter.



2. On the search results window fetch the stock value.



In order to automate this, we need to

- Find search text box and enter stock symbol and press Enter
- Find element where price is shown in search result and extract its value

We essentially find an element in the web browser and perform some action on it. Finding an element in the web application is the key for GUI automation for web. Below is the list of different ways (criteria) for finding a web element.

1. Xpath

Xpath defines the hierarchy of the tags through which you can reach to a particular element. For example, the **span** HTML element which shows price of the stock in the google finance example that we saw above, can be uniquely identified by Xpath

`/html/body/div[1]/div/div[3]/div[2]/div/div[2]/div/div/div[2]/div[2]/div[1]/div[1]/div[1]/span/span`

2. Id

Id attribute of a web element can be the best way to identify an element if it is present and is going to be constant and unique. There are cases however when it is either absent or its value changes every time you access the application in the browser.

3. Name

Name is another attribute of a web element which can be used to identify an element.

4. Link Text

If a HTML page contains a hyperlink element `Search On Google`, you can identify this href element by providing complete text 'Search On Google' as the Link value.

5. Partial Link Text

This criteria is same as **Link**, but in this case, you can give partial string for finding a match.

6. Tag Name

Every HTML element has a tag associated with it. The element can be located using that tag name. For example, textbox has a tag `<input>`.

7. Class Name

Class name is the css class associated with an HTML element. If there is a div element `<div class='someclass'>MyDiv</div>`, you can locate it by giving class name 'someclass'.

8. Css Selector

This Criteria is useful when an HTML element has multiple associated css classes. Consider an HTML element `<div class="btn primary-btn submit"></div>`. To identify this div element you can use Css selector as ".btn.primary-btn.submit"

 Note: What criteria you use to identify a web element depends totally on the web application in question. "Id" is always the most preferred way of locating a web element if it is present and is going to be unique. It is possible that criteria result in more than one matched web element. In such cases you might want to use position of a web element in the returned array to get to element you desire.

1.1 Supported Browsers for GUI Automation/ GUI Spy

The following browser versions are supported in Release 3.4

Browser Support	OS	Chrome	Firefox	IE
Release 3.4	Windows	75 to 91	83 to 89	8 to11
	Linux	80 to 91	83 to 89	-

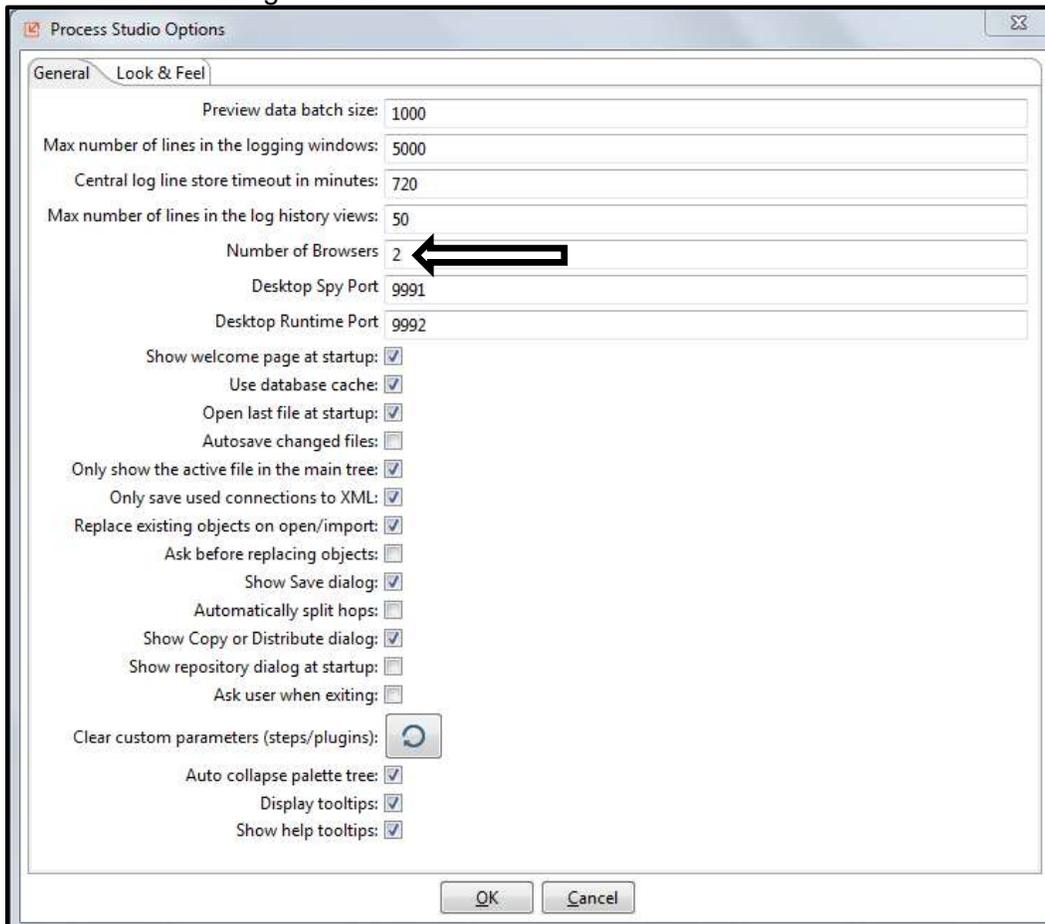
*Chrome Drivers are available on EPD. Download the required version and the System Administrator can upload it via File Management menu, on AutomationEdge Server.

1.2 A note on ‘Web Browser Name’ in all the steps used in GUI automation for web

Start Browser is the step which lets you launch a browser to start any GUI automation for web. One of the configuration fields of this step is **Web Browser Name**. You can set its value either as **BROWSER1** or **BROWSER2**. This parameter basically identifies your browser instance. All the steps to be executed on same browser instance need to have same web browser name as in **Start Browser** step. Typically, you would not need to have two (or more) values for **Web Browser Name**. If your GUI automation for web involves simultaneous use of two applications running in two browsers (say you copy something from one web application and use that value in another web application) you will have to appropriately use **Web Browser Name** in all the steps.

1.2.1 Changing number of identifiers for Web Browser Name

You will rarely come across a use-case where you need to interact at the same time with more than two browsers (running more than two applications). But if you need to increase number of identifiers for Web Browser Name, you can select **Tools >> Options...** from Process Studio menu and change the number as shown in the image below.



1.3 Closing Web Driver Instances

Start browser initializes the web driver. The web driver is closed by Exit Browser. In case you close the browser directly the web driver is still initialized; and if you try to run any new workflow that initializes a web driver it throws an error.

You can clear all browser instances in two ways:

- Navigate to Tools→Clear All Browser Instances and click .
- Enable Clear All Browser Instances checkbox in Start Browser step.

Ideally you should have a practice of having an Exit Browser step as an error handling step for all GUI Automation for web steps.

This is explained in the two sections below.

1.3.1 Browser session is kept open in PS for development productivity

Process Studio is a development environment. GUI Automation for web, workflows may fail during development process; for example, if your GUI Automation steps log into an application and the browser terminates abnormally it does not get a chance to perform logout steps. When you run the workflow again it may not perform as expected: it will not be able to login as the application is already logged in. Since the browser will remain open with this release user has the flexibility to logout before running the workflow from Process Studio.

Hence, an enhanced development feature has been provided. The Process Studio initialized browsers remain open and only exit when browser is closed with Exit Browser step, manually or Process Studio is closed. Also, since you do not have to initialize browser every time it saves initializing time every time the workflow is run.

1.3.2 Tools option to Clear all initialized browser Instances

Note: When a GUI Automation for web, workflow is run on an agent the entire browser session implicitly closes after the end of the workflow even if Exit browser is not put at the end of the workflow. However, when a GUI Automation for web, workflow is run from Process Studio the entire browser session does not close implicitly,

- leaving the browser open even when there is an error during development and browser is manually or abnormally terminated.
- In case you manually terminate a browser the browser driver is still initialized and when you run the workflow the start step will give an error as the browser is already initialized.
- Similarly, in case browser terminates abnormally then you need to clear browser driver instance before you can start a new browser session.

You can clear browser driver instance by navigating to Tools→ Clear All Browser Instances and clicking Clear All Browser Instances

1.3.3 Checkbox in ‘Start Browser’ to Clear all browser Instances

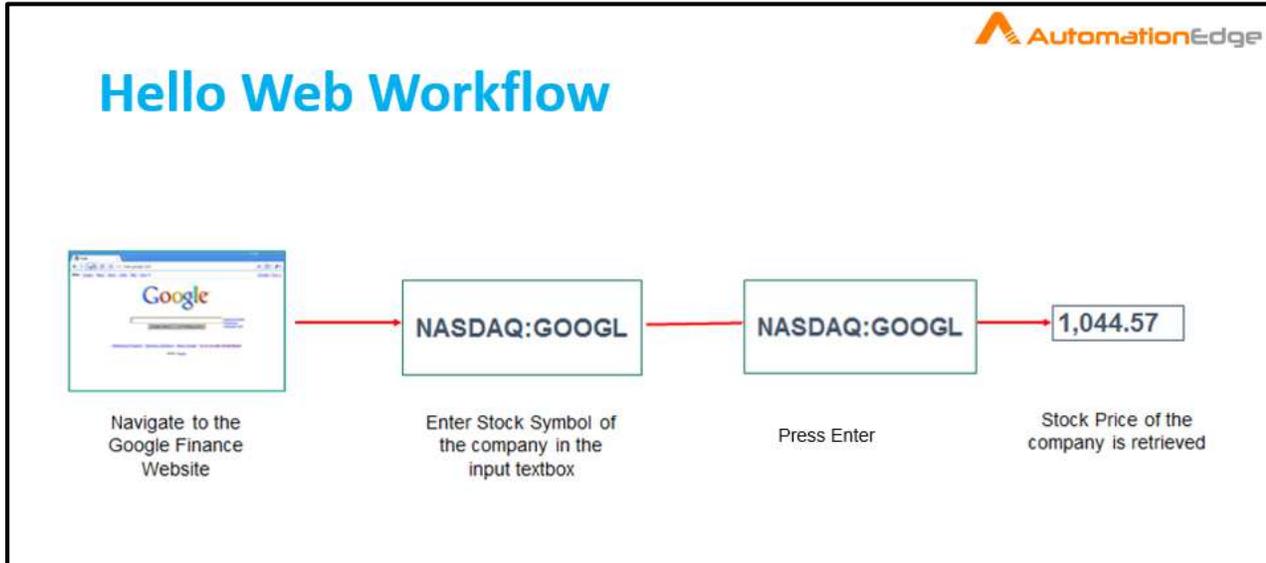
You may also clear browser driver instance by enabling the Clear all browser instances checkbox in ‘Start Browser’ step.

2 Project 1: Stock Value

2.1 Building your first GUI workflow for web: HelloWeb

We will now build a workflow HelloWeb to automate google finance example we discussed earlier.

HelloWeb workflow can be summarised in the screenshot below.



2.2 HelloWeb with GUI Spy

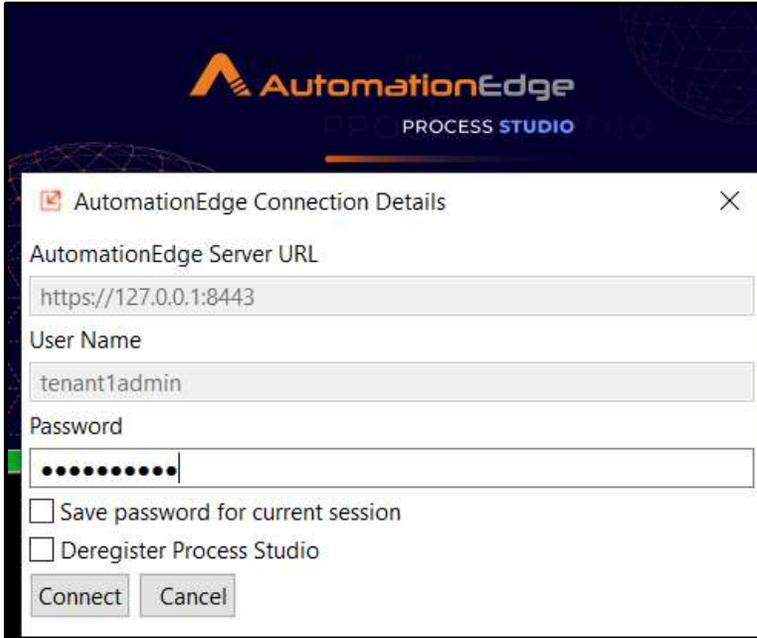
This section is a step by step demonstration of creating a workflow using GUI Spy for web. Follow the steps below to build HelloWeb workflow,

2.2.1 Launch Process Studio and Create a Project

1. Start Process Studio by right-click open or double click of the application.



2. Provide AutomationEdge Connection Details to connect to AutomationEdge server.



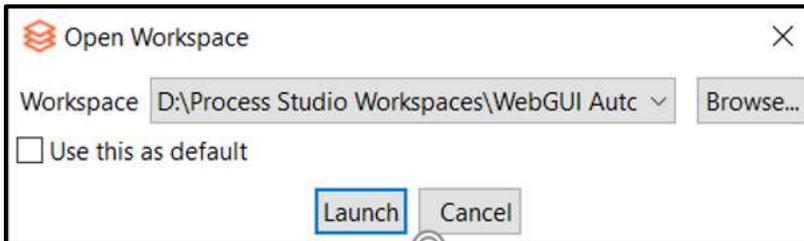
The screenshot shows the 'AutomationEdge Connection Details' dialog box. It contains the following fields and options:

- AutomationEdge Server URL: `https://127.0.0.1:8443`
- User Name: `tenant1admin`
- Password: `.....`
- Save password for current session
- Deregister Process Studio
- Buttons: **Connect**, **Cancel**

3. Identify a folder you wish to designate as your Workspace.

Name	Date modified	Type	Size
 WebGUI Automation	28-10-2020 19:04	File folder	

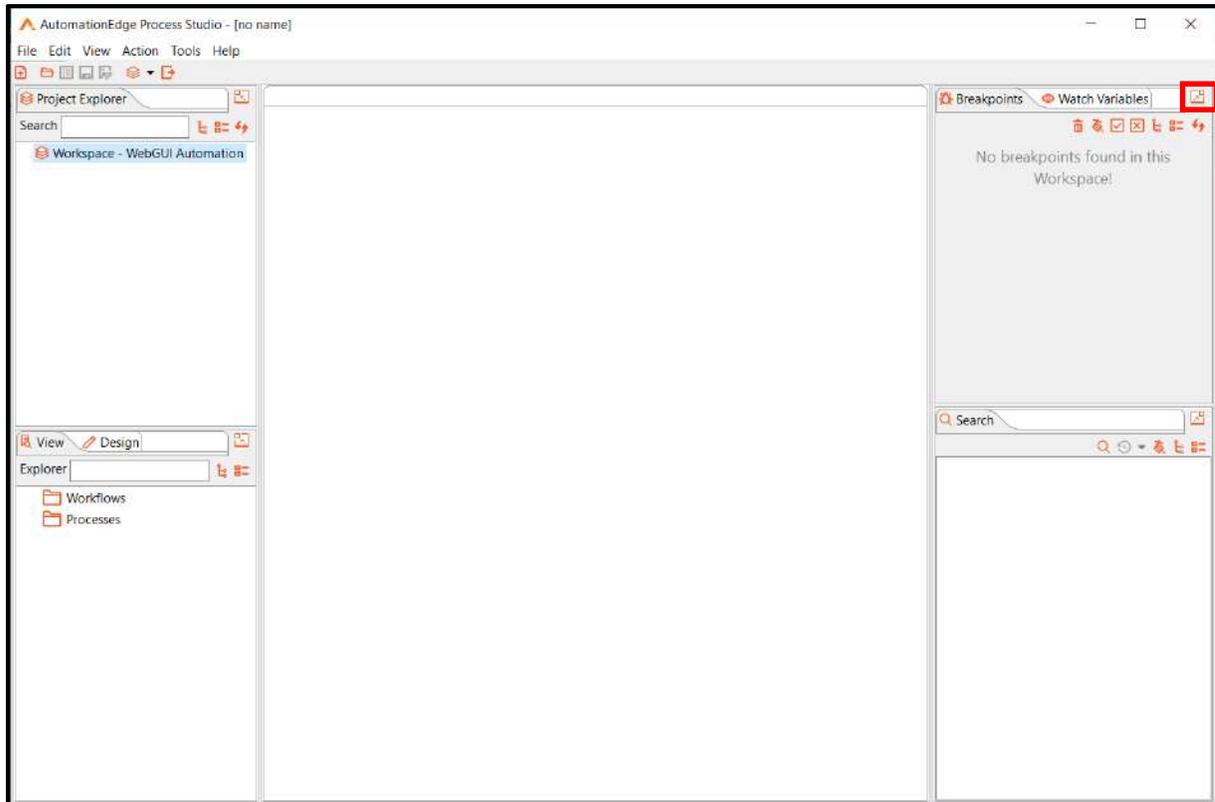
4. Upon connect Open Workspace popup appears. Browse your Workspace folder.
5. Click Launch.



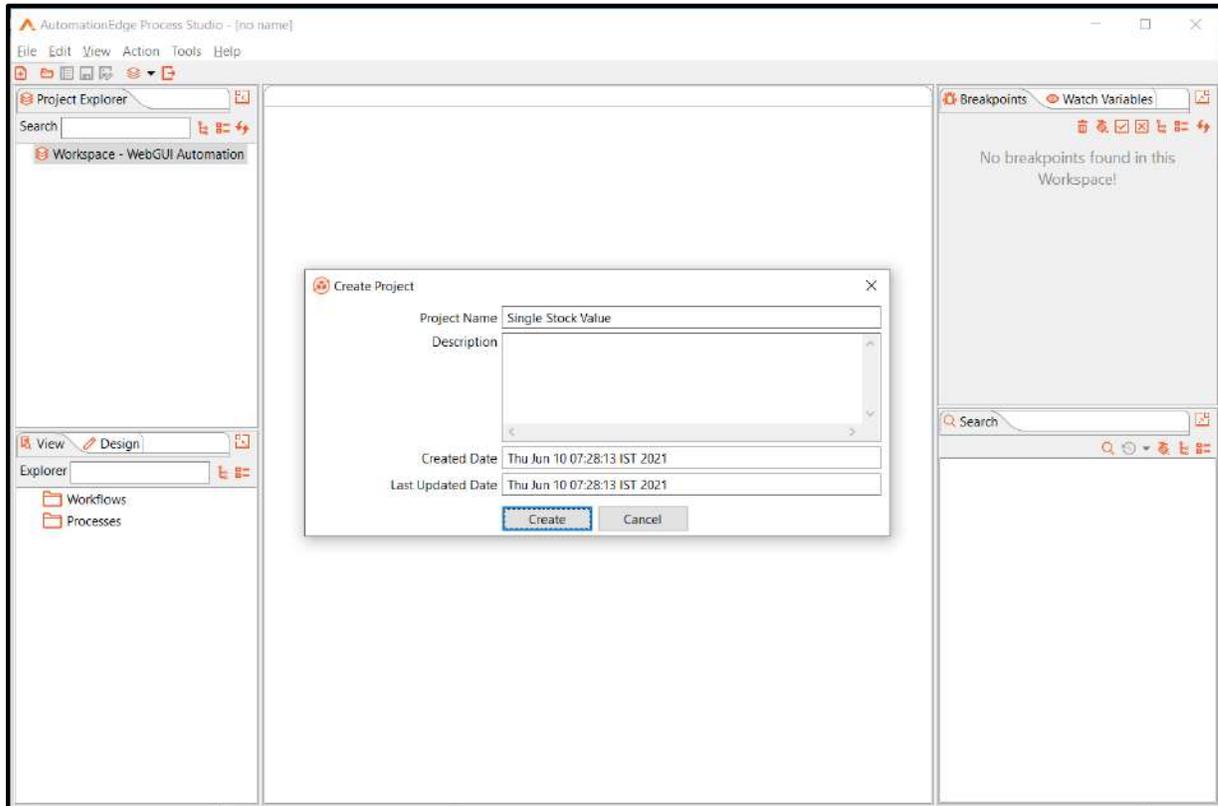
The screenshot shows the 'Open Workspace' dialog box. It contains the following fields and options:

- Workspace: `D:\Process Studio Workspaces\WebGUI Autc` (with a dropdown arrow) and a **Browse...** button.
- Use this as default
- Buttons: **Launch**, **Cancel**

- The screenshot below shows the WebGUI Automation workspace open in Process Studio.
- You may click the icon at the top right corner of the Breakpoints pane to next to Watch Variables to minimise the Breakpoints pane.



8. Right click on the Workspace and click New Project. Create a project called 'Single Stock Value'.

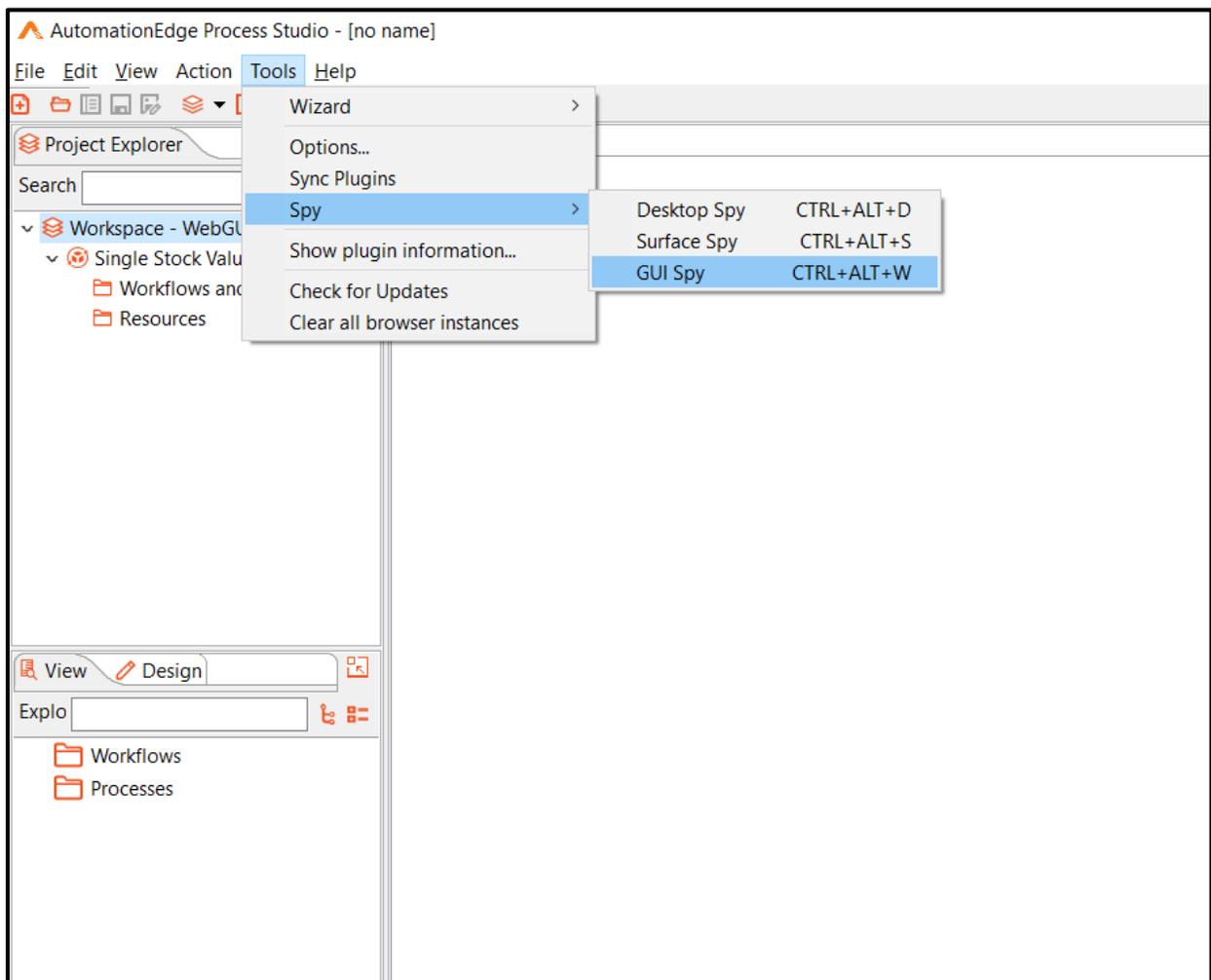


We will follow two approaches

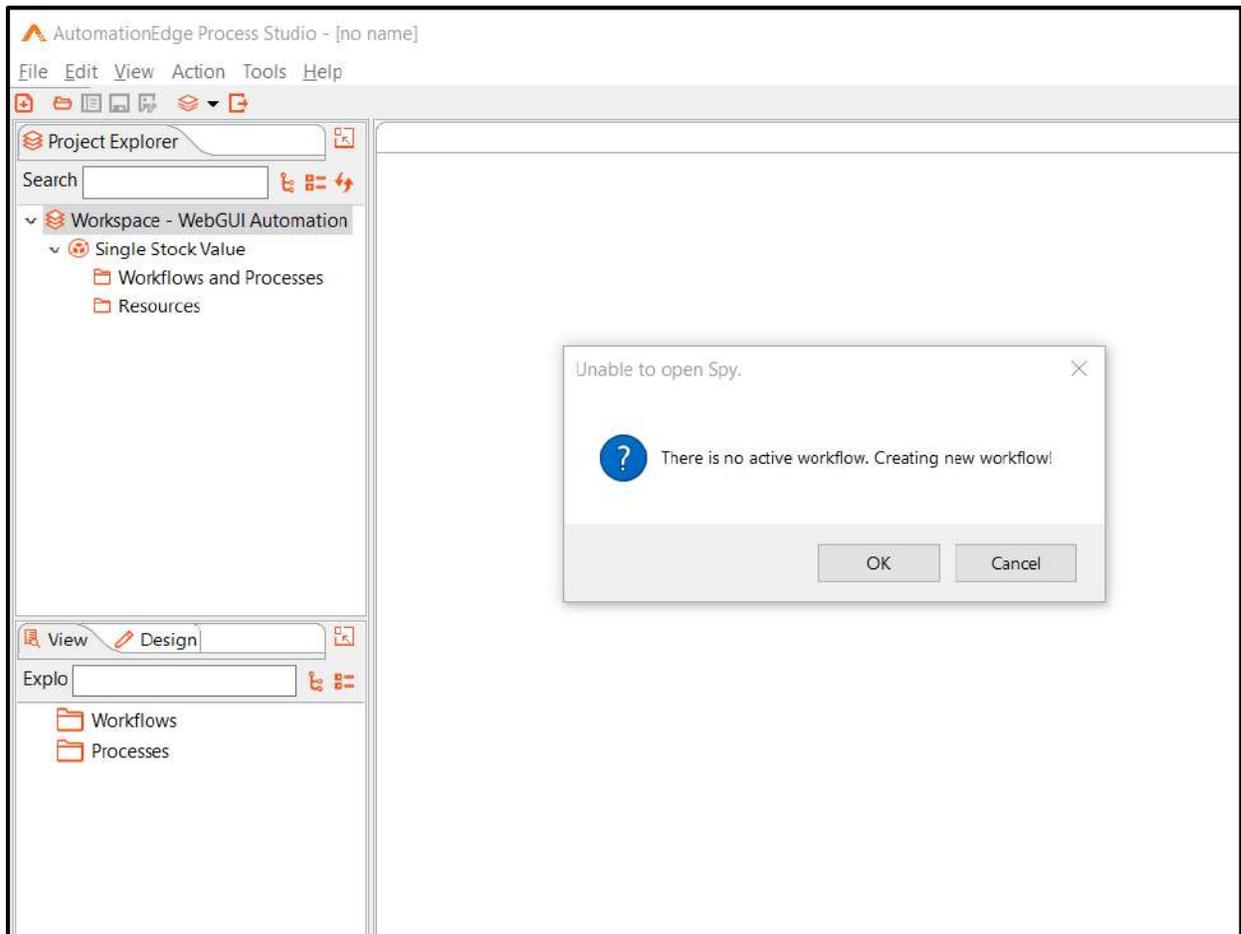
- Directly start with GUI Spy to create a workflow and Generate other steps with GUI Spy which automatically add the Start Browser step as well.
- Alternatively, Manually create a workflow, add a Start Browser step and Generate other steps with GUI Spy.

2.2.2 Create a new workflow with GUI Spy

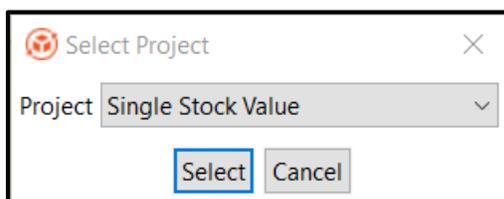
9. From the Tools→Spy menu start GUI Spy.



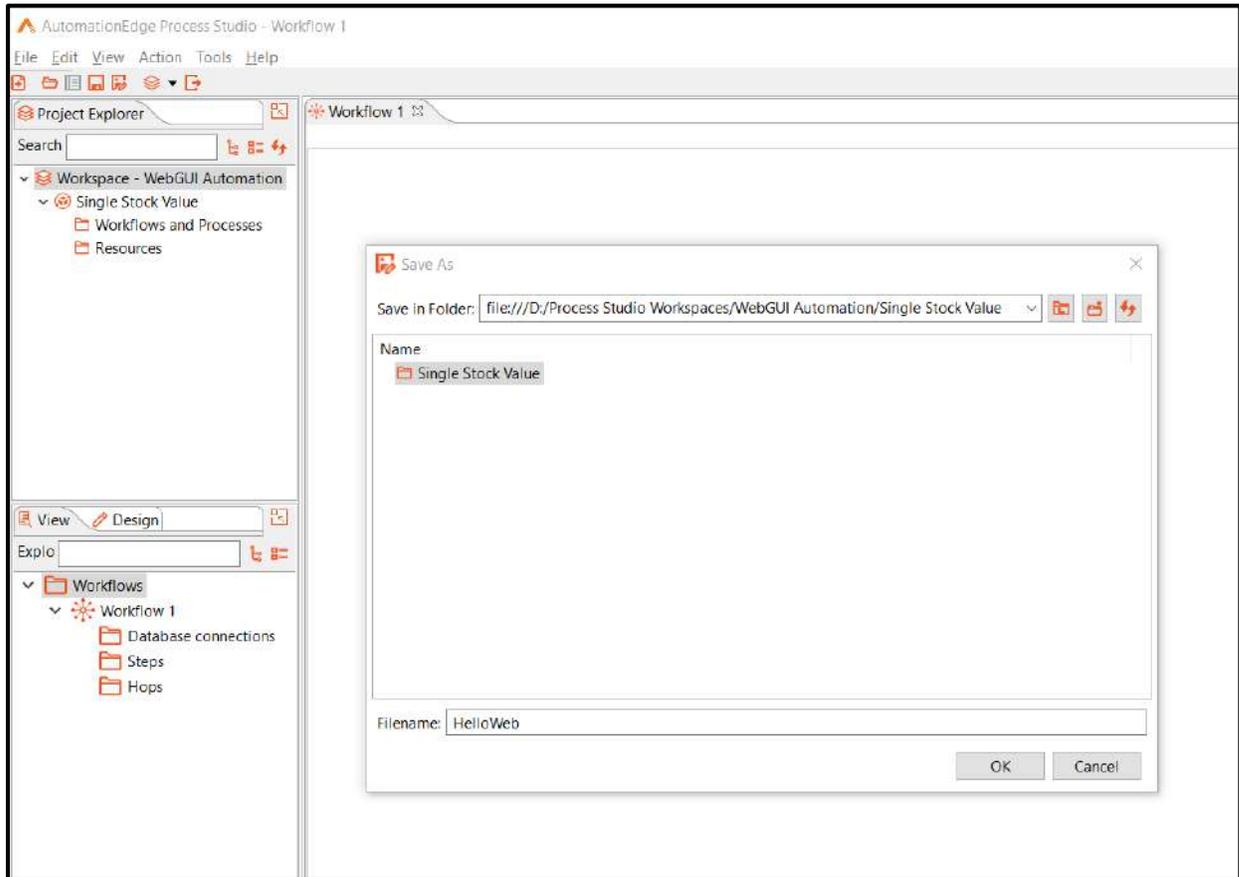
10. It displays a popup, 'There is no active workflow. Creating new workflow!'. Click OK.



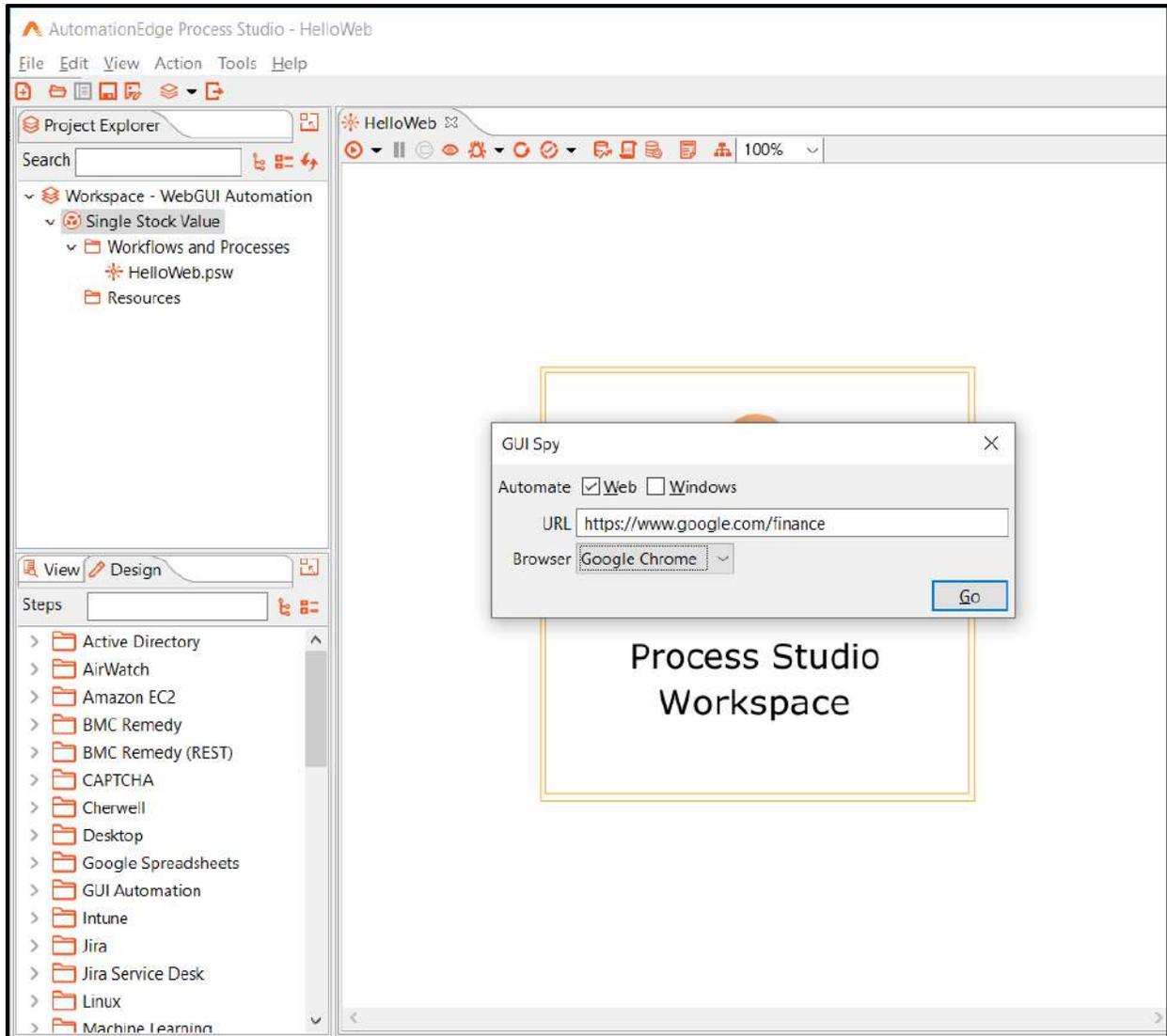
11. A pop up to Select Project appears. Select a project in which you wish to create the workflow.



12. Provide a name for the new workflow. Click OK.



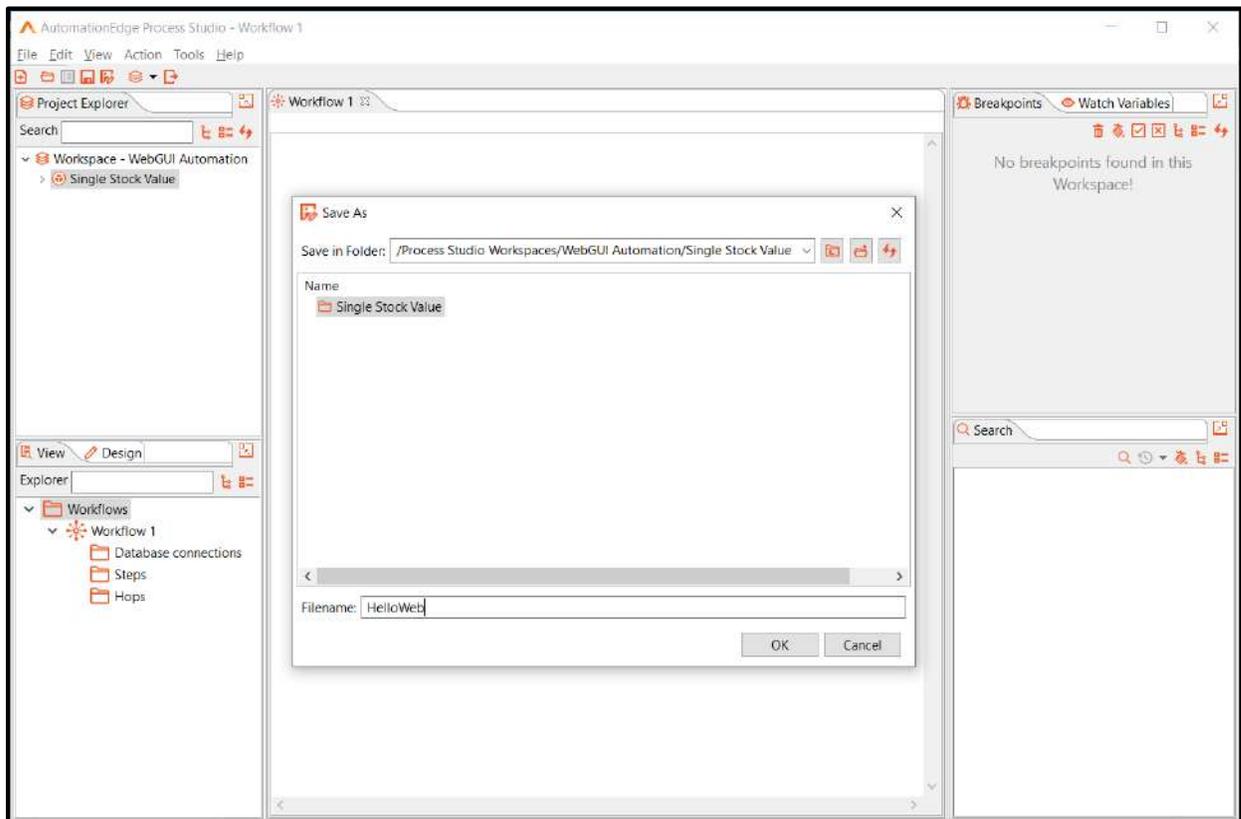
13. Once the workflow is created, the GUI Spy popup appears. Enable the Web checkbox only to Automate. It will automatically add a Start Browser as the first step upon completion of GUI Spy activities. Continue with steps in section Continue with GUI Spy.



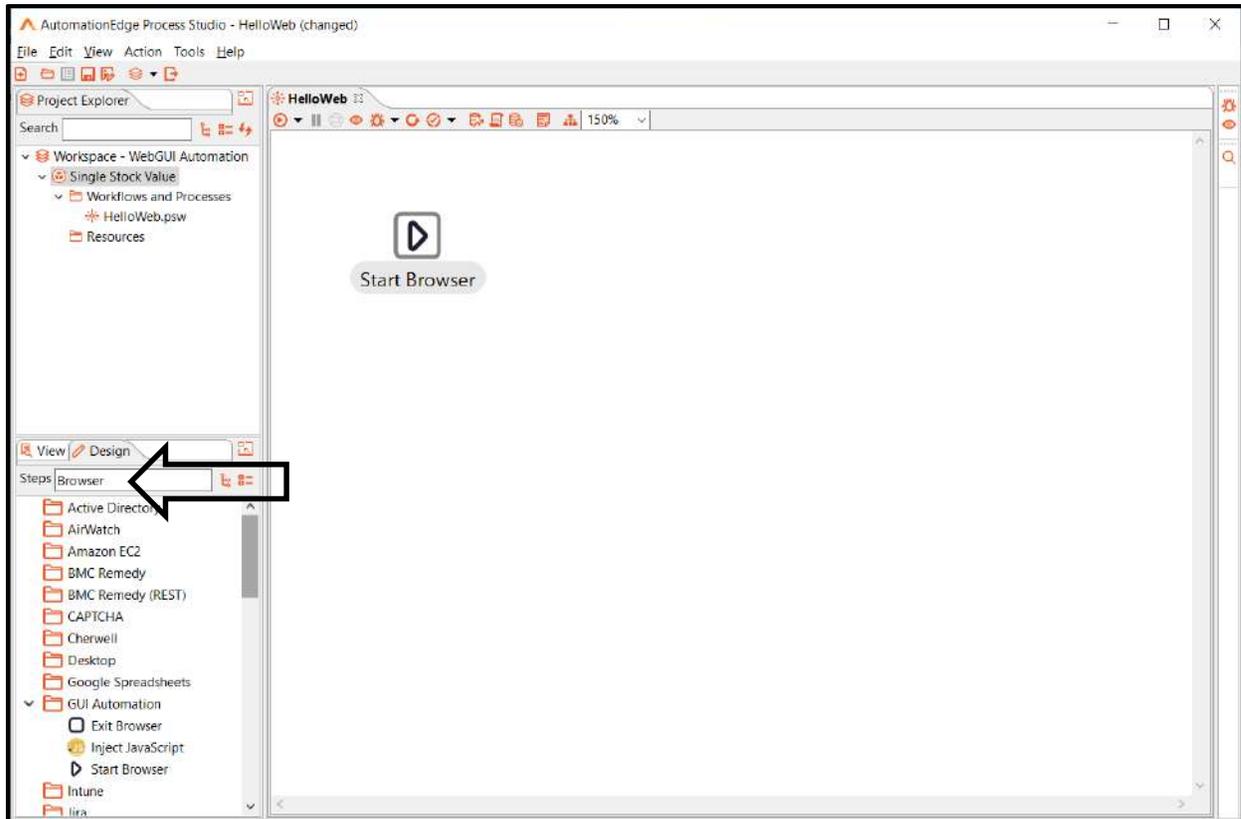
2.2.3 Create a new workflow manually

In this section we will use the second approach; following are the steps to create a workflow manually, add a Start Browser step and then Start GUI Spy.

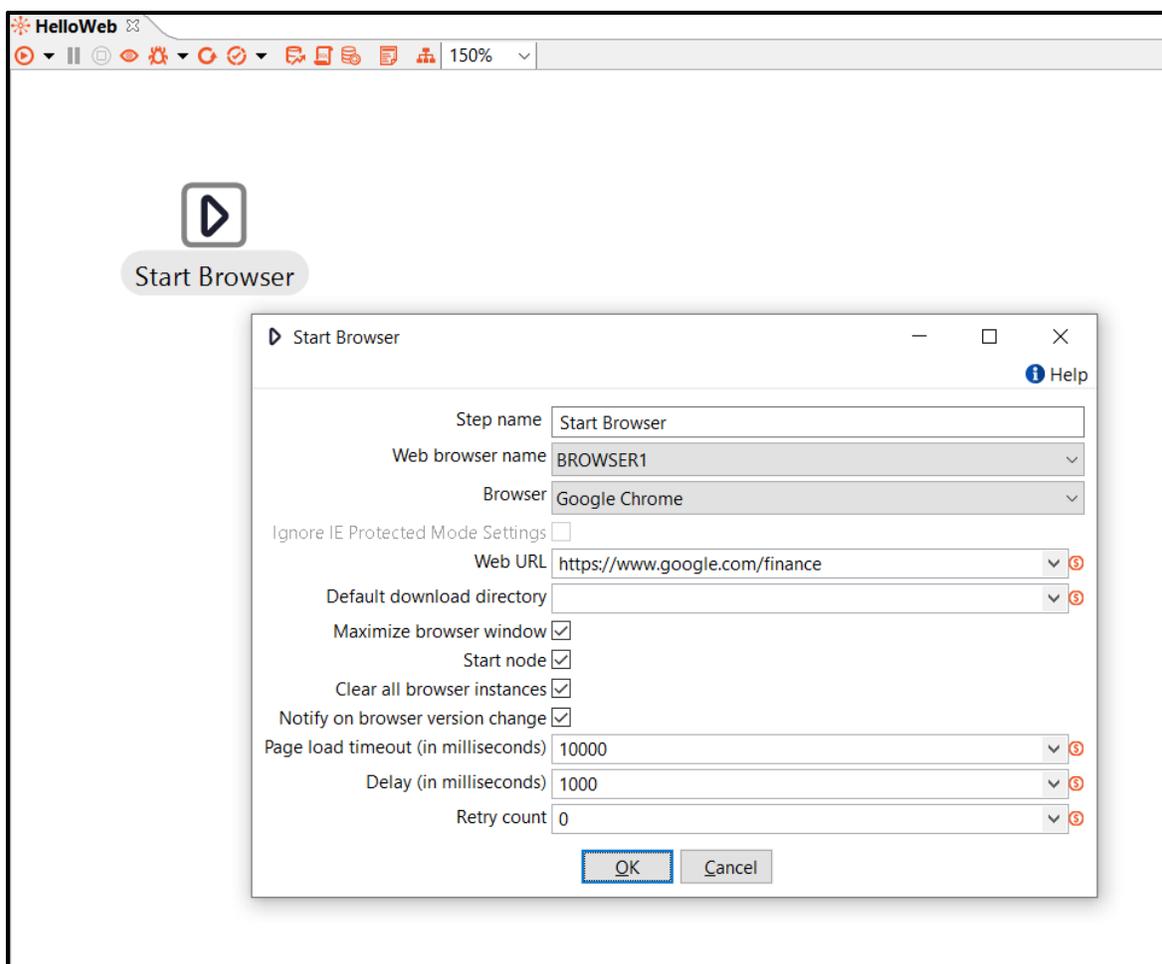
14. Right Click on Workflow and Processes under 'Single Stock Value' project. Select New→Workflow.
15. Create a **Workflow** named **HelloWeb**.
16. The Workflow is saved with an extension psw which stands for Process Studio Workflow. It is saved in the directory location of the Workspace.



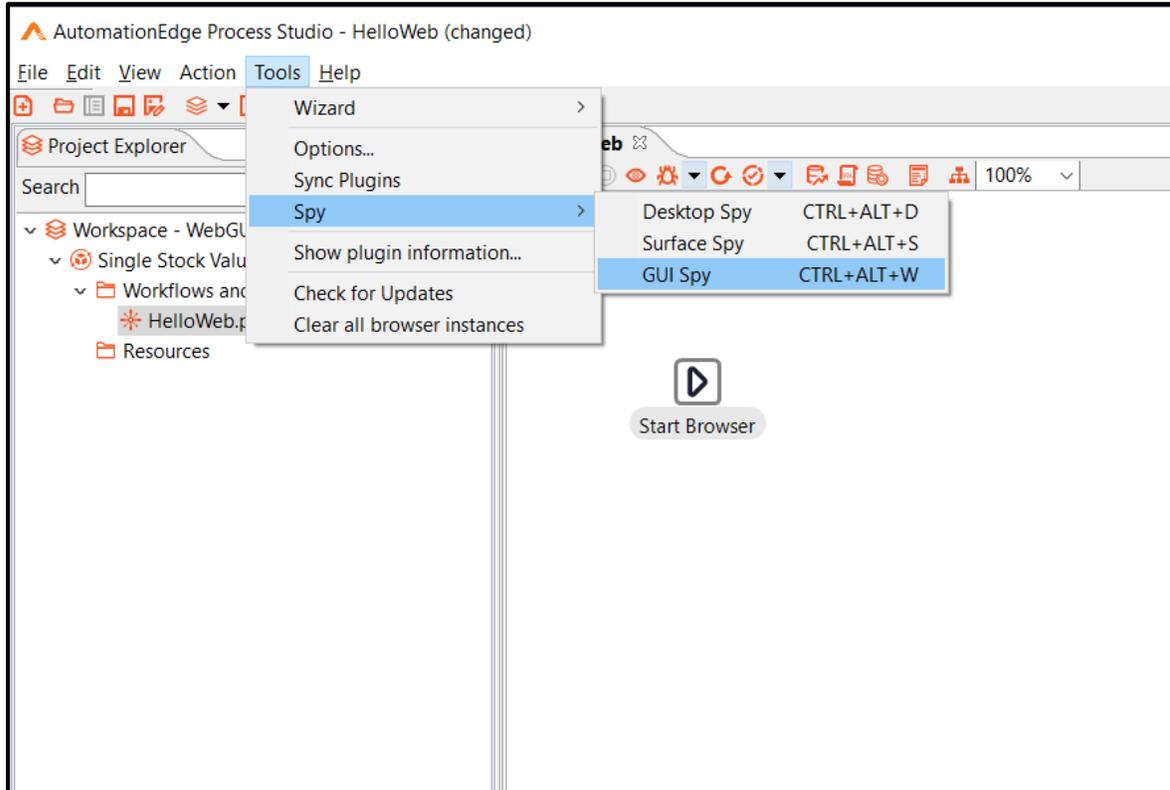
17. Add the Start Browser step manually.
18. Go to the design tab. It is the navigator for Plugin steps. Type Browser in the text box next to **Steps** in the left pane. All the steps having Browser in their name will be displayed.



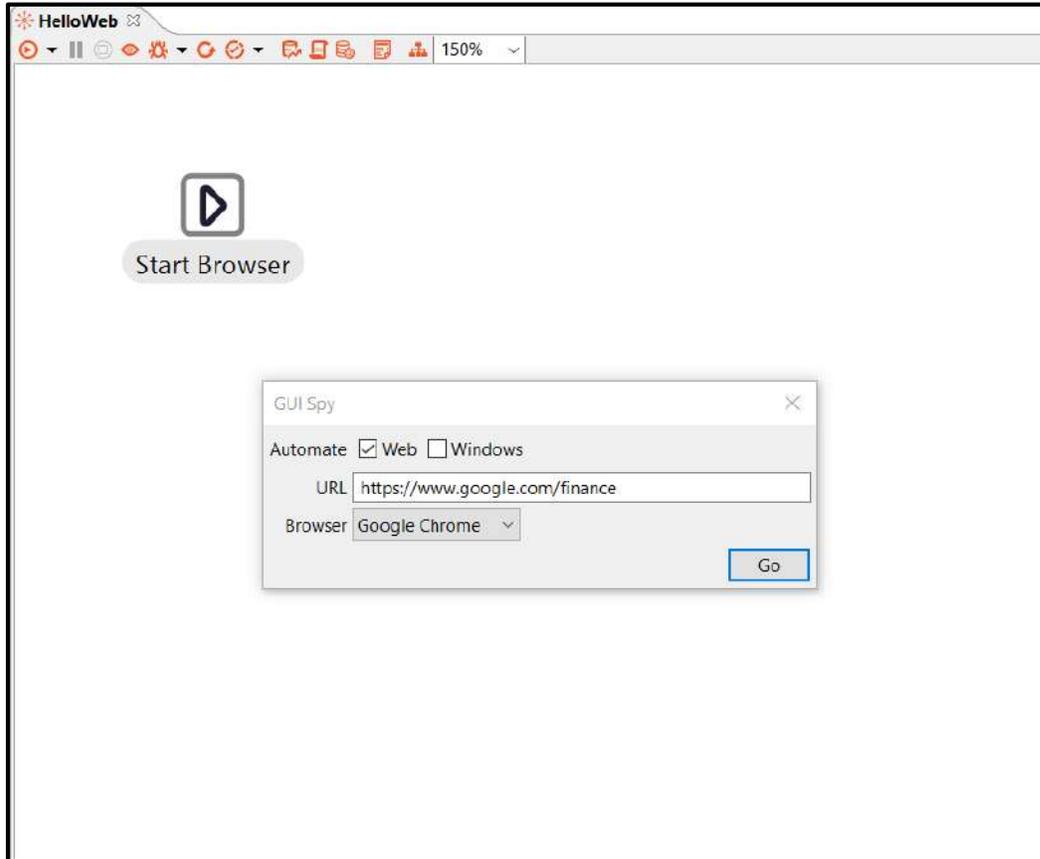
19. Now drag a step named **Start Browser** from the left pane and drop it in the workspace area in the right pane. Double click on this step to edit its properties. Configure this step as shown in the image below. We are essentially specifying browser (make sure browser you select is installed on your machine) and the URL browser should visit when it starts. Use <https://www.google.com/finance> as the Web URL. **Process Studio supports Firefox (version below 47), Chrome (version 56 upwards) and IE (version 11).**
 Note: Start browser initializes the web driver. The web driver is closed by Exit Browser. In case you close the browser directly the web driver is still initialized; and if you try to run any new workflow that initialises a web driver it throws an error. You can clear all browser instances by navigating to Tools→Clear All Browser Instances. Or you may enable Clear all browser instances in the Start Browser step so that whenever the workflow is initialised it clears all browser instances.
20. Make sure you check “Start Node” check box. As we had seen earlier, a workflow works with rows and for steps in a workflow to do any meaningful work, one or more rows have to be ingested into the workflow. By checking “Start Node” checkbox, we tell this step to generate a row. Row which “Start Browser” step generates does not have any columns or fields. Though it is an empty row, it’s still a row and that helps kick off execution of subsequent steps. We will see an example later where “Start Browser” is not supposed to generate a row and so, we will uncheck “Start Node” checkbox. Click **OK** and save the workflow. Enabling check box Notify on browser version change will send an email to Tenant Administrator and Workflow Administrator once on browser version change.



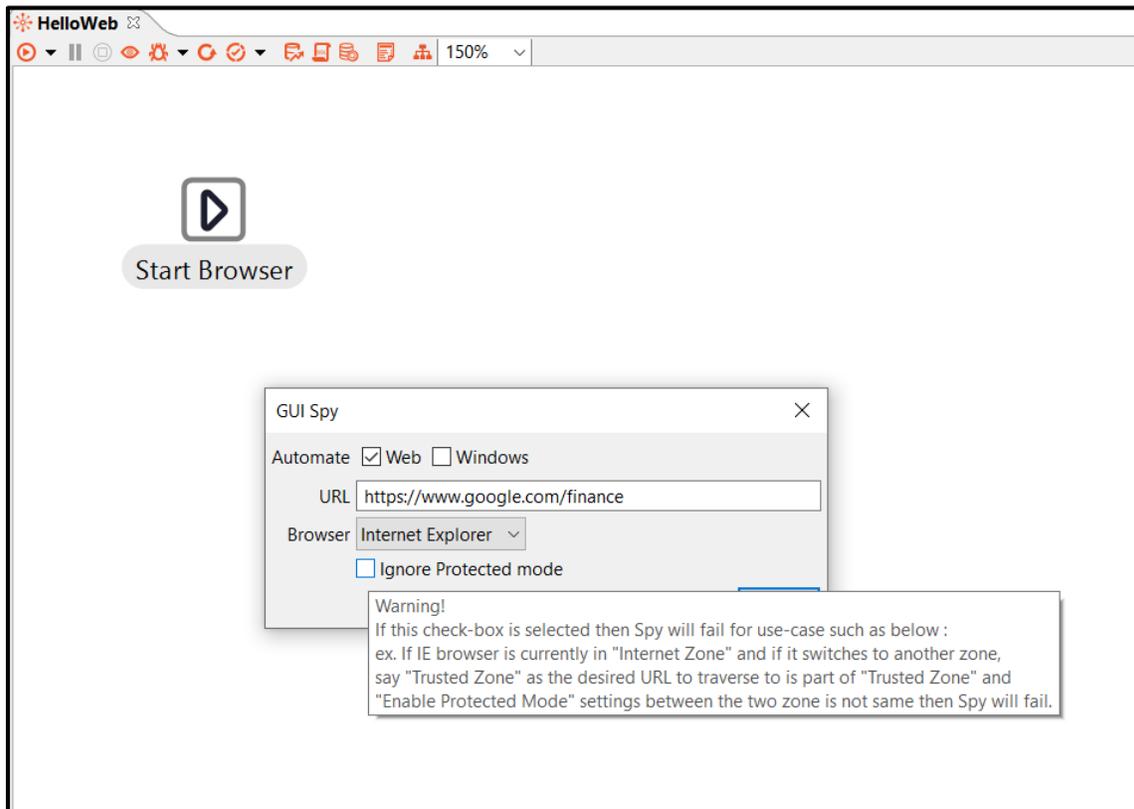
21. We now want to provide stock symbol in the search box to get its price. We will now start GUI Spy to identify an element locator for the search box. In Process Studio navigate to Tools→Spy→GUI Spy menu and click the GUI Spy menu. Alternatively, you may use the shortcut keys Ctrl+Alt+w.



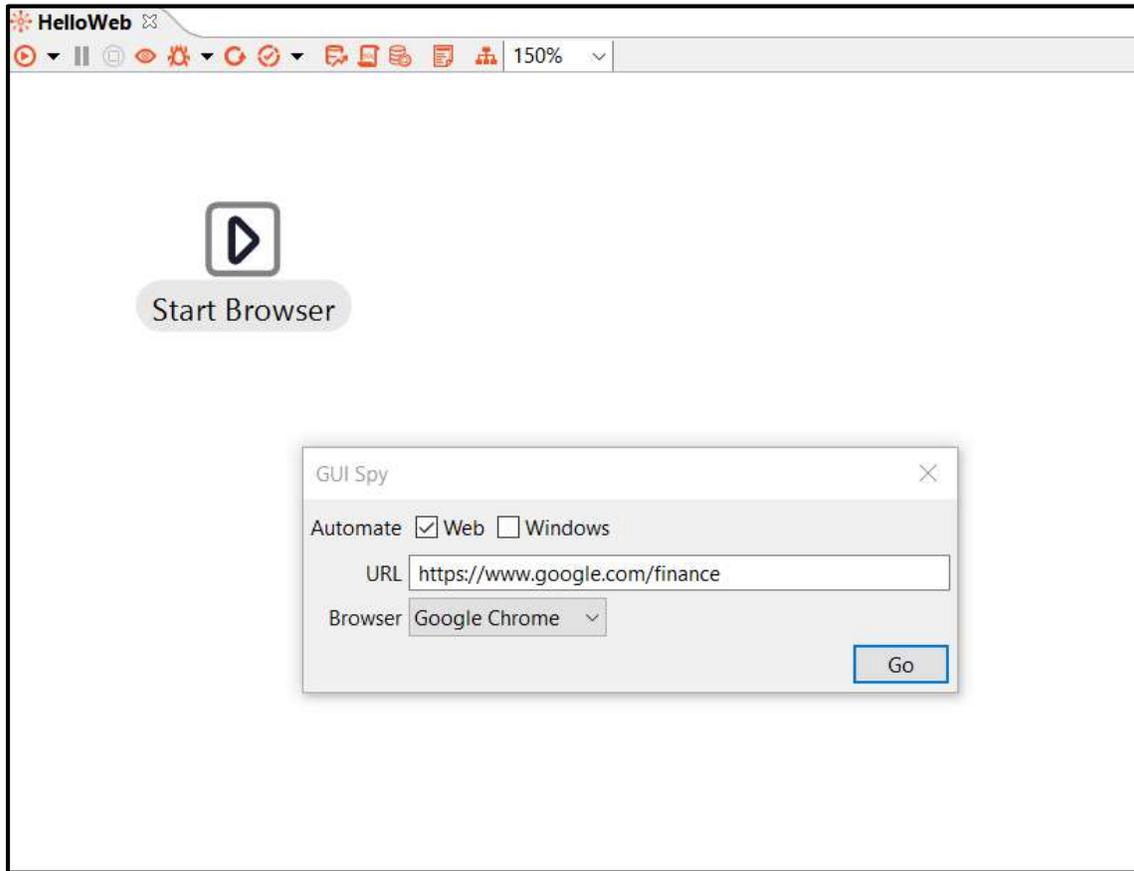
22. A GUI Spy window opens.
23. Since this use case is completely web based enable on Web Automate checkbox.
24. Provide the desired URL and select a browser of your choice Choose a browser of your choice Process Studio supports Firefox (version below 47), Chrome (version 56 upwards) and IE (version 11).



25. Only if IE is selected as the browser a 'Ignore IE Protected Mode Settings' checkbox is visible. Starting version 7, IE introduced the notion of 4 zones; namely Intranet, Internet, Trusted Sites and Restricted Sites. The following two scenarios are to be noted in GUI automation for web using IE.
- The default behaviour in GUI automation using IE is to check "Enable Protected Mode" settings for all the zones and throw an error when this setting is different for any two zones. 'Ignore IE Protected Mode Settings' checkbox in Web-Spy can handle the scenario. When "Ignore Protected Mode" setting is CHECKED in Web-Spy, "Protected Mode" setting check for different zones is ignored and it starts without error no matter what "Protected Mode" settings for different zones are.
- However, note the Warning! that GUI automation for web workflow will fail when it traverses to another zone with a different 'Enable Protected Mode' setting.



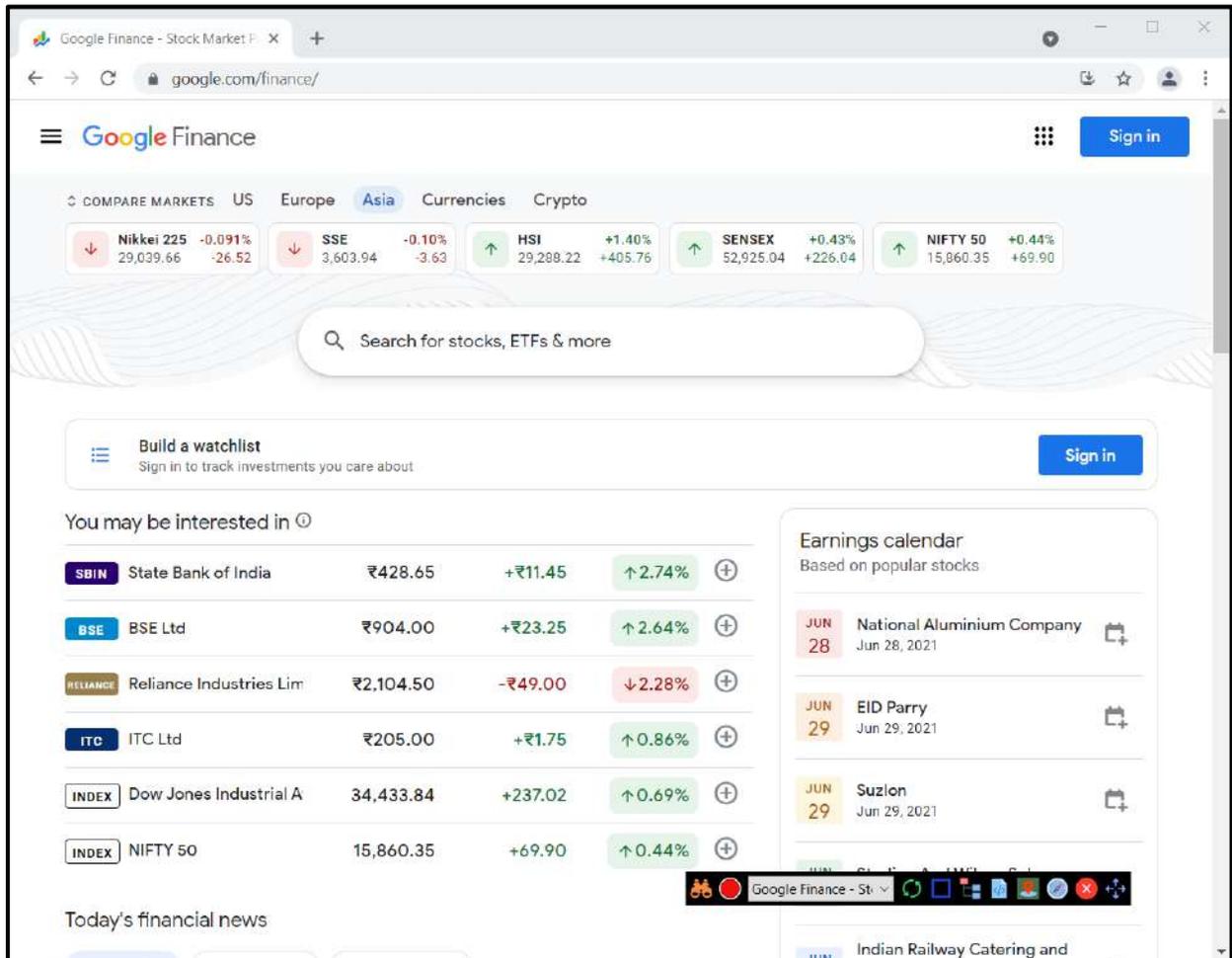
26. Provide the URL <https://www.google.com/finance>, and select browser Google Chrome. Click Go.



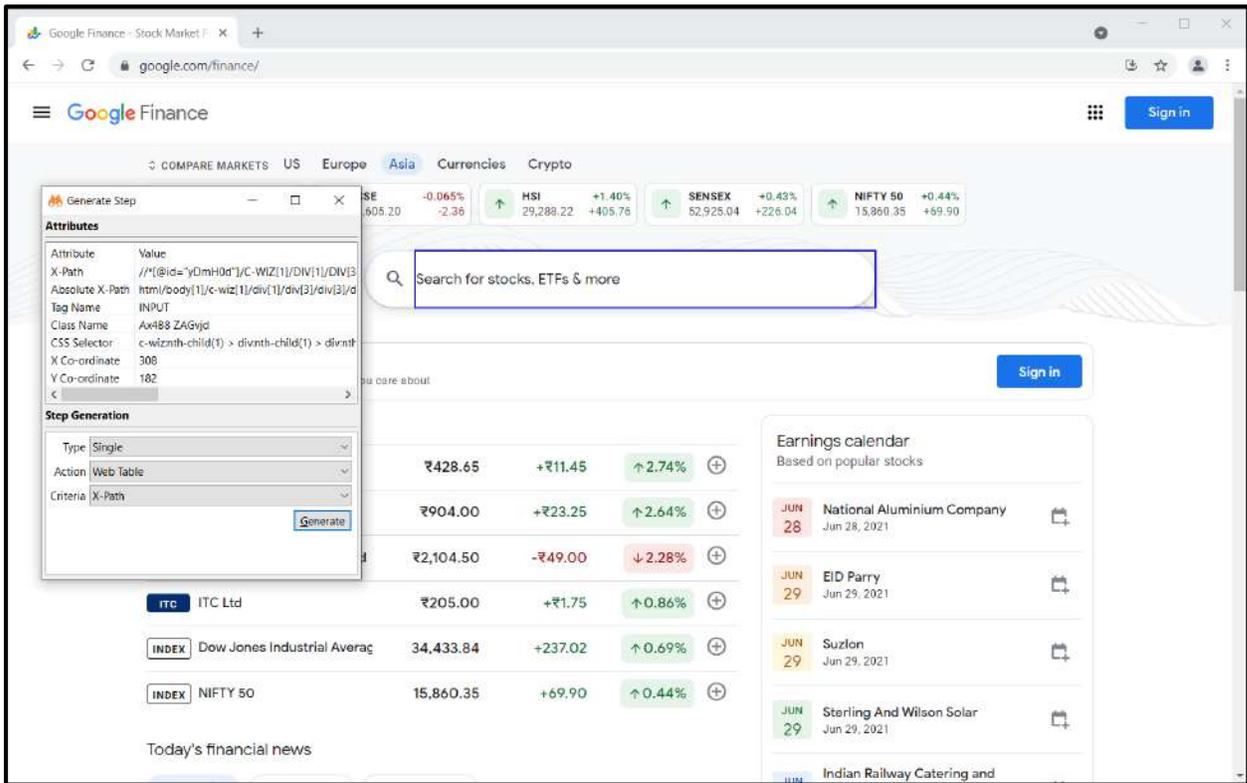
2.2.4 Continue with GUI Spy

Irrespective of whether you have directly started using GUI Spy, or you first added a Start Browser step continue with the following steps,

27. The URL specified in GUI Spy opens in a new browser window. GUI Spy toolbar opens in the foreground of the browser window.



28. Hover over the Search Box where you want to specify the Stock Symbol. You can see a blue box in the search box. Now click the back tick key (`).

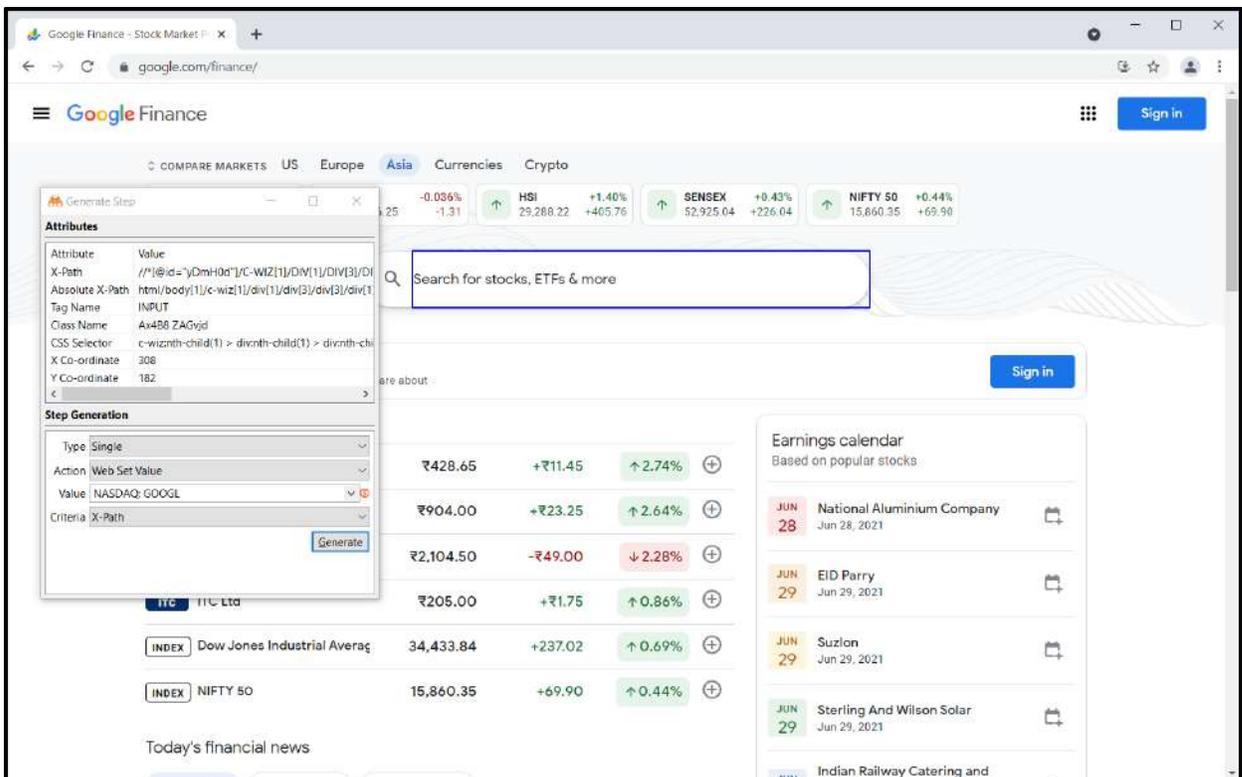


29. The following table describes the Generate Step Configurations. Generate Step Configurations can be used to automatically generate a step in the workflow.

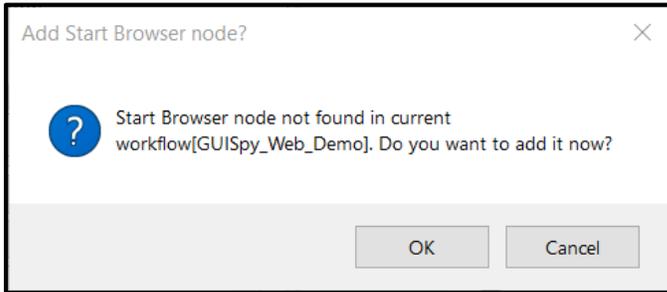
Attribute	Values
Type	<ul style="list-style-type: none"> Single Composite
Action	<ul style="list-style-type: none"> Web Click Web Drop Down Selection Web Element Condition Web Get Value Web Loop Table Web Page Scroll Web Set Value Web Switch Frame Web Switch Window Web Table Web Wait Until

Attribute	Values
	<ul style="list-style-type: none"> Web Actions
Criteria	<ul style="list-style-type: none"> Id Class Name Name Tag Name X-Path CSS Selector X Co-ordinate Y Co-ordinate

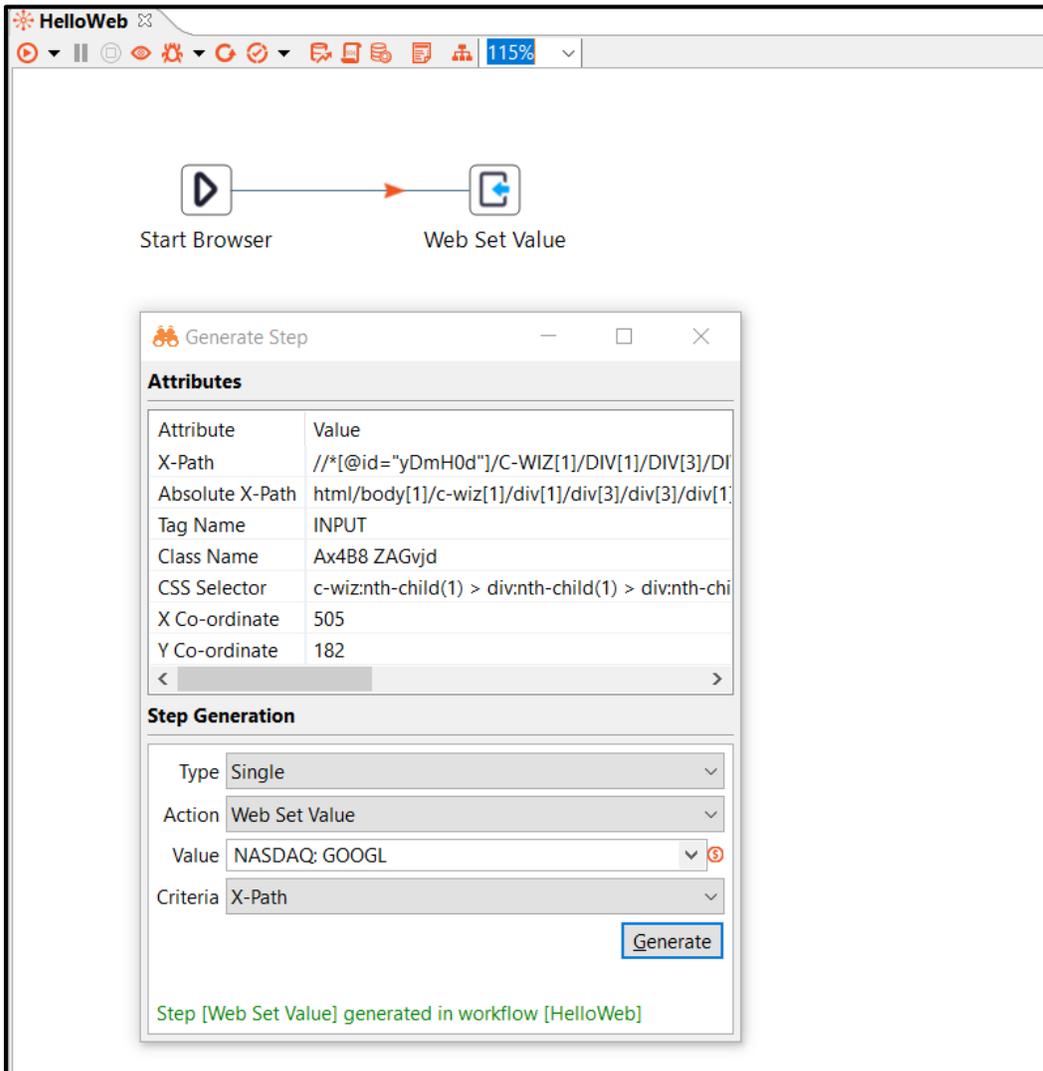
30. The GUI Spy window is now populated with the Search Box element attributes, and the options to Generate a step as seen below. As the text box to enter stock symbol has an **Id** attribute which is unique, we decide to locate the Search Box element using **Id**. We wish to create a single step with Action Web Set Value. We also provide a value to set, NASDAQ: GOOGL. Click Generate.



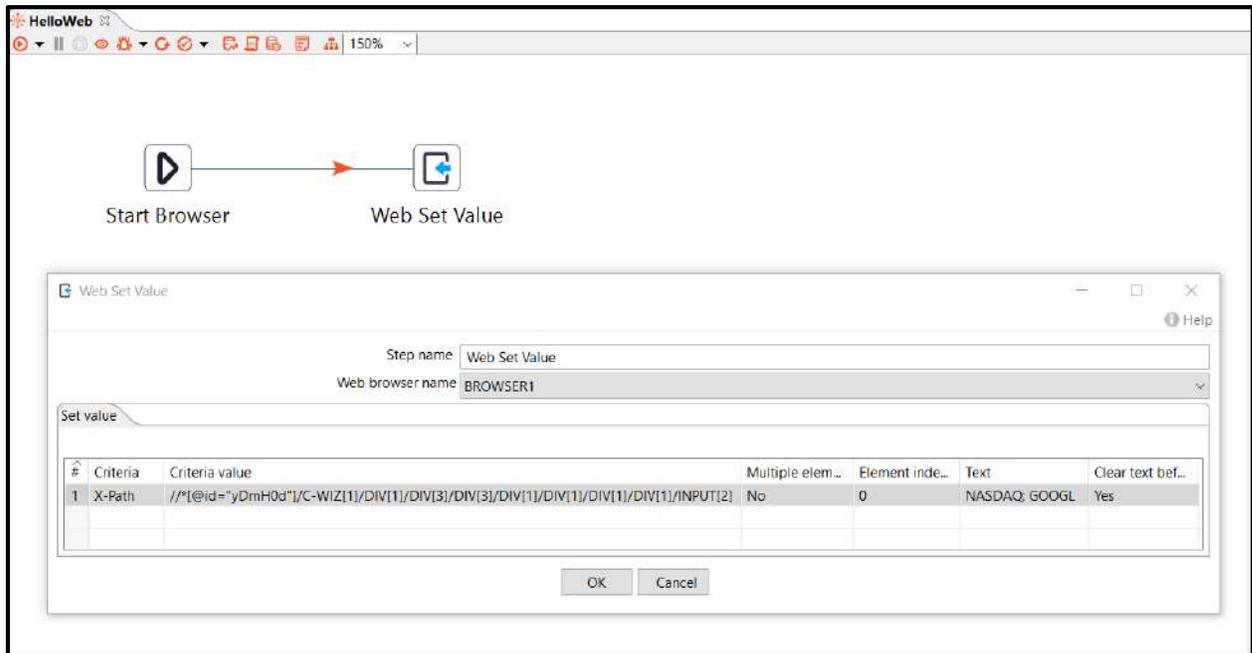
- If you used the first approach to to directly start with GUI Spy without creating a Start Browser step; when you click Generate an Add Start Browser node? Pop up appears, since the Start Browser step has not yet been added. Click OK to create Start browser step in the workflow.



- A message in green text appears at the bottom of the Generate Step dialog window –Step[Web Set Value] generated in workflow [HelloWeb].



33. In Process Studio, you can see that a new step Web Set Value is created with configurations as seen below.



34. Now go back to the GUI Spy browser. After setting stock symbol we wish to press Enter. The Web Click step has three options –
- Normal click,
 - Press enter and
 - Click using java script.
35. Once gain hover over the Search for Stocks field. It should be highlighted by a blue box. If it is not highlighted, click the Spy button on the GUI Spy window. Now hover again, it should be highlighted in blue. Click the back tick(`) button. The Generate Step window opens. Configure the Generate Step to generate a single step for **Web Click**.
36. Click Generate button.

The screenshot shows the Google Finance website with a GUI Spy window and a Generate Step window overlaid. The GUI Spy window is positioned over the search bar, and the Generate Step window is open, showing the configuration for a Web Click step.

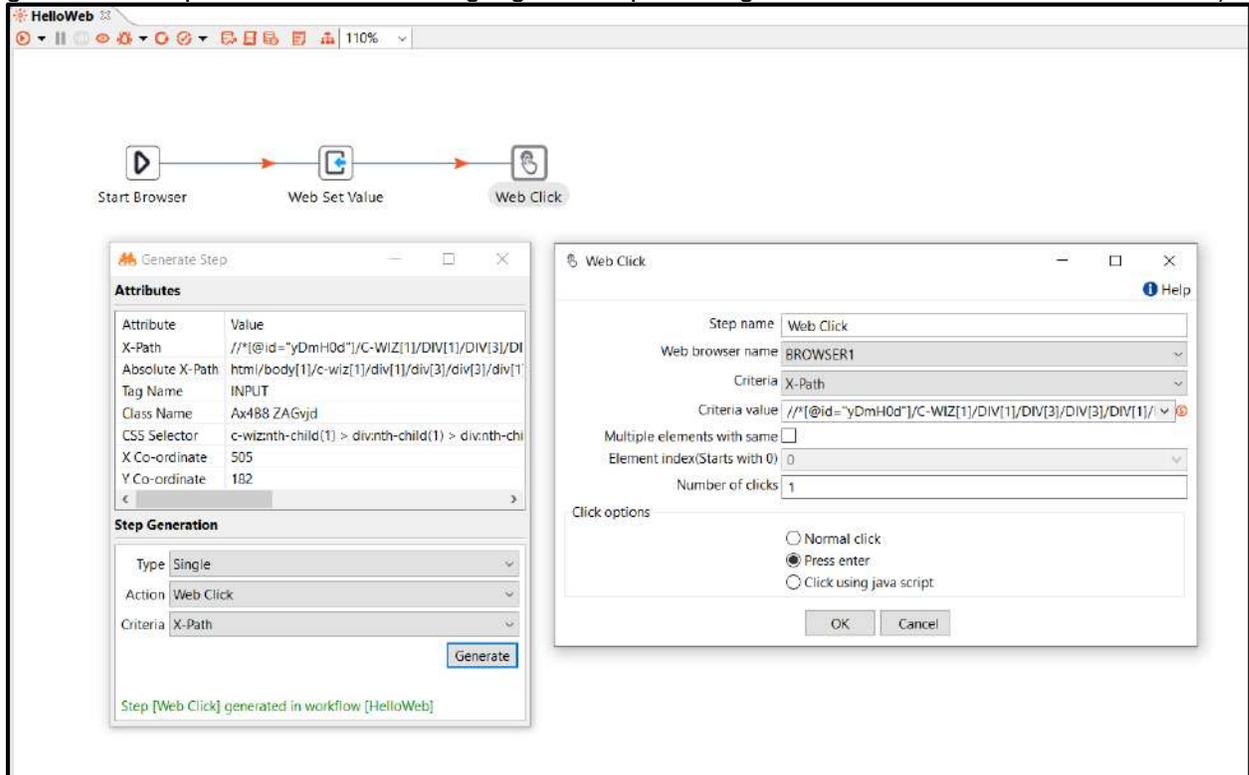
Generate Step Configuration:

- Attributes:**
 - Attribute: Value
 - X-Path: //*[id="yDmH0d"]/C-WIZ[1]/DIV[1]/DIV[3]/DIV[1]
 - Absolute X-Path: Html/body[1]/c-wiz[1]/div[1]/div[3]/div[3]/div[1]
 - Tag Name: INPUT
 - Class Name: Aw4BB ZAGjd
 - CSS Selector: c-wiznth-child(1) > divnth-child(1) > divnth-child(1)
 - X Co-ordinate: 308
 - Y Co-ordinate: 182
- Step Generation:**
 - Type: Single
 - Action: Web Set Value
 - Value: NASDAQ:GOOGL
 - Criteria: X-Path

The background shows the Google Finance website with a search bar and a list of stocks. The search bar is highlighted in blue. The list of stocks includes:

Stock	Price	Change
AMC Entertainment Holdings Inc	\$54.06	↓ 4.66%
Dow Jones Industrial Average	34,433.84	↑ 0.69%
Facebook, Inc. Common Stock	\$341.37	↓ 0.53%
GameStop Corp.	\$209.51	↓ 1.32%
Tesla Inc	\$671.87	↓ 1.17%
INFC Ltd	₹205.00	+₹1.75 ↑ 0.86%
Dow Jones Industrial Average	34,433.84	+237.02 ↑ 0.69%
NIFTY 50	15,860.35	+69.90 ↑ 0.44%

37. A message in green text appears at the bottom of the Generate Step dialog window –Step[Web Click] generated in workflow [HelloWeb].
38. Since Web Set Value was selected the Web Click step is added to Process Studio and connected to the Web Set Value. (Note: If any step is highlighted in Process Studio the new generated step is added after the highlighted step else it get added at the end of the workflow.)

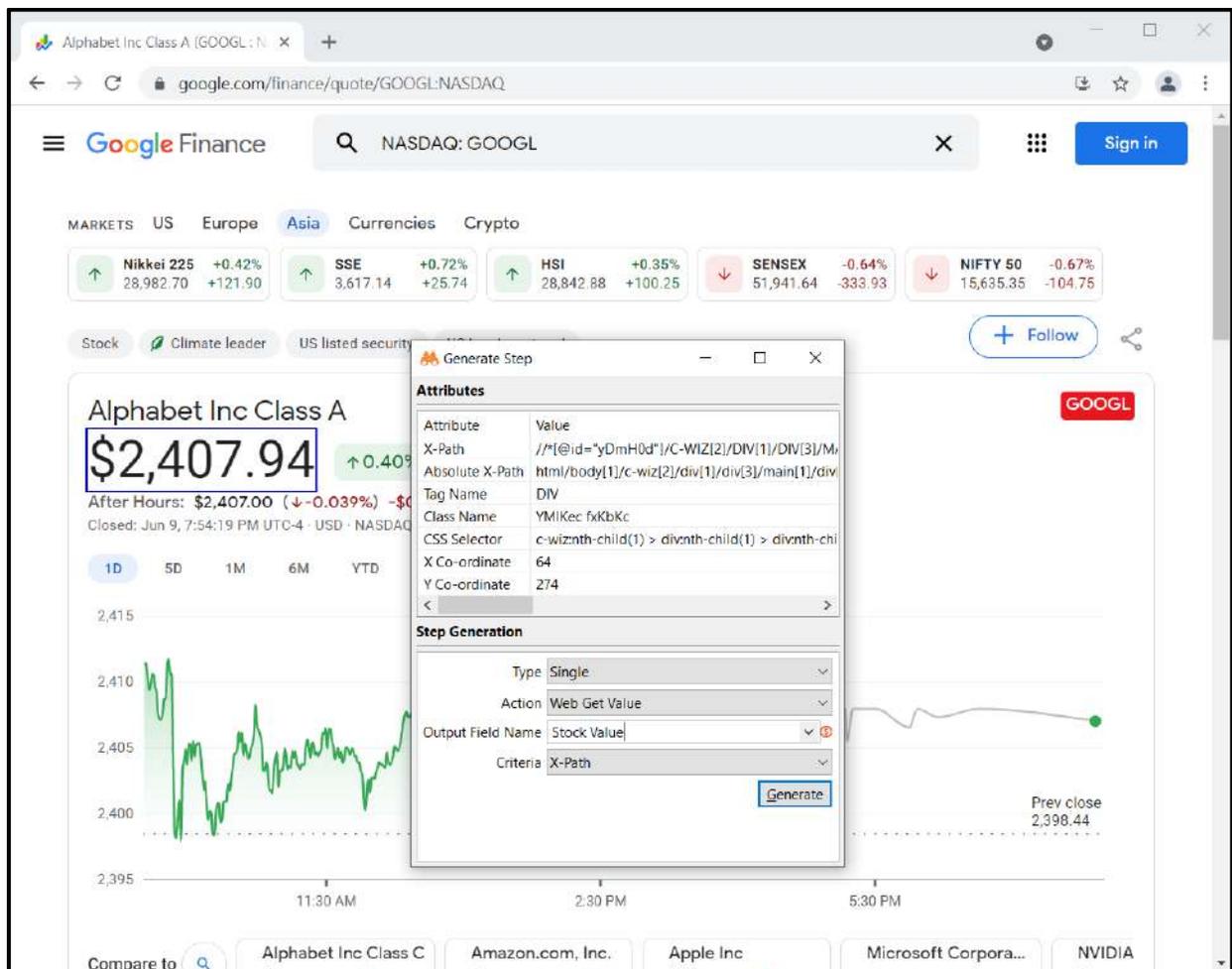


39. Now go back to the GUI Spy browser. Enter a value in the search box. Press Enter.
40. You can see that the Stock Value for NASDAQ: GOOGL is visible as seen below. Hover over the Stock value so that it is highlighted in blue. In case it is not highlighted first click the Spy button and then hover over the Stock Value. Click the back tick (') key on the keyboard.

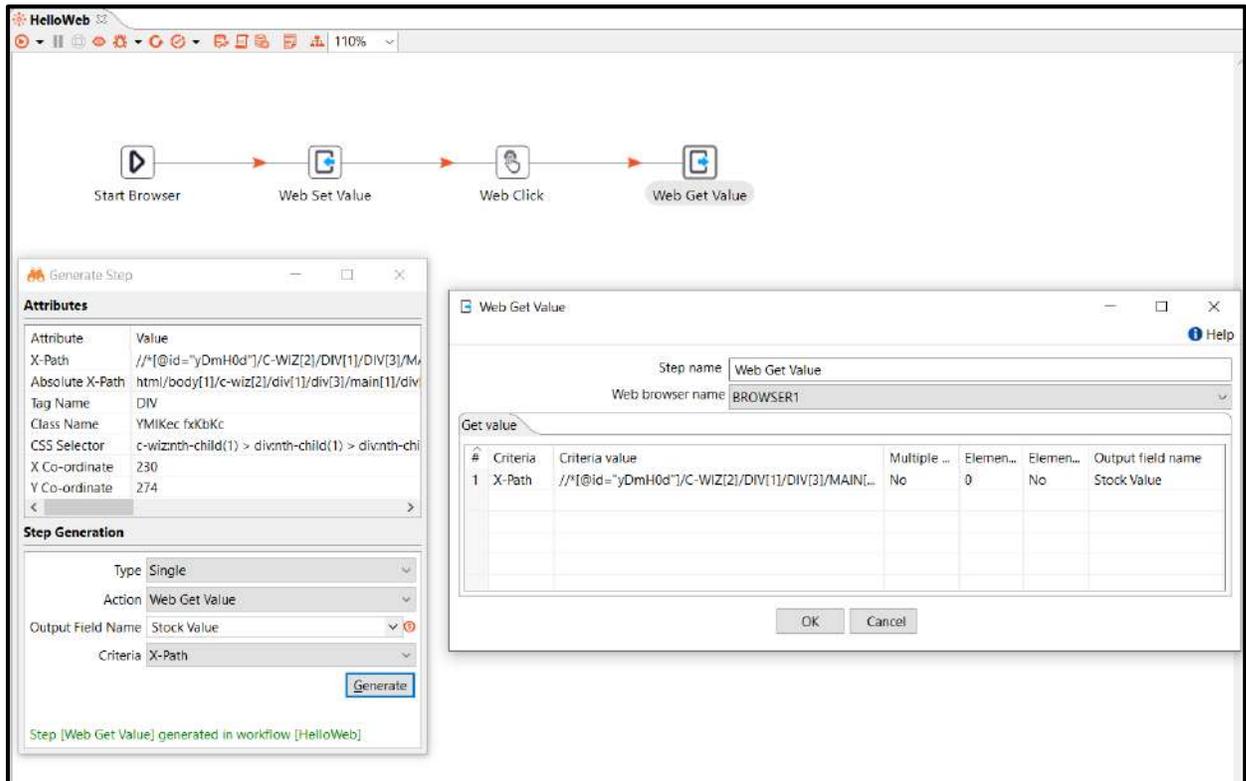
Configure Generate step as shown below for a single step to Get Value. **Web Get Value** step lets you search one or more text fields in a web page.

Stock price is shown in the HTML document inside a **** tag. Unfortunately, **Id** of this tag though unique, changes every time you access the page. Hence we can't use **Id** of this element to locate it. In this case we use relative Xpath with value `//*[@id="knowledge-finance-wholepage__entity-summary"]/div/g-card-section/div/g-card-section/div[1]/span[1]/span/span[1]`. Hence we use element search criteria as X-Path.

Output Field Name lets you specify name of the field or column which would have value of stock price. With this, empty row, which **Start Browser** generated now has a field (or a column) called "Stock PriceThe output value is stored in a field Stock Value. Click Generate.



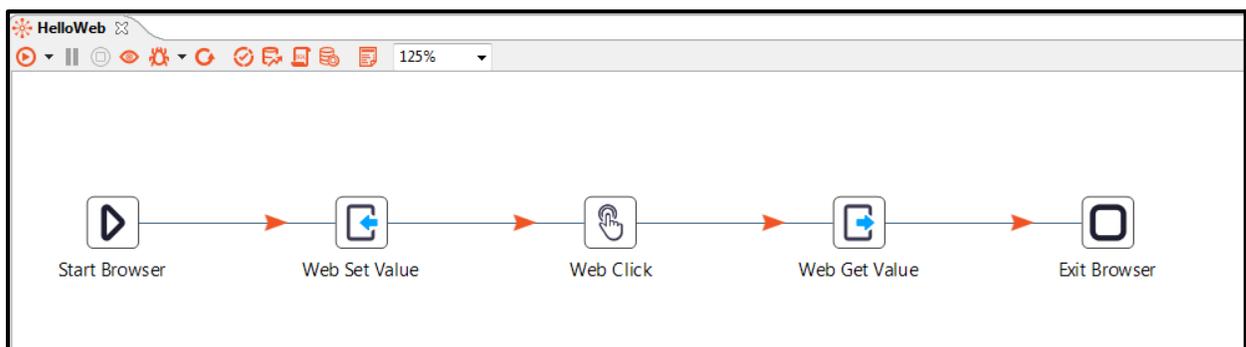
41. Web Get Value Step Generated message is displayed in a callout at the bottom of the web page. Web Get Value step is added to the workflow as seen below.



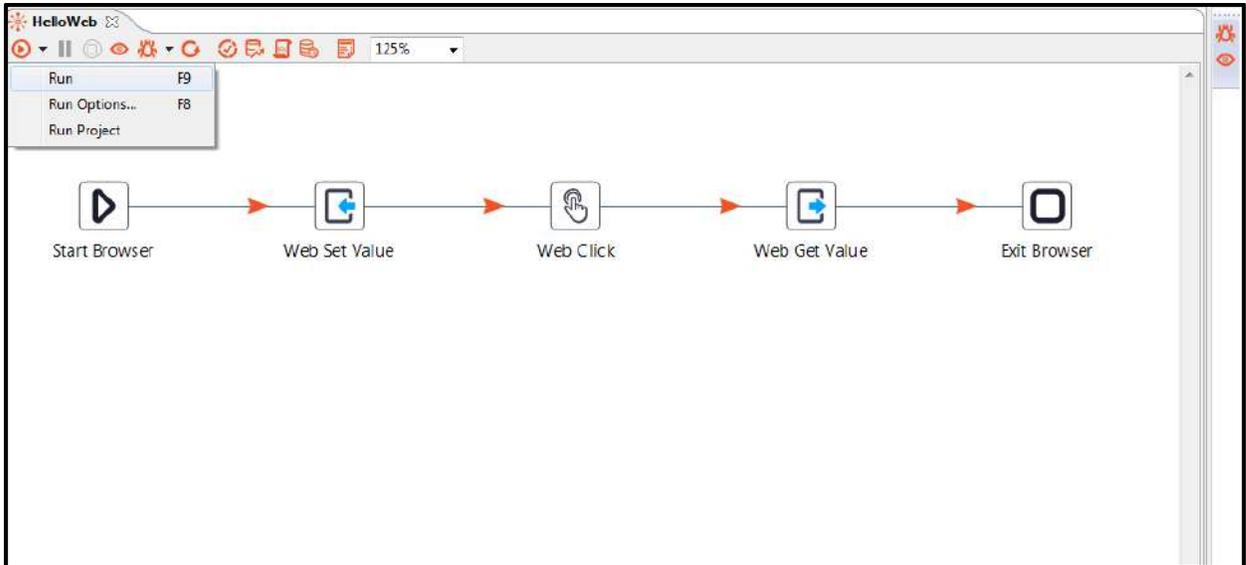
42. We now want to quit the browser. Drag **Exit Browser** node and make a connection between **Web Get Value** step and **Exit Browser** step. Double click **Exit Browser** step to edit its properties and make sure BROWSER1 is selected (as all in previous steps) as Web Browser Name. Your workflow should look like this.

Note: Start browser initializes the web driver. The web driver is closed by Exit Browser. In case you close the browser directly the web driver is still initialized; and if you try to run any new workflow that initialises a web driver it throws an error. You can clear all browser instances by navigating to Tools→Clear All Browser Instances.

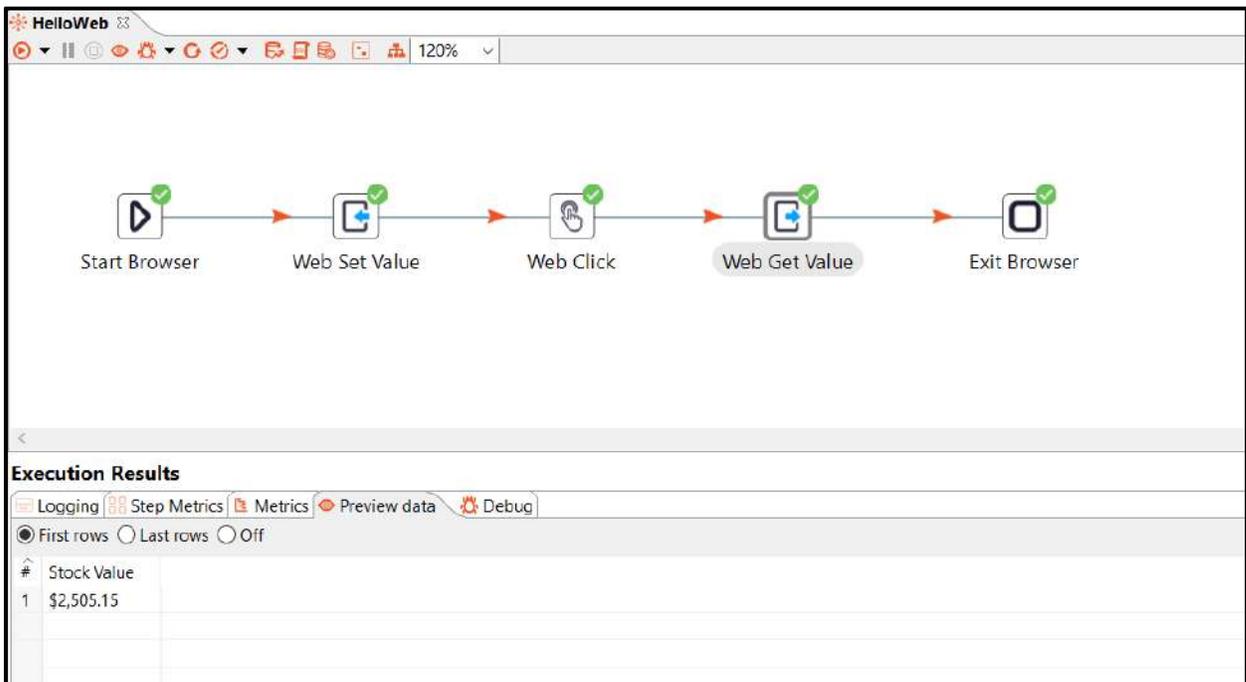
Ideally you should have a practice of having an Exit Browser step as an error handling step for all GUI Automation for web steps.



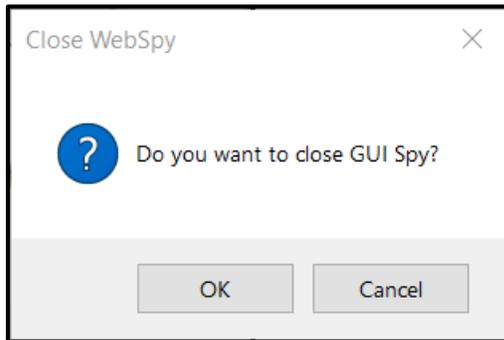
43. Run the workflow by clicking the arrow or the two options in the drop down i.e. Run and Run Options or Run Project
44. To use Run Project make sure you have select a Main workflow/process for the project by editing it.



45. In case the workflow is not saved it automatically saves the workflow and starts execution. In the screenshot below, you can see the successful execution of the workflow. In the **Preview Data** tab below, you see the value of field **Stock Price**. When workflow runs you can see Chrome (or the browser you choose) getting launched and search operation being performed.



46. Click on the Close GUI Spy icon. A pop-up warning message appears. Click OK to close GUI Spy.



47. A sample workflow HelloWeb was created to demonstrate GUI Spy for web in this section.

3 Project 2: Multiple Stock Values

3.1 Making HelloWeb Generic

We will now make HelloWeb generic i.e. we will extend HelloWeb created in Project 1, to read from a file (in this case CSV file) with a list of stock symbols and get current stock price of every stock.

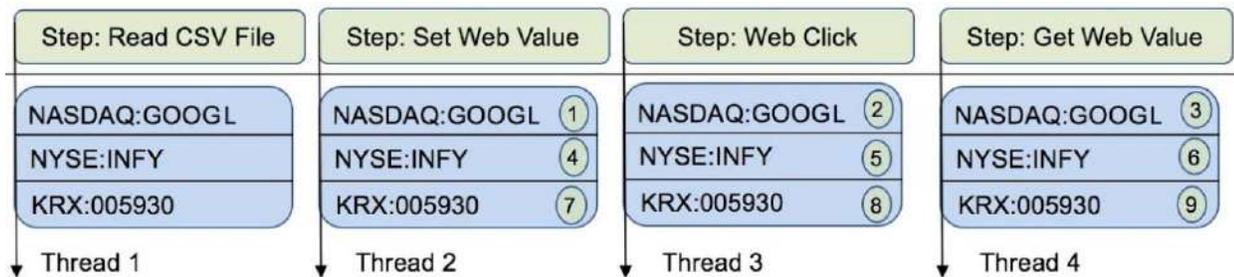
3.1.1 Having a list of stock symbols in CSV file

Create a CSV file `stock_symbols.csv` with a header row with Company Name and Stock Symbol and several row entries with contents as shown below.

<code>stock_symbols.csv</code>
Company Name, Stock Symbol
Alphabet Inc.,NASDAQ: GOOGL
Infosys Ltd., NYSE: INFY
Samsung Electronics Co Ltd, KRX:005930

3.1.2 Processing multiple rows sequentially in multi-threading

In order for the workflow execution to succeed with multiple rows in CSV file (say three in this case), execution flow has to happen as shown in the figure below with numbers from 1 to 9 (enter stock symbol in search box, click search button and get the value of the price from search result, repeat this for all the stock symbols). You can think of each step as being a worker thread with queues for input and output rows. Each thread processes its rows as they become available. Each step executes in its own thread and we can't guarantee execution order as shown in the figure.



Even if execution were to happen without errors in this case, results might be incorrect. What we essentially want is: Steps **Set Web Value**, **Web Click** and **Get Web Value** to execute in that order for every row in the CSV file. In a way, we want to combine these steps into one step and we can achieve exactly that by,

1. putting steps in between Start Loop and Continue Loop steps.
2. putting steps in another workflow and calling this workflow from the first.

We will showcase three methods of processing multiple rows sequentially in multi-threading in the following three workflows. Following are the advantages of each of these methods.

- **HelloWebGeneric**
In HelloWebGeneric we will modify HelloWeb to additionally read multiple stock values from a csv file. To achieve rows serialization, we will use Loop. We will put steps between a Start Loop and Continue Loop step. Loop will help us pass a single row at a time from the CSV file to the Loop steps and achieve serialization.
- **HelloHierarchy**
In HelloHierarchy workflow we will modify HelloWebGeneric. In HelloHierarchy we will replace Loop. We will achieve serialization of rows by putting the four steps in a child Workflow Executor step. It is a good idea to use parent-child workflows when you want to modularize your workflow.

3.2 Loop in Workflows: Components and Features

In the HelloWeb we did GUI automation (for Web), with static values. In HelloWebGeneric we will achieve GUI automation for multiple input rows. Steps in a workflow execute in parallel. We will achieve serialization using Workflow Loop steps.

3.2.1 Loop Components

Workflow Loop consists of two steps:

1. Loop Start

Loop Start marks beginning of loop execution. Loop Start step features as below are discussed in the following section.

- Single Threaded
- Additional Fields
- Number of Iterations or Dynamic Conditions
- True/False Output Hops

2. Loop Continue

Continues to Loop Start step for further decision making.

Additional Loop steps include the following,

3. Web Loop Table

While, Loop and Continue Loop steps are always in pairs, WebLoopTable step is used to loop records of a table.

4. Break Loop

Break Loop step is used to break a loop based on conditions.

3.2.2 Loop Start Features

The following are Loop Start step features,

1. Single Threaded: The steps in the loop will be executed as a single thread. The other input rows are paused until a continue signal is received from loop continue step.
2. Input Row: Additional fields can be added to this input row, which would be populated during the loop iteration and available after the loop iteration. All other fields generated by steps during loop iteration will be destroyed i.e. their scope would be valid only for that particular iteration.
3. Loop Condition: A dynamic condition block is provided. This can be used to construct simple as well as complex conditions. Based on the output of this condition, the input row will be routed to either continue (in case of condition failure) or end the loop (in case of condition success).
4. Implicit Counter: The user can provide a counter name and number of iterations for which the loop would execute.
5. Output Hops: There will be two output hops, for 'true' and 'false' condition. It is mandatory to have Loop Continue at the end of true path.

3.2.3 Loop Continue Features

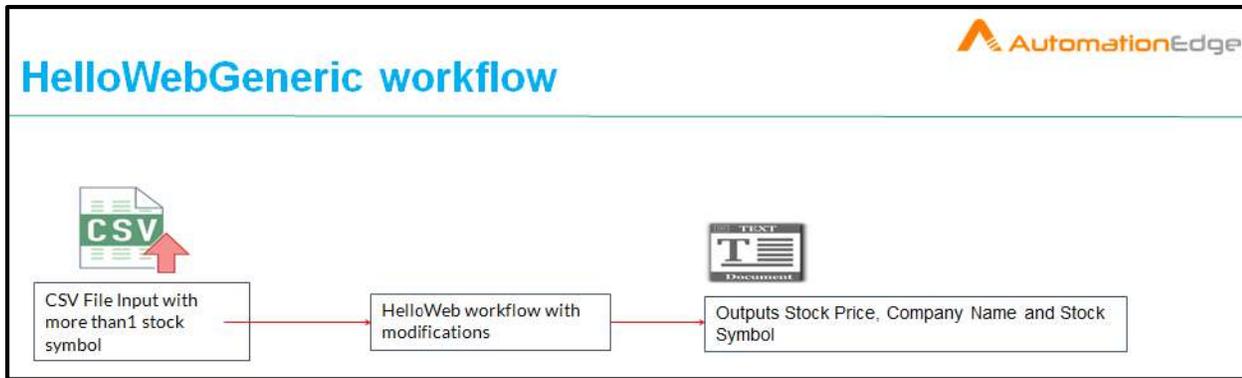
All the additional generated fields will be passed. Also the counter is incremented before the row is forwarded.

3.3 Building your second GUI Automation for web workflow: HelloWebGeneric

In HelloWebGeneric workflow we will fetch multiple stock values by reading stock symbols from a csv file. We will modify HelloWeb workflow with introduction of new steps –

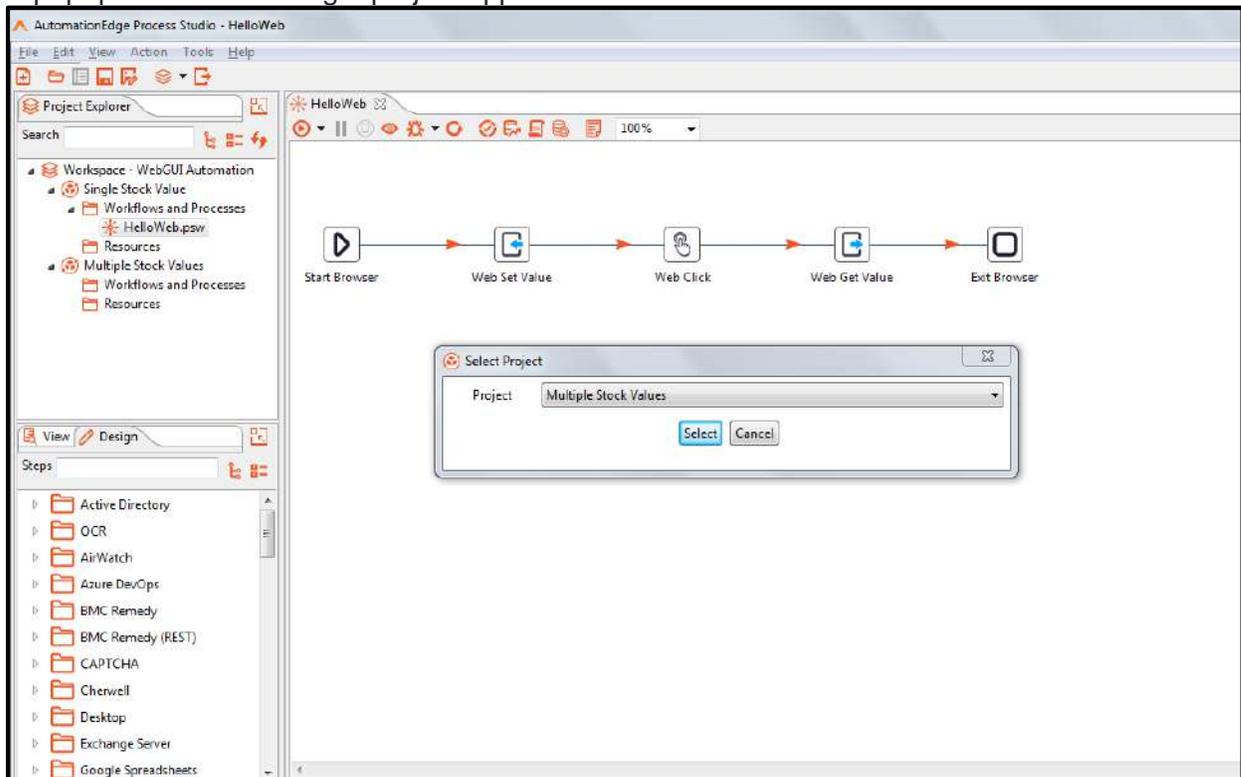
- Web URL Navigation to navigate to the <https://www.google.com/finance> for each stock symbol.
- Start Loop and Continue Loop steps to achieve rows serialization.
- Text file output file to capture the csv file input rows and the captured stock symbol.

HelloWebGeneric can be summarized in the screenshot below.

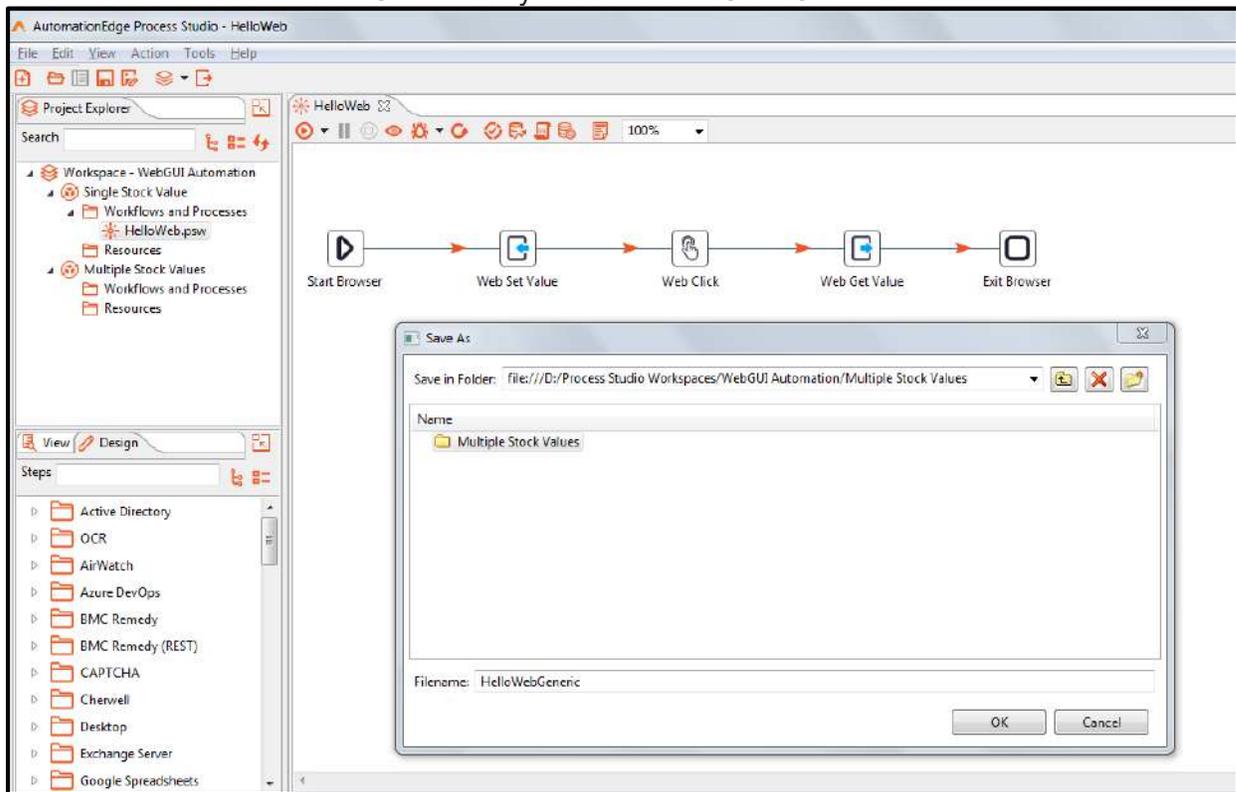


We will create our second project in the same workspace – WebGUI automation. Following are the steps to create the workflow: HelloWebGeneric

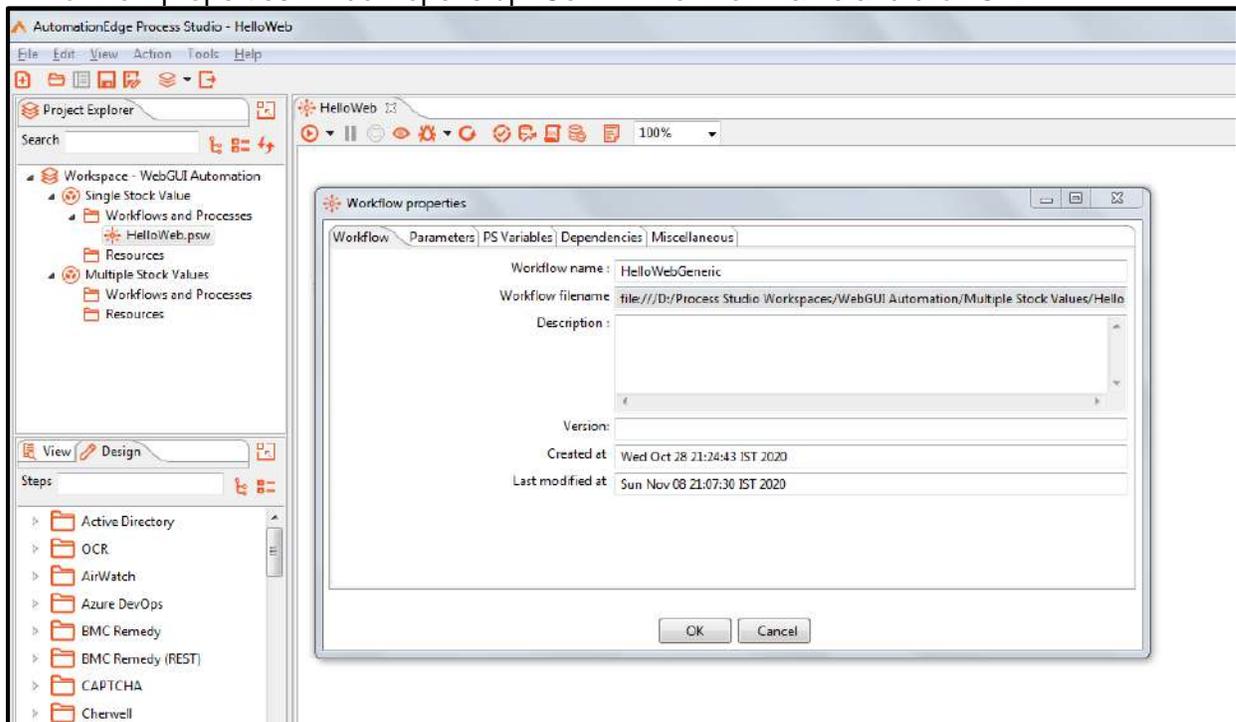
1. Right click on WebGUI Automation workspace to create a new project. Name the project Multiple Stock Values. Multiple Stock Values is visible below.
2. To make a copy of HelloWeb workflow in the first project - Single Stock Value and save it in the project - Multiple Stock Values:
3. Open HelloWeb workflow from the first project as seen below. From the File menu click Save As. A popup to select the target project appears.



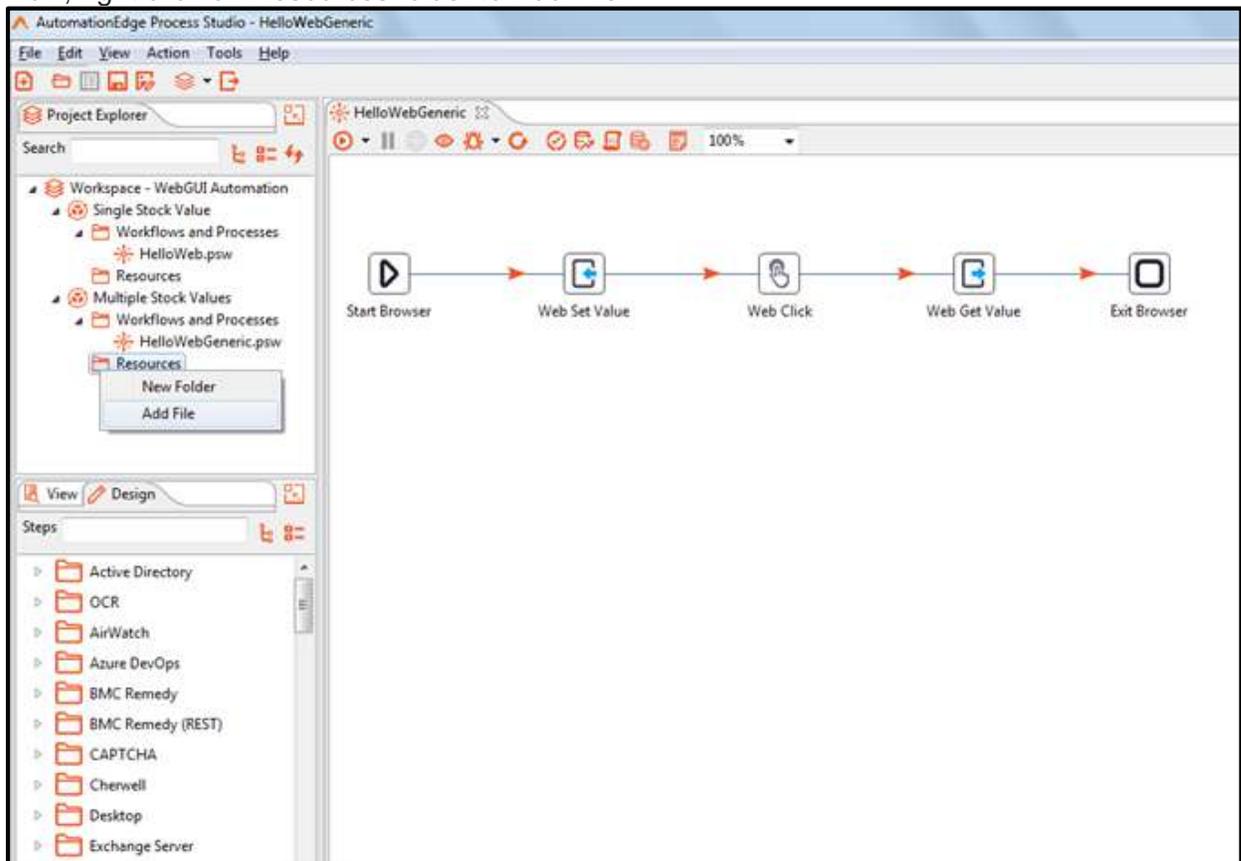
- Provide the name 'HelloWebGeneric' to your Workflow. Click OK.



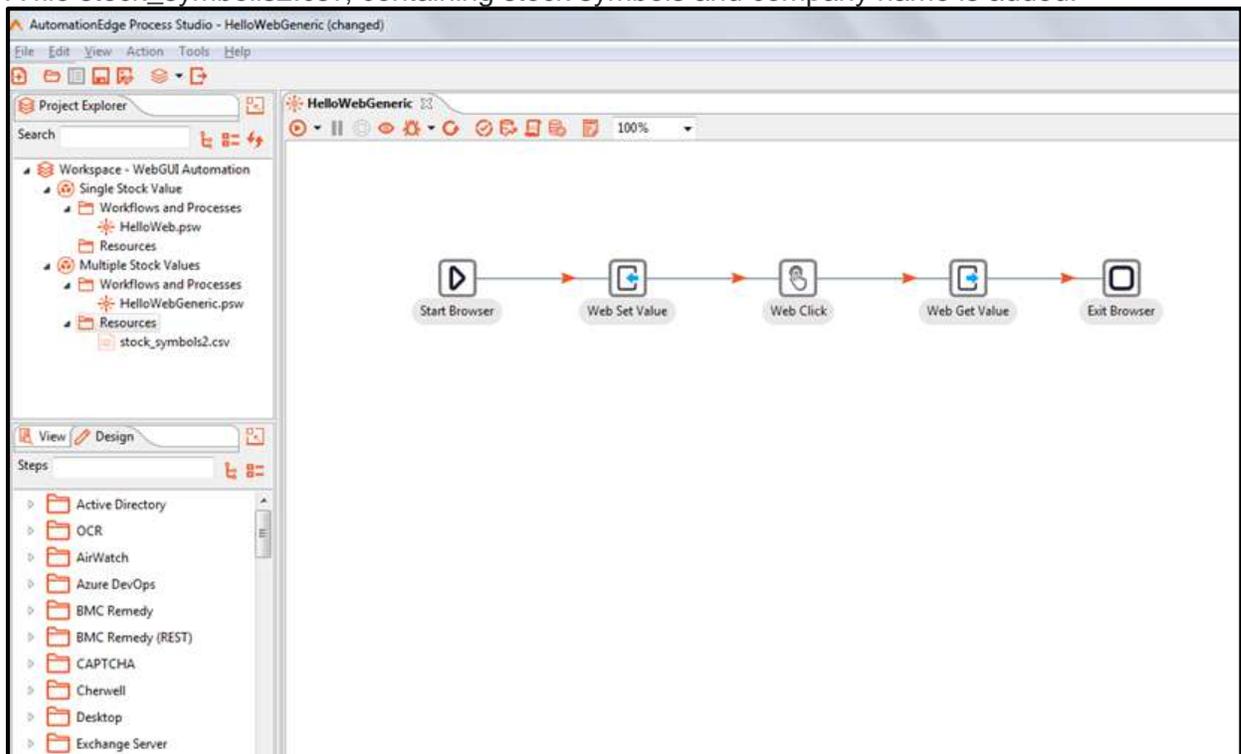
- A Workflow properties window opens up. Confirm workflow name and click OK.



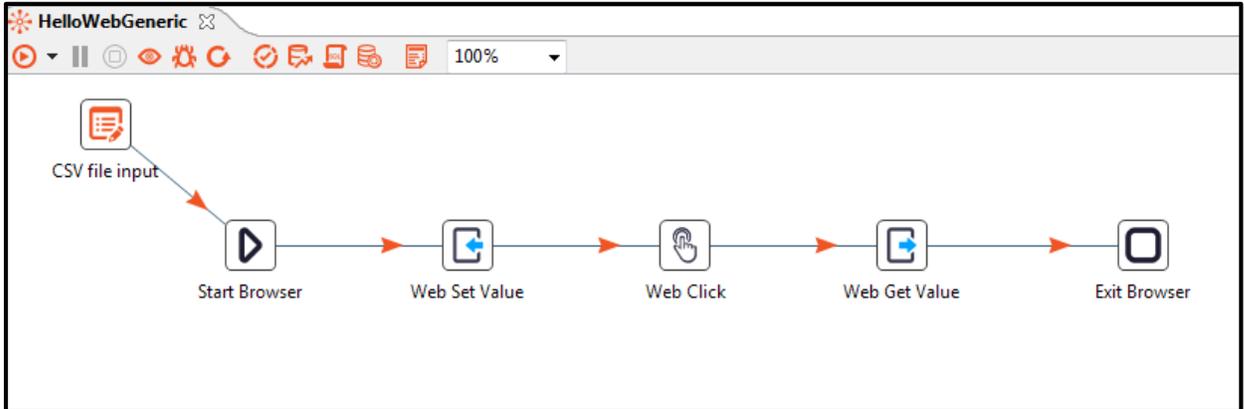
- Now, right click on Resources folder to Add File.



- A file stock_symbols2.csv, containing stock symbols and company name is added.

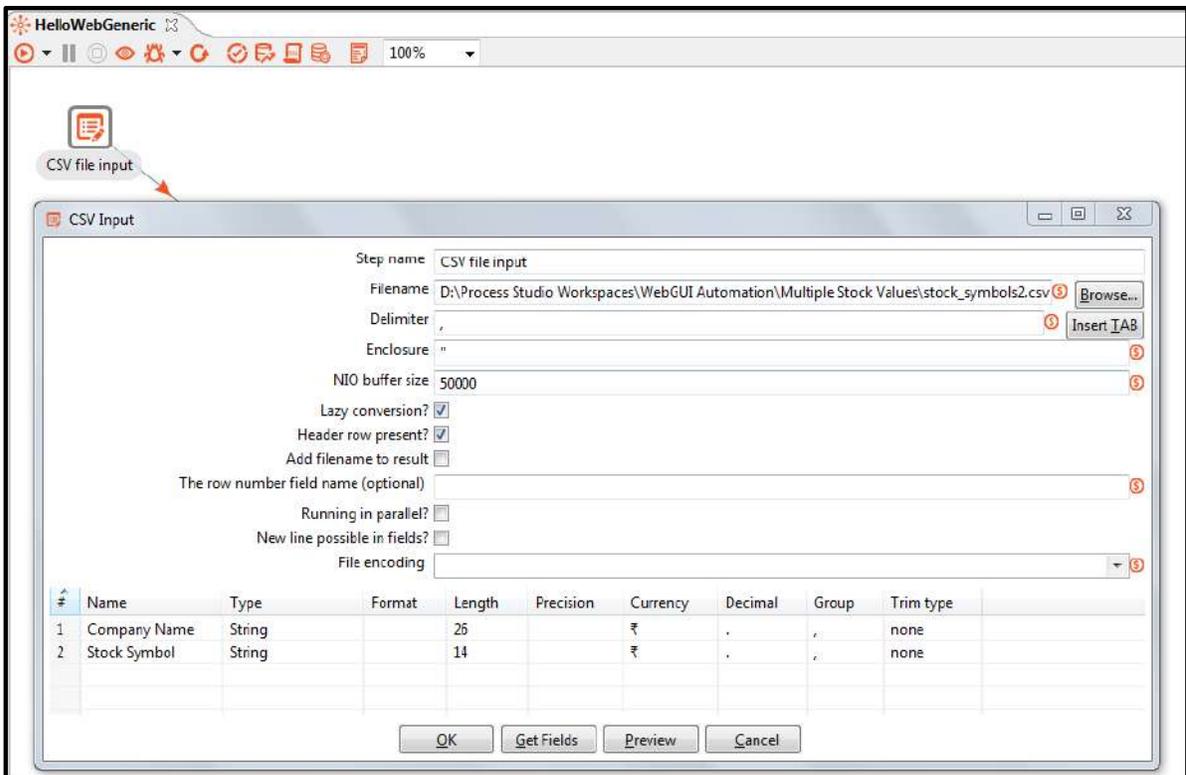


8. HelloWebGeneric workflow is now open in Process Studio as shown below.
9. Add **CSV file input** step to the workflow and connect it to **Start Browser** step. Your workflow should look like this.

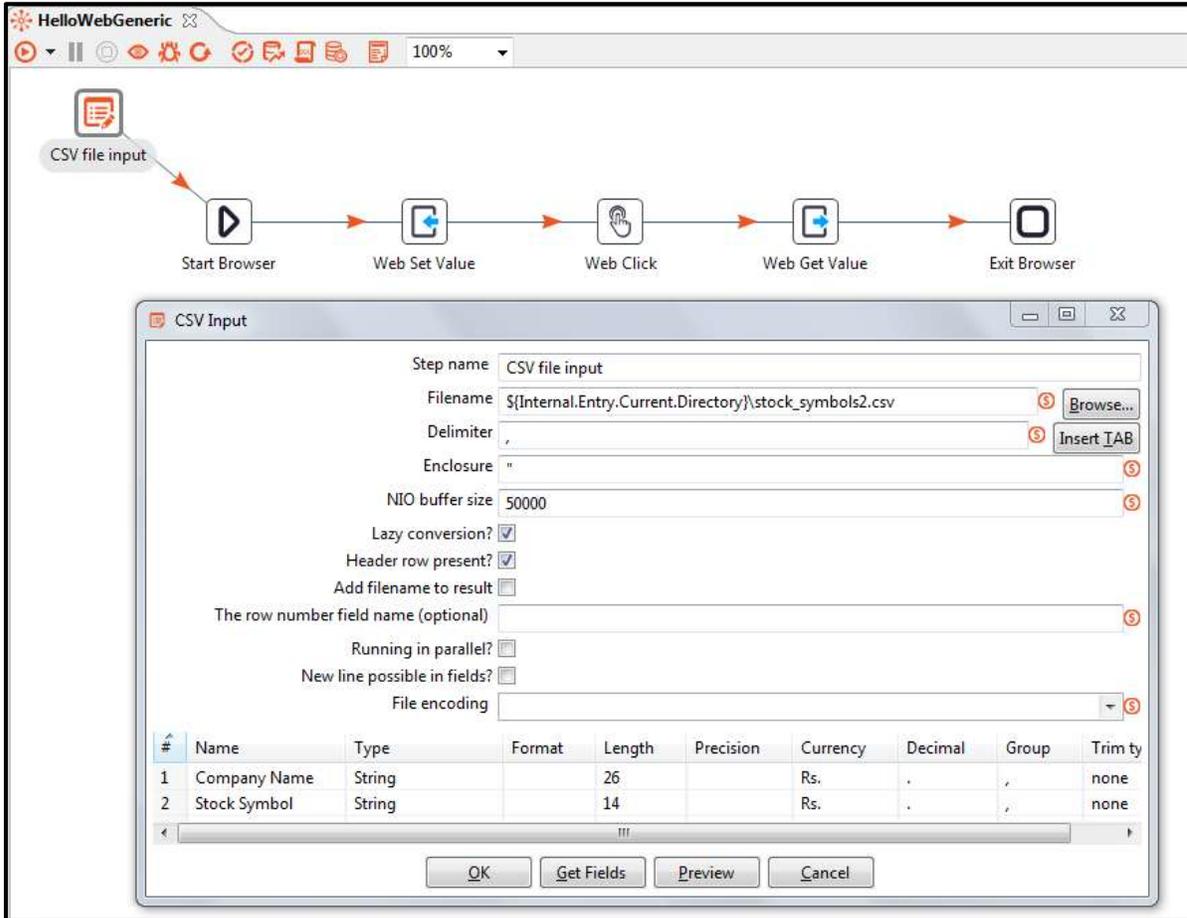


10. Double click on **CSV file input** step to configure its properties. Browse your local file system and choose stock_symbols.csv file. Click **Get Fields** and click **OK** and **Close** on two ensuing dialogs. This should result in two fields **Company Name** and **Stock Symbol** listed as shown below. Click **OK** and save the workflow.

In this workflow CSV file input acts as an INPUT step which is going to ingest data into the workflow. Please make sure, stock_symbols.csv file has a header row and can have multiple rows corresponding to a stock symbol.



- Once you have fetched the CSV header fields you may parameterise the filename. Replace the file of the path with `${Internal.Entry.Current.Directory}`. `${Internal.Entry.Current.Directory}` is an internal variable that refers to the same file path where the current workflow or process is stored. This makes the workflow and file portable together to any other location and there will be no need to change the Filename as it is parameterised to point to the workflow directory.



The screenshot displays the Process Studio interface for a workflow named "HelloWebGeneric". The workflow consists of five steps: "Start Browser", "Web Set Value", "Web Click", "Web Get Value", and "Exit Browser". A "CSV file input" step is positioned at the beginning of the workflow.

The "CSV Input" dialog box is open, showing the following configuration:

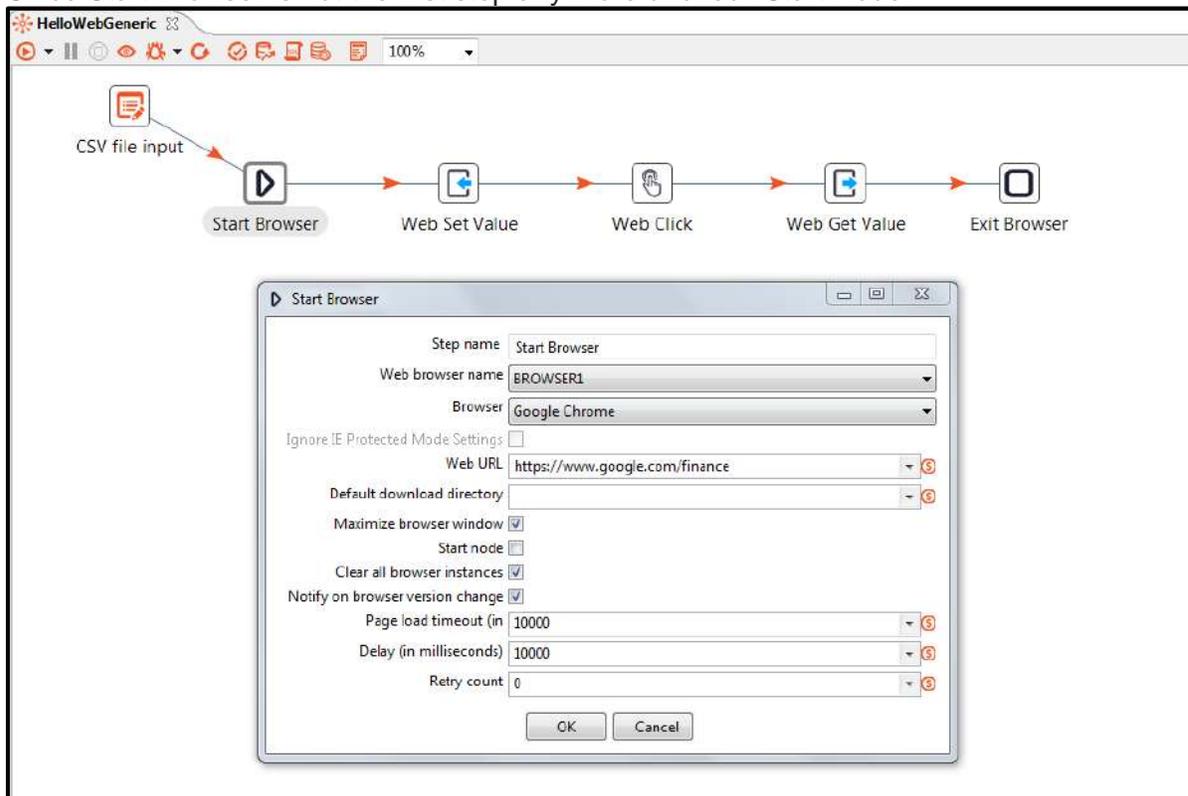
- Step name: CSV file input
- Filename: `${Internal.Entry.Current.Directory}\stock_symbols2.csv` (with a "Browse..." button)
- Delimiter: , (with an "Insert TAB" button)
- Enclosure: "
- NIO buffer size: 50000
- Lazy conversion?
- Header row present?
- Add filename to result?
- The row number field name (optional):
- Running in parallel?
- New line possible in fields?
- File encoding:

The dialog also includes a table with the following data:

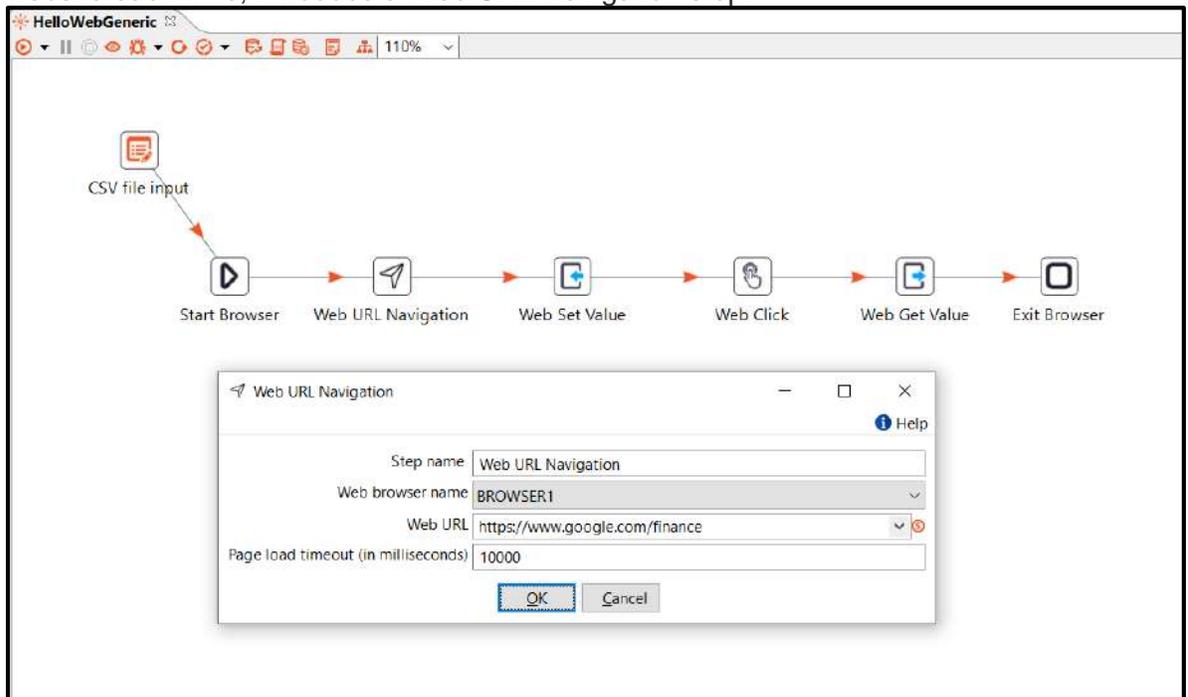
#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim ty
1	Company Name	String		26		Rs.	.	,	none
2	Stock Symbol	String		14		Rs.	.	,	none

Buttons at the bottom of the dialog include "OK", "Get Fields", "Preview", and "Cancel".

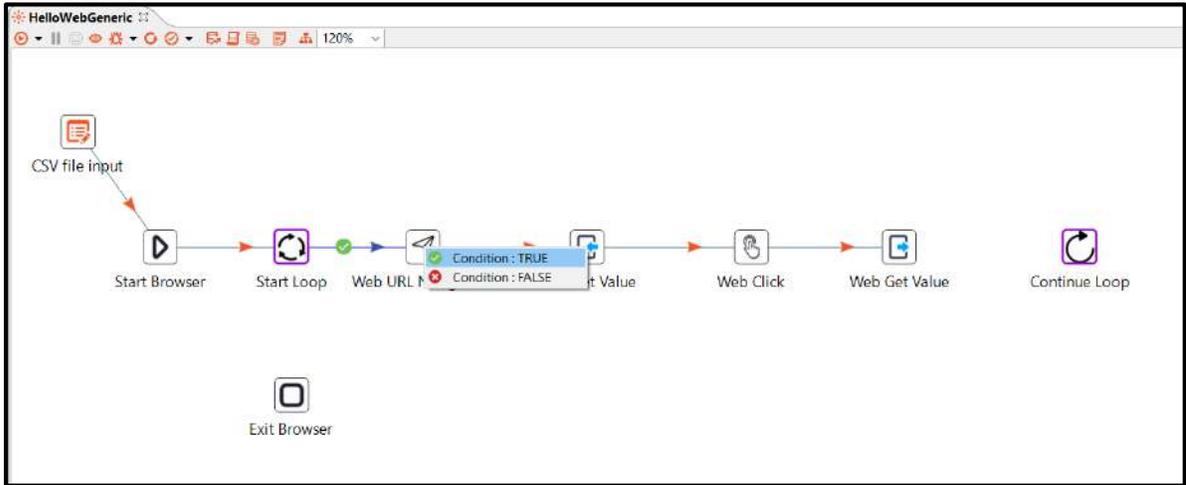
12. Since Start Browser is not the first step any more uncheck Start Node.



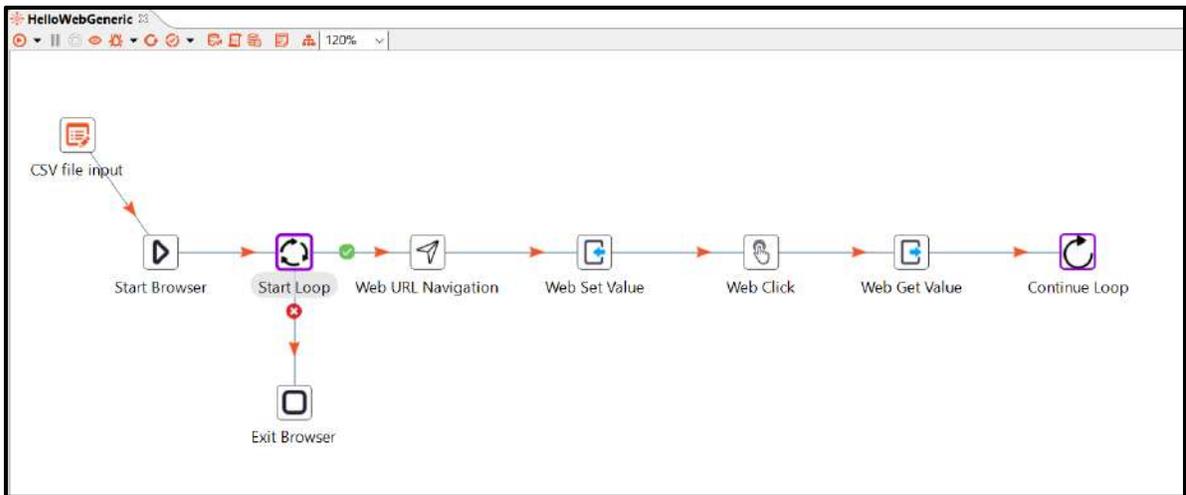
13. Since there are multiple rows it is required to navigate to the <https://www.google.com/finance> website each time, introduce a Web URL Navigation step.



14. In Process Studio the steps start in parallel. We need to ensure that rows of stock symbols pass one by one for stock value retrieval. Hence, we shall introduce the Loop steps.
15. Connect Start Loop to Web URL Navigation step with Condition: TRUE hop.



16. Place Continue Loop at the end of the workflow as seen below.
17. Connect Start Loop to Exit Browser step with the Condition: FALSE hop.



18. Double Click Start Loop and configure it to perform 1 iteration of input rows from the CSV file. Provide an Iterator Name itr1.

The screenshot shows the Process Studio interface with a workflow and a configuration dialog for the 'Start Loop' step.

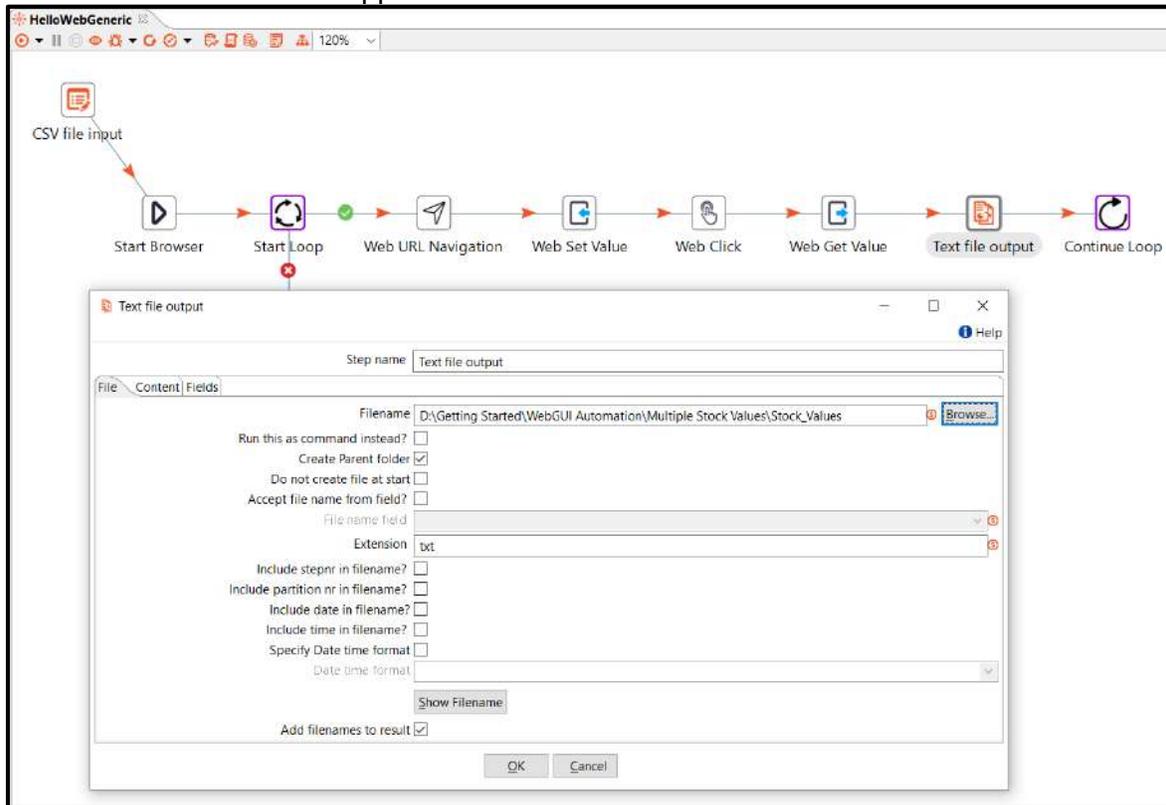
Workflow:

- Starts with a 'CSV file input' icon.
- Followed by 'Start Browser'.
- Then 'Start Loop' (highlighted with a red 'X' and a red arrow pointing to the dialog).
- From 'Start Loop', the flow branches:
 - One path goes down to 'Exit Browser'.
 - Another path goes right to 'Web URL Navigation'.
- The main flow continues from 'Web URL Navigation' to 'Web Set Value', 'Web Click', 'Web Get Value', and finally 'Continue Loop'.

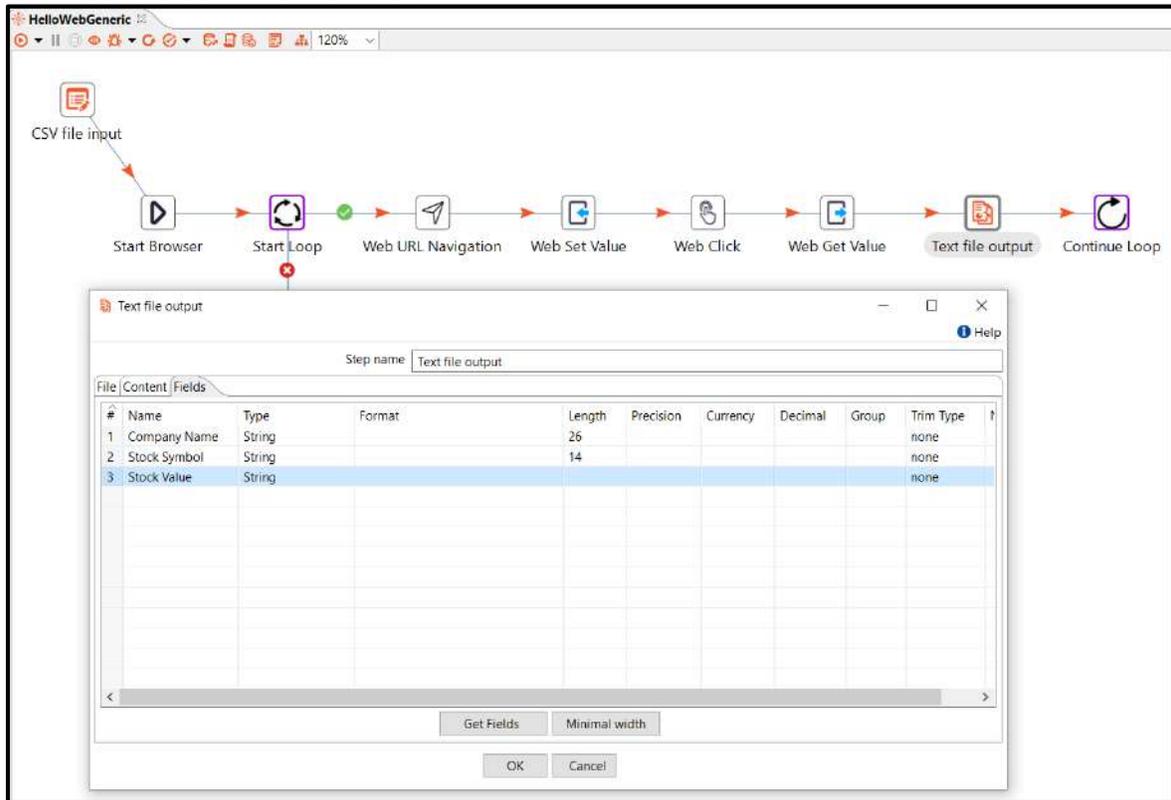
'Start Loop' Configuration Dialog:

- Step name: Start Loop
- Loop Name: PS_Start Loop79001226561600
- Conditional Loop?:
- Iterator Name: itr1
- Number of iterations: 1
- Additional Fields: (Empty table)
- Condition(s): (Empty field)

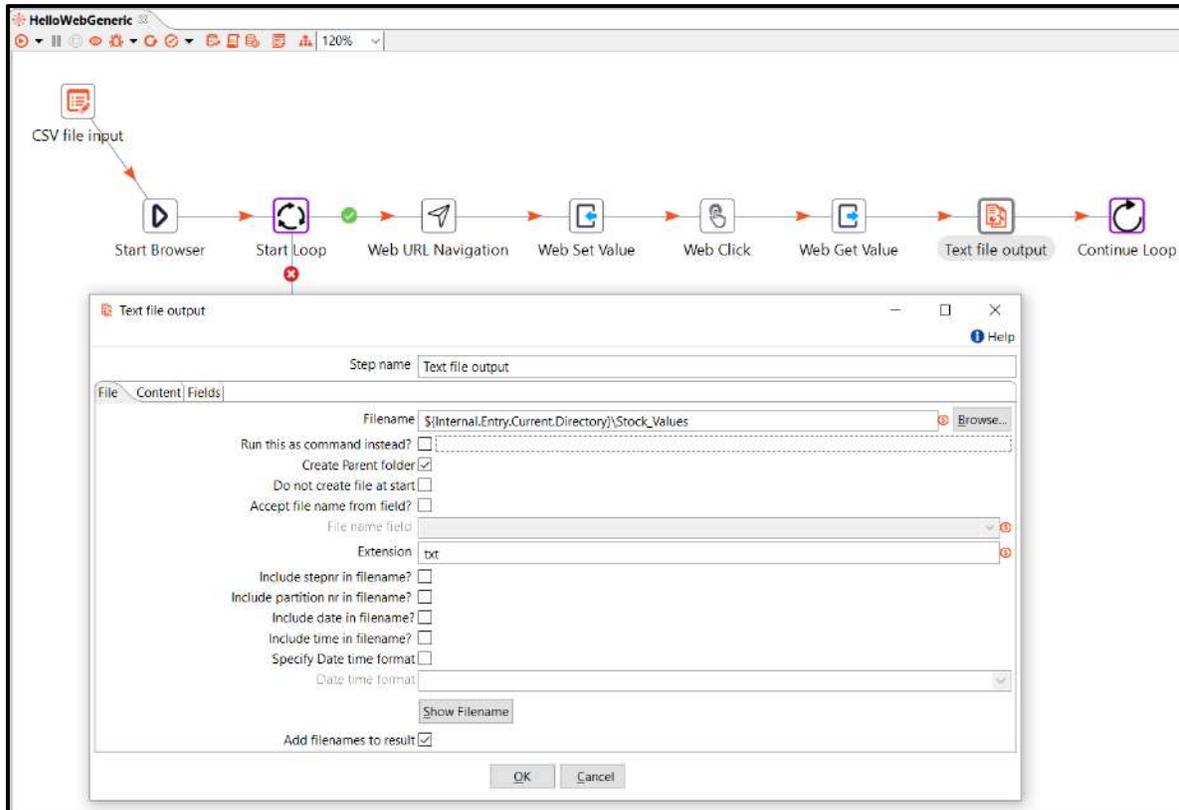
19. Also introduce a Text output file to capture the input fields from the csv file as well as the captured stock value. You can see the **Text file output** step at the desired location. Double click **Text file output** to configure it.
20. Browse to an output file path. In this case we are creating a new filename Stock_Values.
21. In the Content tab enable Append Checkbox.



22. Click Get Fields in the Fields tab.



23. Replace the file location with internal parameter `${Internal.Entry.Current.Directory}`.
`${Internal.Entry.Current.Directory}` is an internal parameter and refers the current location of the workflow. This gives portability to copy the workflow and file to any other location together.



Details about every parameter field are:

Parameter Field	Description	Comments
Parameter	Used for entering name of the Parameter	Name of the parameter, used internally everywhere.
Display Name	Used for entering Display Name of Parameter	Display name of the parameter, used to show the parameter in AutomationEdge Web UI. Note: Following special characters are allowed in parameter display name: . dot / slash , comma & ampersand () opening and closing bracket
Type	Used for selecting data type of the Parameter	Supported types are Boolean, Credential, Date, Enum, File, Integer, Number and String
List of Values	Used for entering List of Values for the Parameter	The field is enabled when parameter type is File. In this case we need to provide a comma separated list of file types(for e.g. .csv, .xlsx, .docx, .xism etc.)
Default Value	Used for entering Default value for the Parameter	Default value of the parameter
Secret	Boolean indicating if parameter is secret or not	Sensitive parameters like password should be marked as secret. AutomationEdge encrypts those parameters before saving then in database and doesn't show their values in the AutomationEdge Web UI.
Optional	Boolean indicating if parameter is optional or not	
Runtime	Boolean indicating if parameter is runtime or not	If value of this parameter is true, AutomationEdge prompts user to enter value of this parameter when running a workflow. If it is false, AutomationEdge expects users to configure it during design time.

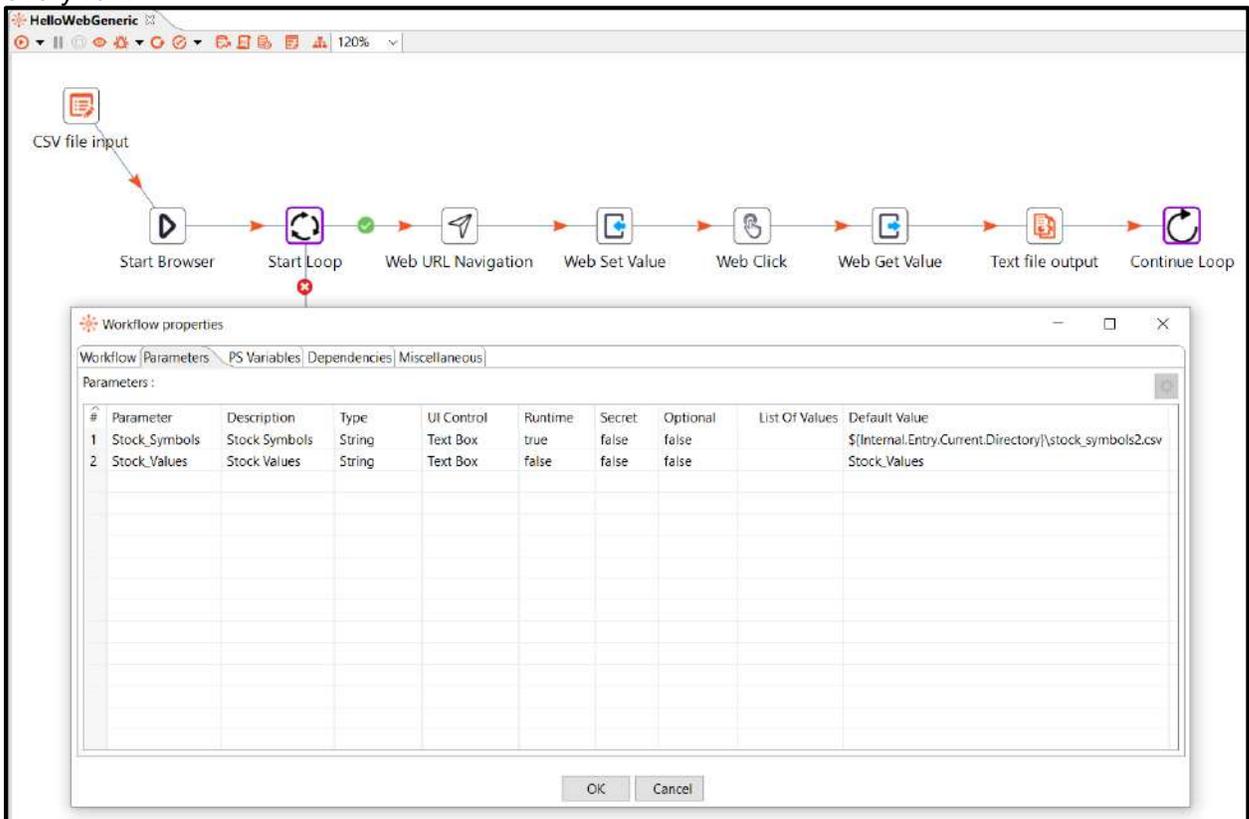
All the business processes developed in Process Studio ultimately have to be published to AutomationEdge Server so that they can be invoked by the users. Except for parameter name and type, all other parameter fields have significance when business process is published to **AutomationEdge Server** and invoked from there.

3.5 Parameterising HelloWebGeneric

Currently the workflow takes input file from a fixed location. At run time from AutomationEdge catalog we need the flexibility to browse a new file every time the workflow is executed. Also, we wish to have a desired name of the output file every time. In this section we will parameterize the input file and the output filename. Output files are stored on AutomationEdge server.

We will now revisit **HelloWebGeneric** workflow we built earlier and parameterize it.

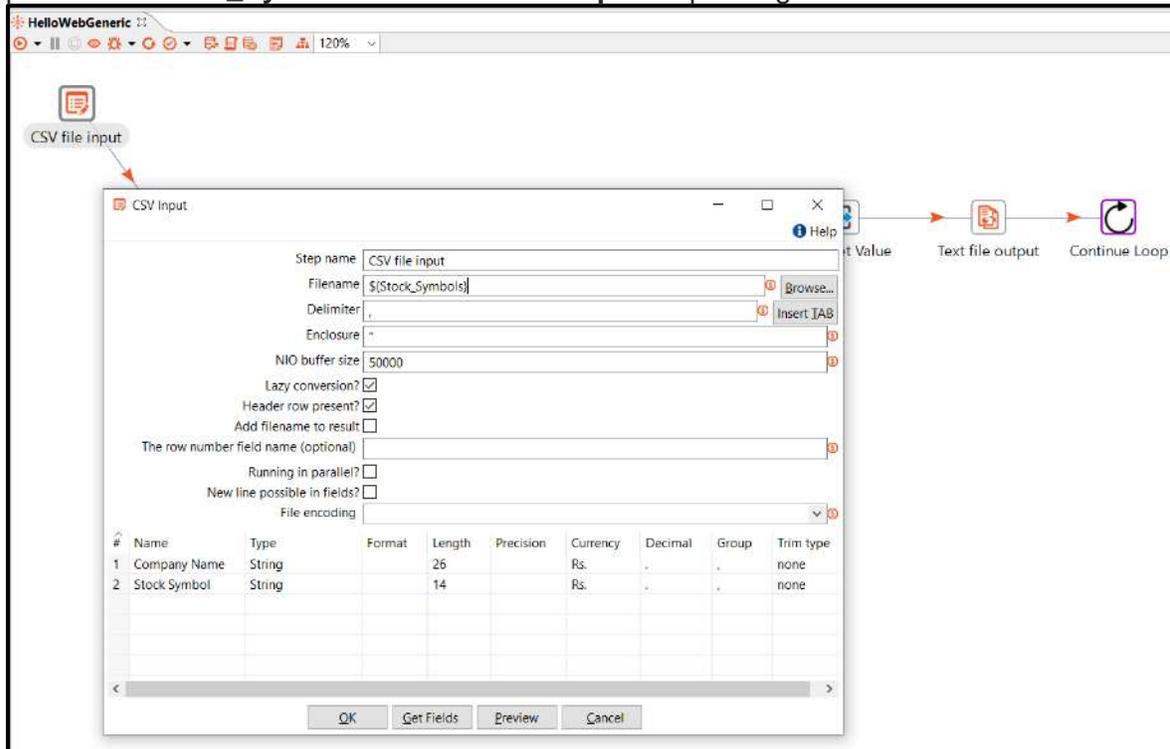
1. click anywhere on the empty space in the workflow. Create a parameter of type file for the input file and of type String for the output filename. For parameter of type file you need to provide the permissible file extensions. Optionally provide Default values for the parameters. This will ease workflow execution from Process Studio as you need not provide values for these parameters on every run.



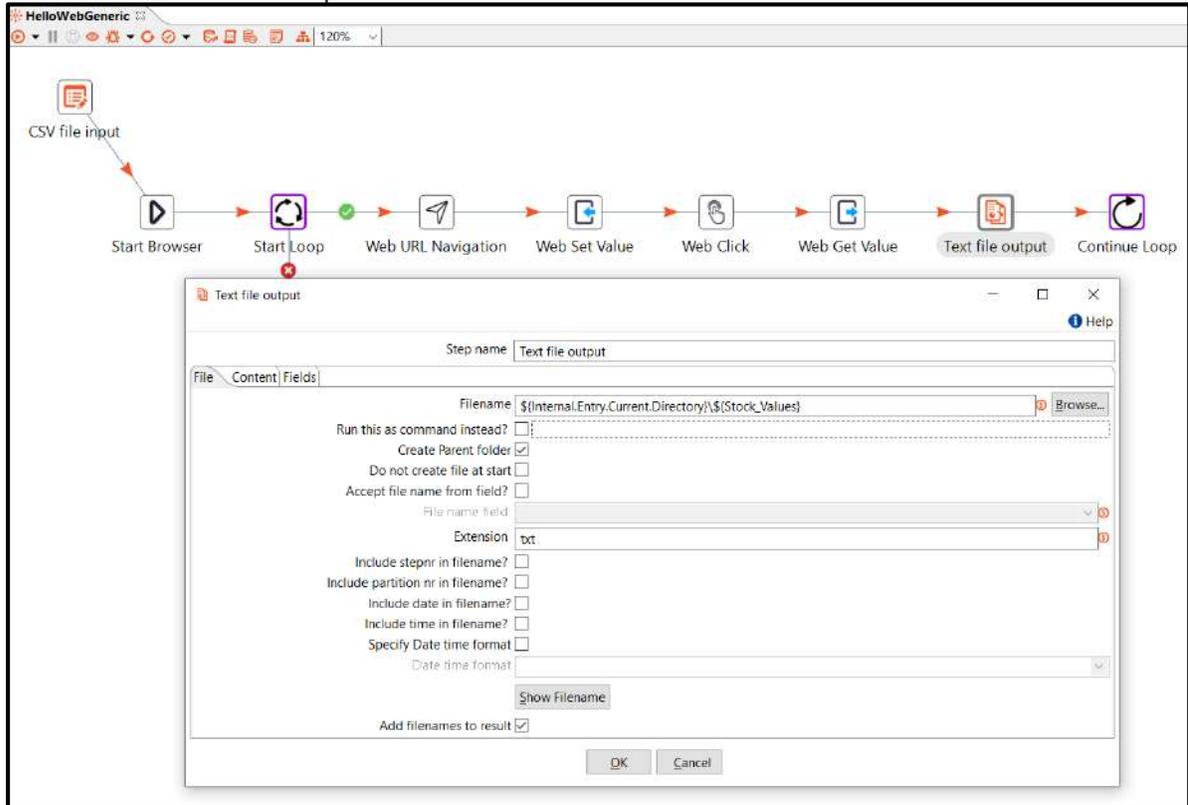
2. As seen in the windows explorer snapshot below stock_symbols2 is in the same directory as the workflow HelloWebGeneric.psw. Hence, we could use 'Internal.Entry.Current.Directory' parameter as seen below.

📄	HelloWeb.psw	28-10-2020 23:13	PSW File	23 KB
📄	HelloWebGeneric.psw	29-10-2020 10:01	PSW File	23 KB
📄	stock_symbols2	04-05-2018 15:44	Microsoft Excel Comma Sep...	1 KB

3. Replace `${Internal.Entry.Current.Directory}\stock_symbols2.csv` with the newly created parameter **Stock_Symbols** in the **CSV file input** step configuration. Click OK.



4. Replace `${Internal.Entry.Current.Directory}\Stock_Values` with `${Internal.Entry.Current.Directory}\${Stock_Values}`. We have created a parameter only for the filename and not the file path. Click OK.



5. Run the workflow and see the results in the **Preview data** tab as shown.

Execution Results

Logging Step Metrics Metrics Preview data Debug

First rows Last rows Off

#	Company Name	Stock Symbol	itr1	Stock Value
1	Alphabet Inc.	NASDAQ:GOOGL	0	\$2,505.15
2	Infosys Ltd.	NYSE:INFY	0	\$21.07
3	Samsung Electronics Co Ltd	KRX:005930	0	₩80,000.00
4	Apple	NASDAQ:AAPL	0	\$139.96
5	Amazon	NASDAQ:AMZN	0	\$3,510.98

6. In this project we created a sample RPA GUI automation workflow - HelloWebGeneric with multiple input records.

4 Project 3: Modularity

(Modularity with Parent-Child Workflows)

Child processes/workflows can be added to a workflow with Process Executor and Workflow Executor steps. Child processes/workflows can help in modularising large workflows.

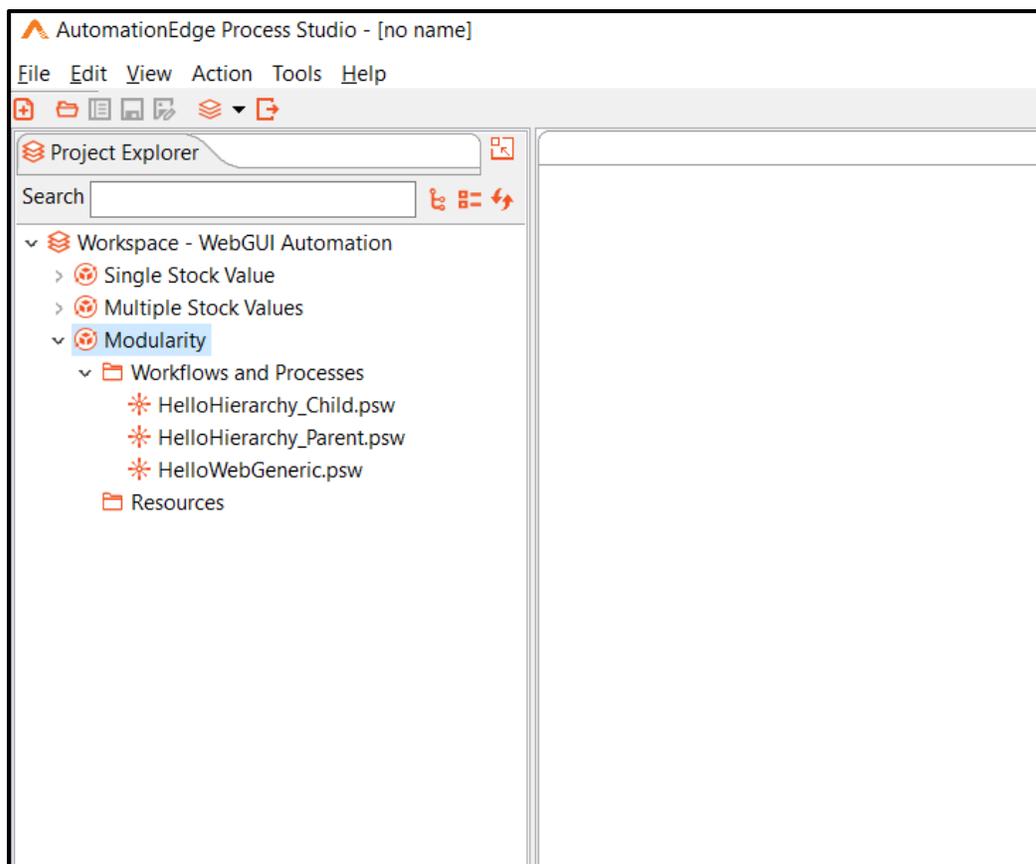
4.1 Building third RPA workflow with Child workflow: HelloHierarchy

In HelloHierarchy workflow we are introducing Workflow Executor step for modularization with parent child workflows.

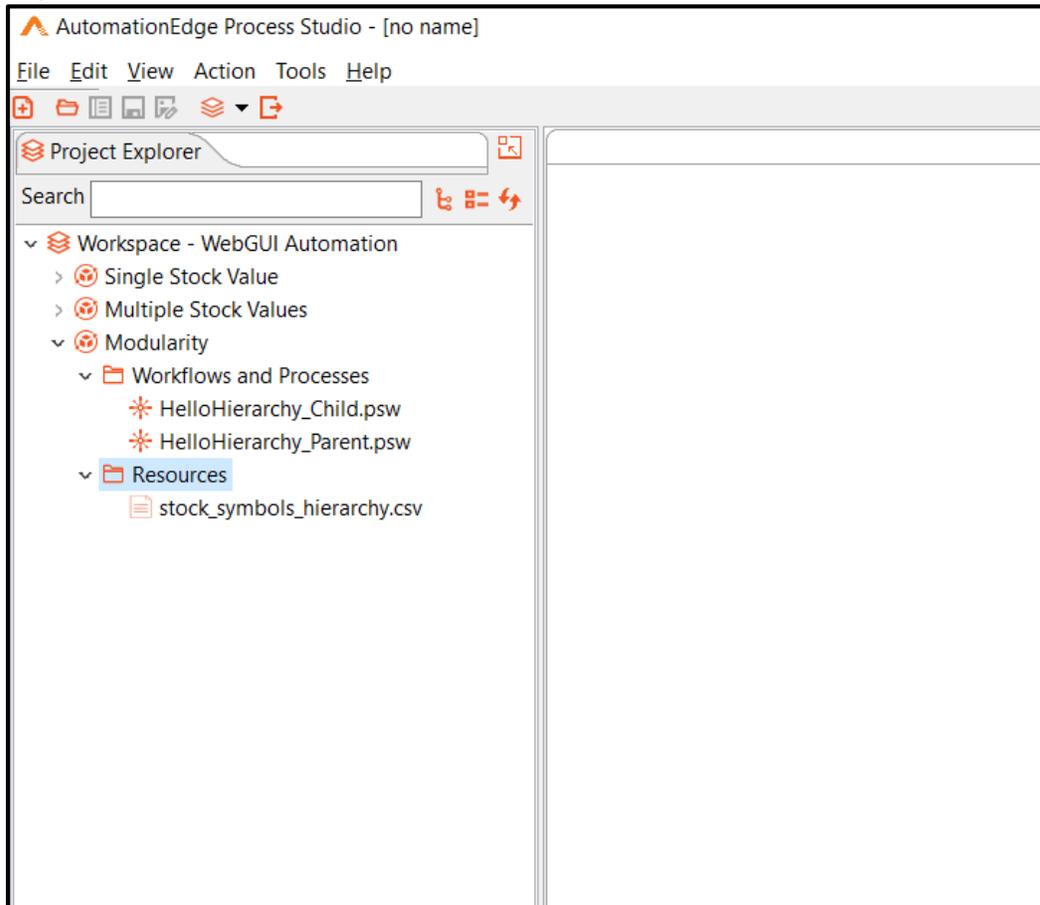
We will modify HelloWebGeneric workflow to introduce a Workflow Executor step and do away with Loop steps. The Workflow Executor step calls a child workflow which can combine steps to achieve rows serialization. Serialization will be achieved by passing one row at a time to the workflow executor step.

Following are the steps to create the workflows in a hierarchy. We will call the parent workflow HelloHierarchy_Parent and the child workflow HelloHierarchy_Child.

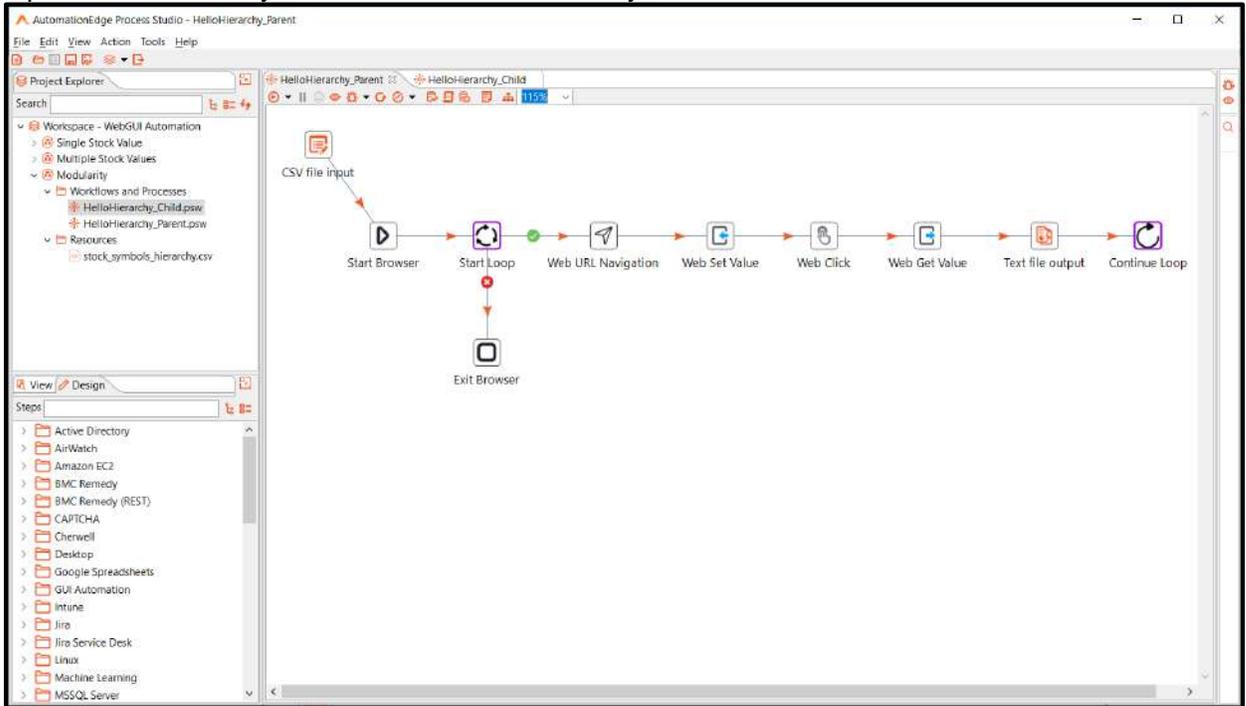
1. Firstly, export project 'Stock Values from Multiple Sites' and import it as a project Modularity.
2. Open HelloWebGeneric workflow. Create two copies of HelloWebGeneric, save the imported HelloWebGeneric as HelloHierarchy_Parent and HelloHierarchy_Child.



3. Delete the HelloWebGeneric workflow from Modularity project.
4. Right click on Resource and select Add file to add the file stock_symbols_hierarchy.



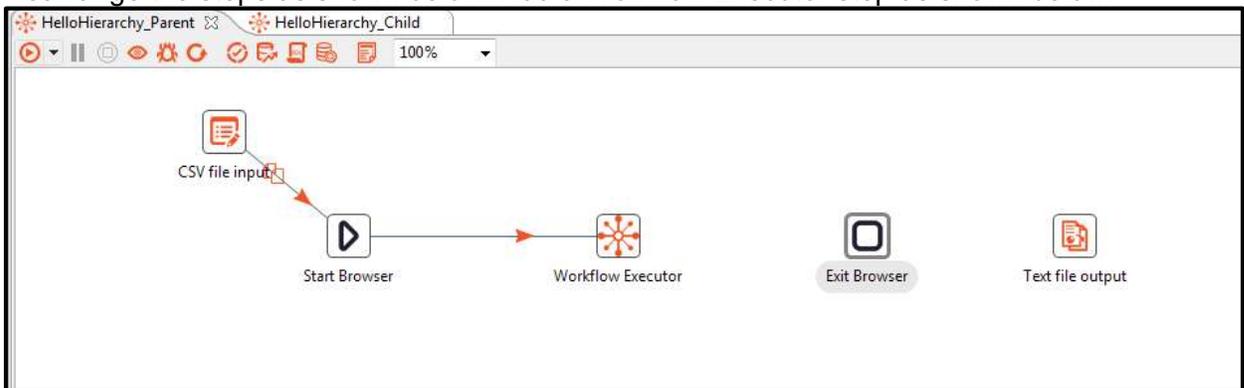
5. Open HelloHierarchy_Parent and HelloHierarchy_Child as seen below.



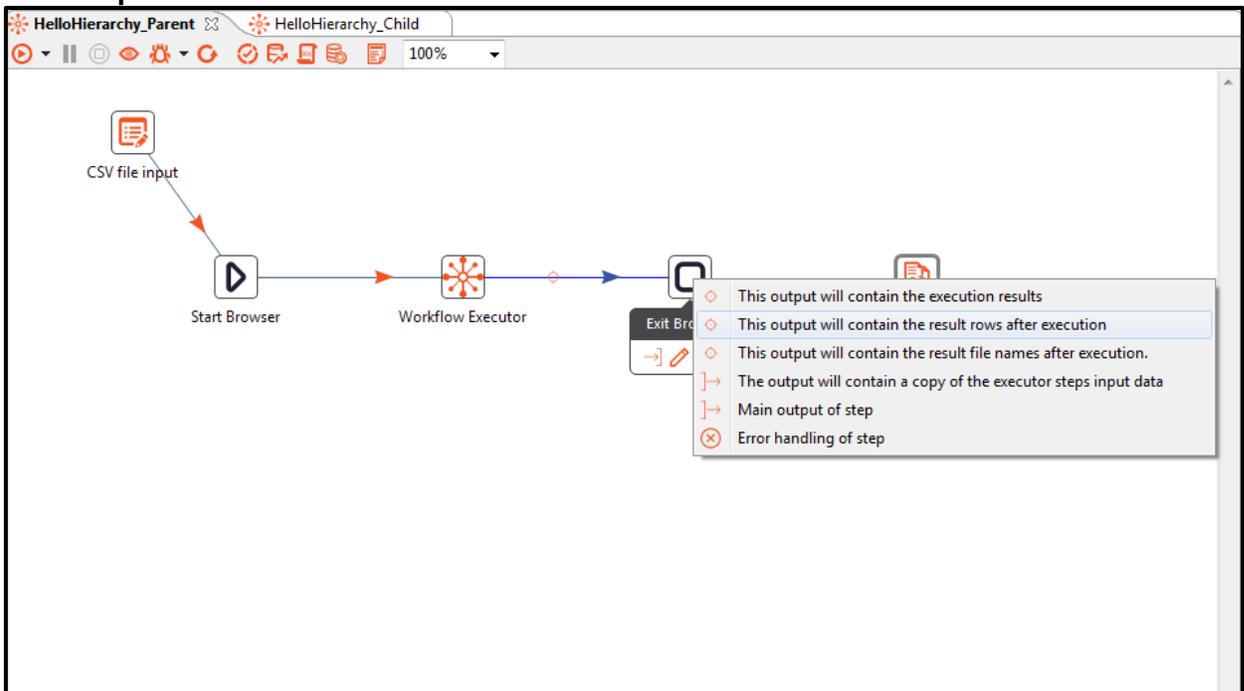
6. We want to have the data processing steps to the child workflow HelloHierarchy_Child. In the parent workflow i.e. HelloHierarchy_Parent delete all steps except those seen in the screen below.



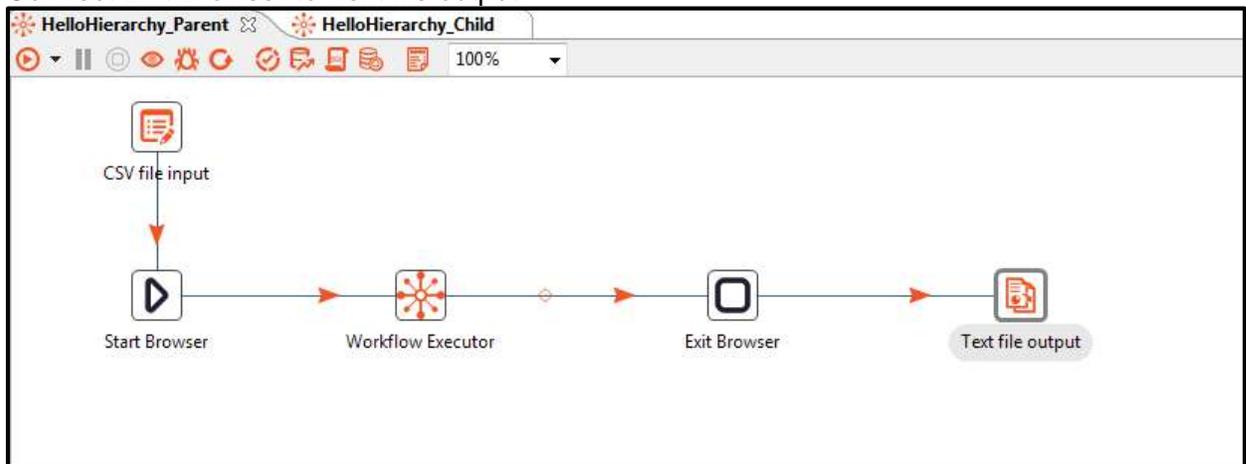
7. Rearrange the steps as shown below. Add a Workflow Executor step as shown below.



8. Make a connection between **Workflow Executor** and **Exit Browser** choose connection type as **This output will contain the result rows after execution**.



9. Connect Exit Browser to Text file output.

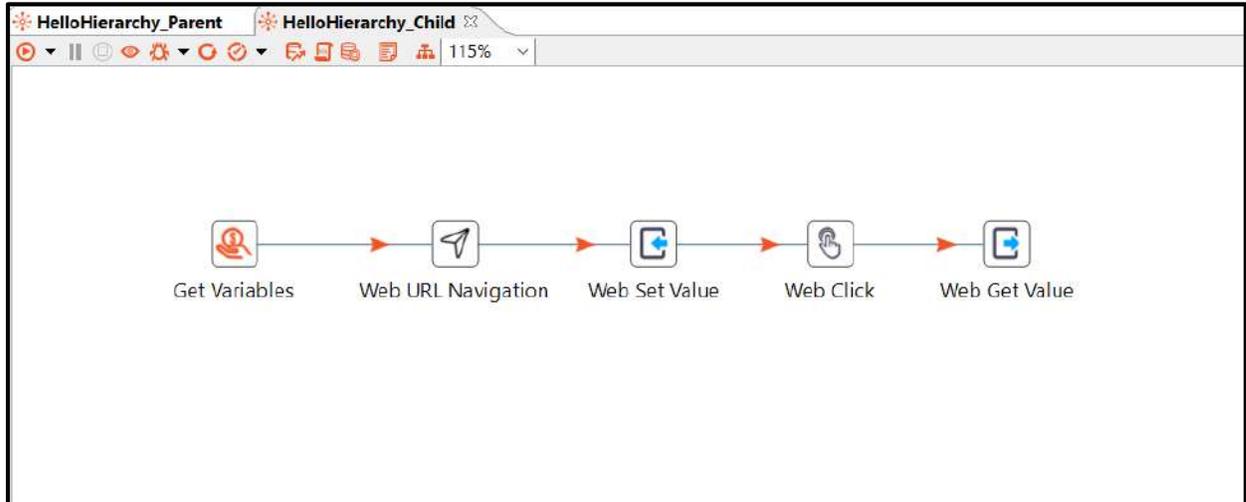


Before going to the next steps, let us take a look at different connection types for a connection starting from **Workflow Executor** step.

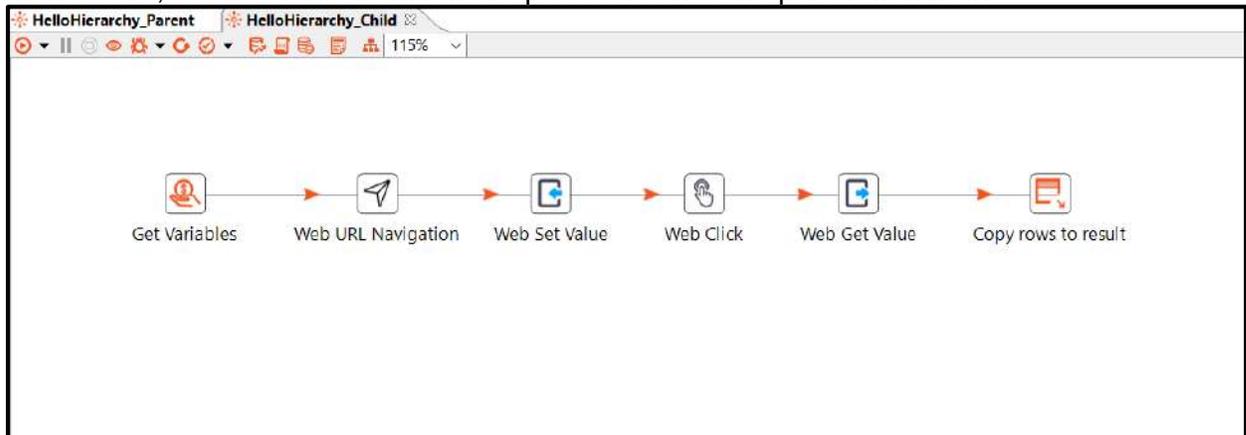
As shown in above image, there are six connection types for a connection starting from **Workflow Executor** step.

#	Connection Type	Description
1	This output will contain the execution results	Step after Workflow Executor in the calling workflow will get execution statistics of all the steps of the invoked workflow
2	This output will contain results rows after execution	Step after Workflow Executor in the calling workflow will get all the output rows generated by invoked workflow
3	This output will contain the result file names after execution	After you execute any workflow, outputs files generated can be made available in the workflow result. This connection type, output file(s) generated in the invoked workflow are available in the calling workflow
4	This output will contain a copy of the executor steps input data	Step after Workflow Executor in the calling workflow will get all the rows which were passed to the invoked workflow
5	Main output of step	Step after Workflow Executor gets rows neither of the calling workflow nor the called workflow
6	Error handling of step	Step after Workflow Executor is the error handling step in case workflow executor step fails.

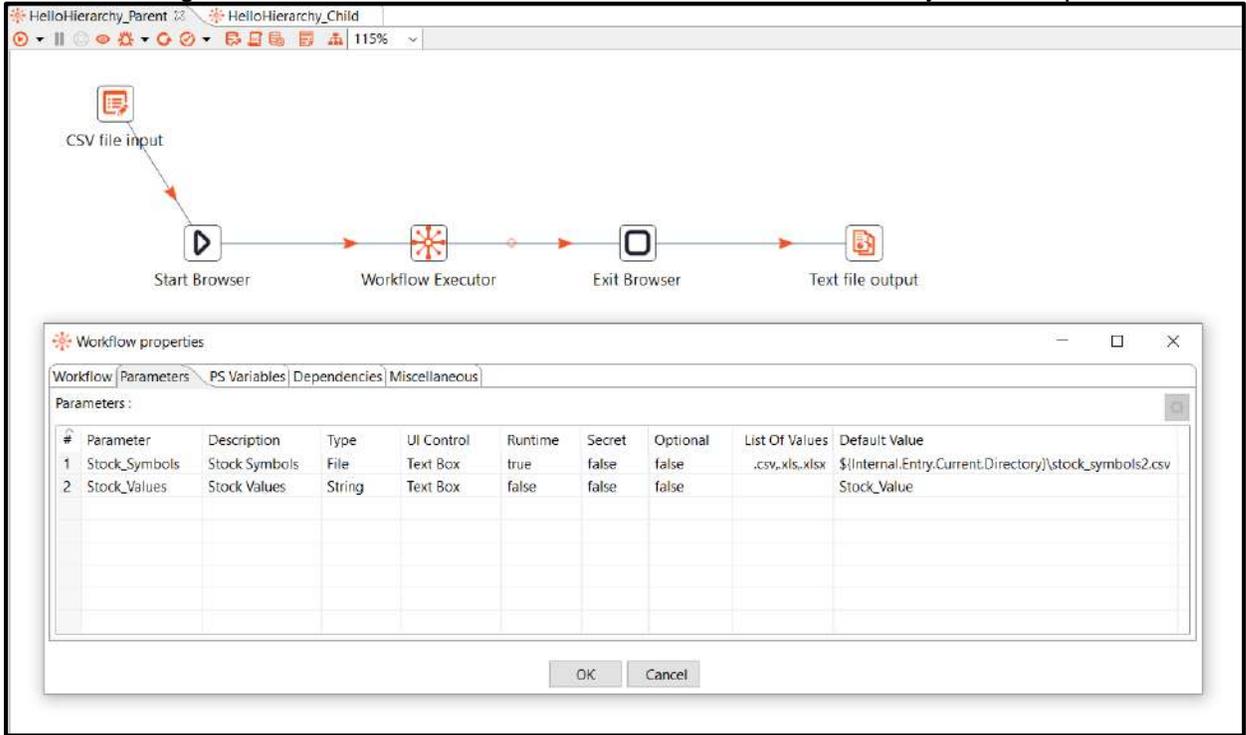
10. Open the child workflow HelloHierarchy_Child. Delete the CSV file input, Start Browser, Loop, Exit Browser, Text File Output steps.
11. For the workflow to do any meaningful work, there has to be one or more rows ingested into the workflow and HelloHierarchy_Child is no exception. For ingesting data into a workflow, we need a step of type input. In our example, we add a Get Variables step as the first step in the workflow. Get Variables acts like an input step. Make a connection starting from Get Variables to **Web URL Navigation 2** step as shown below.
12. The Child workflow looks as below.



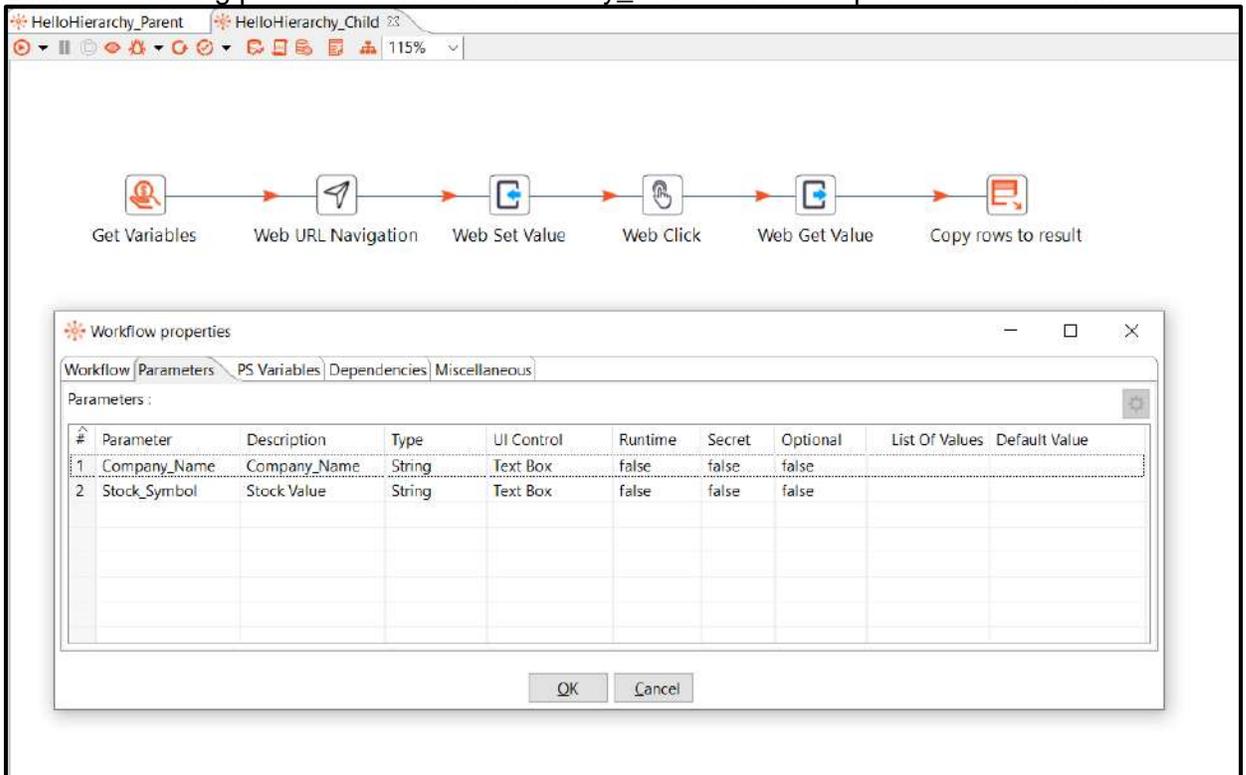
13. Add a Copy rows to result step in the end. Now drag step **Copy rows to result** and make a connection between **Web Get Value** and **Copy rows to result** step as shown below. This step makes sure, results of this workflow are passed back to the parent workflow.



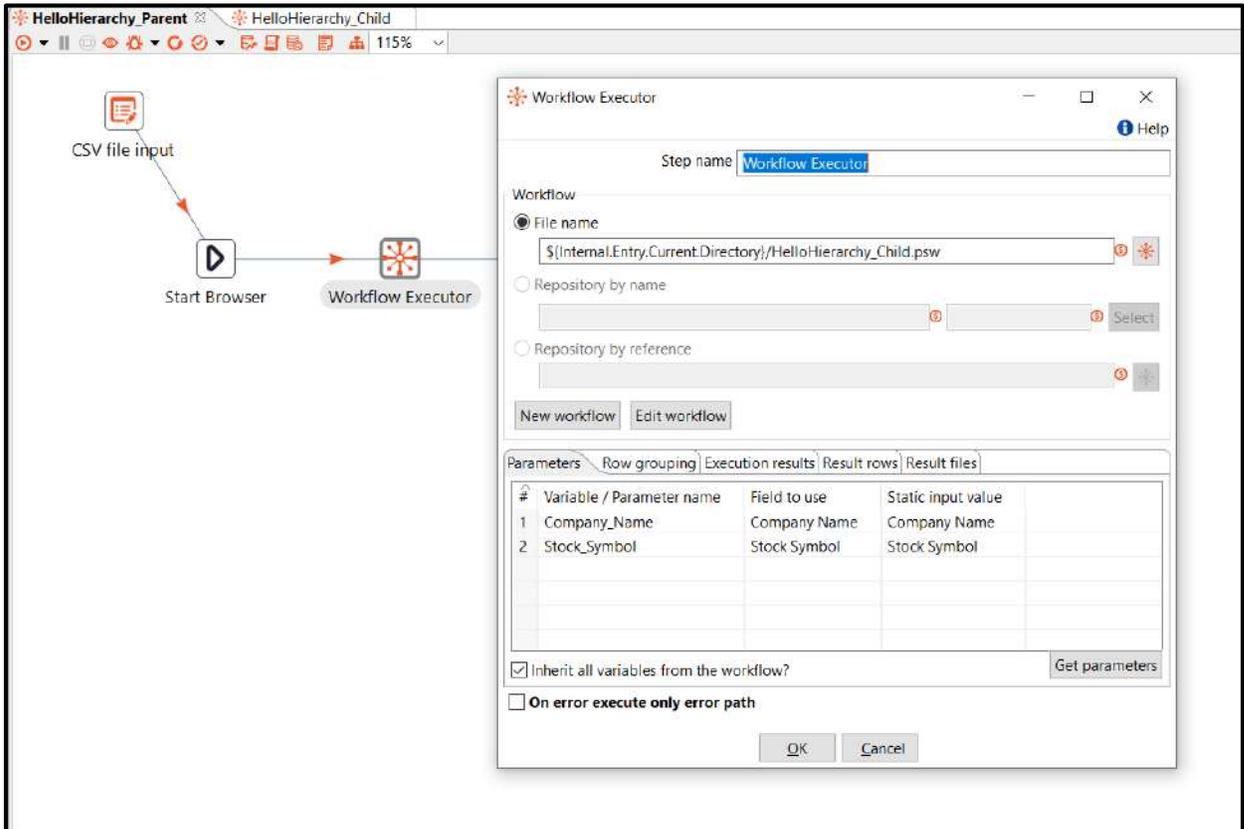
- Open Workflow properties by right clicking on the canvas and selecting Properties or by simply double clicking in the canvas area. Set the correct value for the Stock Symbols input file.



- Please note the column headers of the CSV file input in the HelloHierarchy_Parent. We need these columns in HelloHierarchy_Child to set Stock Symbols.
- Eliminate existing parameters in HelloHierarchy_Child and create parameters as shown below.



17. Double click Workflow Executor. Click **Browse...** button and choose HelloHierarchy_Child.psw workflow in **File name** field. Click **Get parameters**, this fetches parameters specified for workflow HelloHierarchy_Child.
18. Choose appropriate fields from the drop down list in the Field to use column. This way, we have essentially passed fields in the workflow HelloHierarchy_Parent as a parameter to workflow HelloHierarchy_Child, by mapping the fields to the parameters. This will happen for every row processed by HelloHierarchy_Parent workflow.

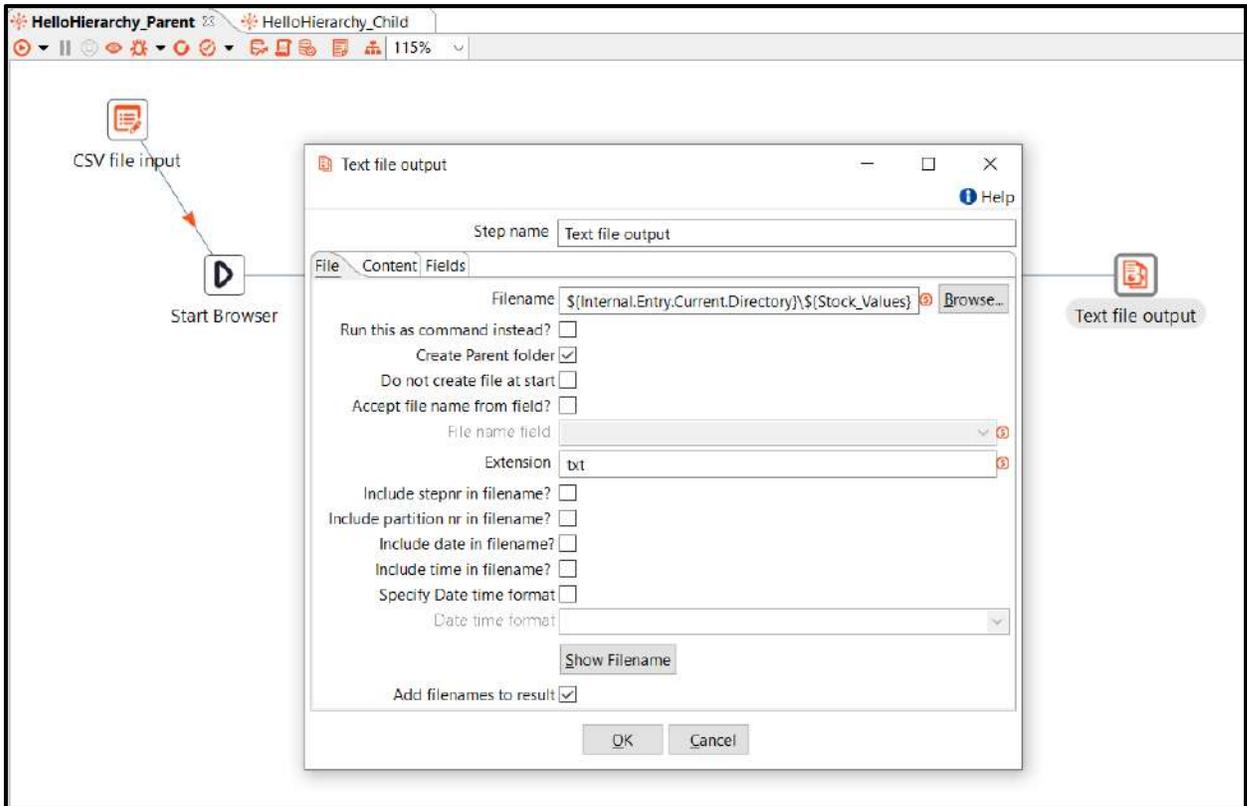


The screenshot shows the Process Studio interface. On the left, a workflow diagram is visible with a 'CSV file input' node connected to a 'Start Browser' node, which is then connected to a 'Workflow Executor' node. On the right, the 'Workflow Executor' configuration dialog is open. The 'Step name' is 'Workflow Executor'. Under the 'Workflow' section, 'File name' is selected, and the path is '\$[Internal.Entry.Current.Directory]/HelloHierarchy_Child.psw'. Below this, there are options for 'Repository by name' and 'Repository by reference', both currently empty. There are 'New workflow' and 'Edit workflow' buttons. The 'Parameters' tab is active, showing a table with the following data:

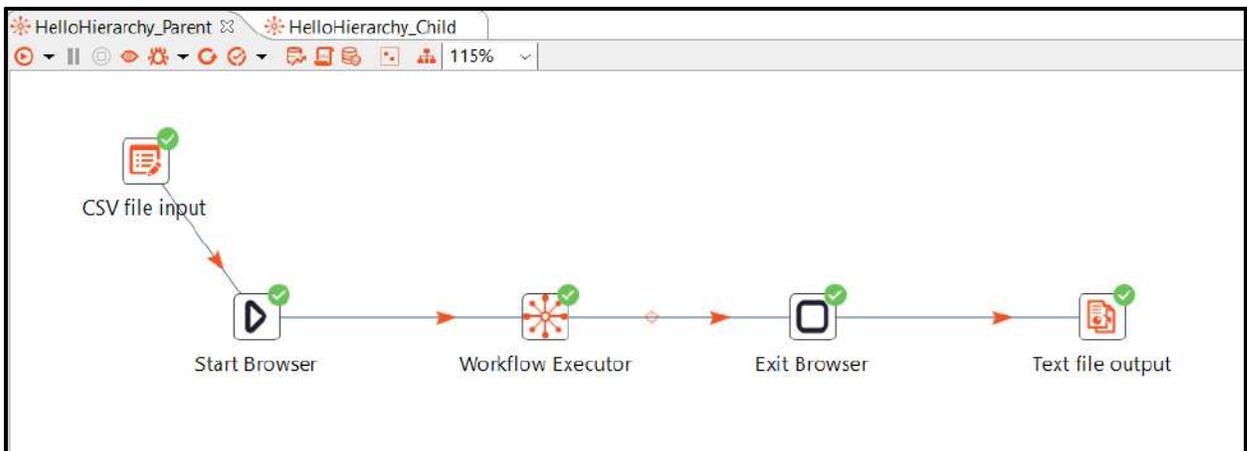
#	Variable / Parameter name	Field to use	Static input value
1	Company_Name	Company Name	Company Name
2	Stock_Symbol	Stock Symbol	Stock Symbol

At the bottom of the dialog, there is a checked checkbox for 'Inherit all variables from the workflow?' and an unchecked checkbox for 'On error execute only error path'. There is a 'Get parameters' button and 'OK' and 'Cancel' buttons at the bottom.

- Configure **Text file output** step by double clicking on it. Specify a filename where you would want stock prices to be saved. You may leave the fields tab blank. The output, rows from the child workflow will be populated here.



- Run HelloHierarchy_Parent. You can see a successfully executed workflow below.



21. Examine the contents of file Stock_Vaues.csv. You will see a row corresponding to each stock and a header row with column names corresponding to the output rows from workflow executor step.

Company Name;	Stock Symbol;	Stock Value
NASDAQ:GOOGL;	Alphabet Inc. ;	\$2,381.18
NYSE:INFY ;	Infosys Ltd. ;	21,388.95
KRX:005930 ;	Samsung Electr;	280,400.00
NASDAQ:AAPL ;	Apple ;	\$124.28
NASDAQ:AMZN ;	Amazon ;	\$3,218.65

5 Project 4: Tables & Lists

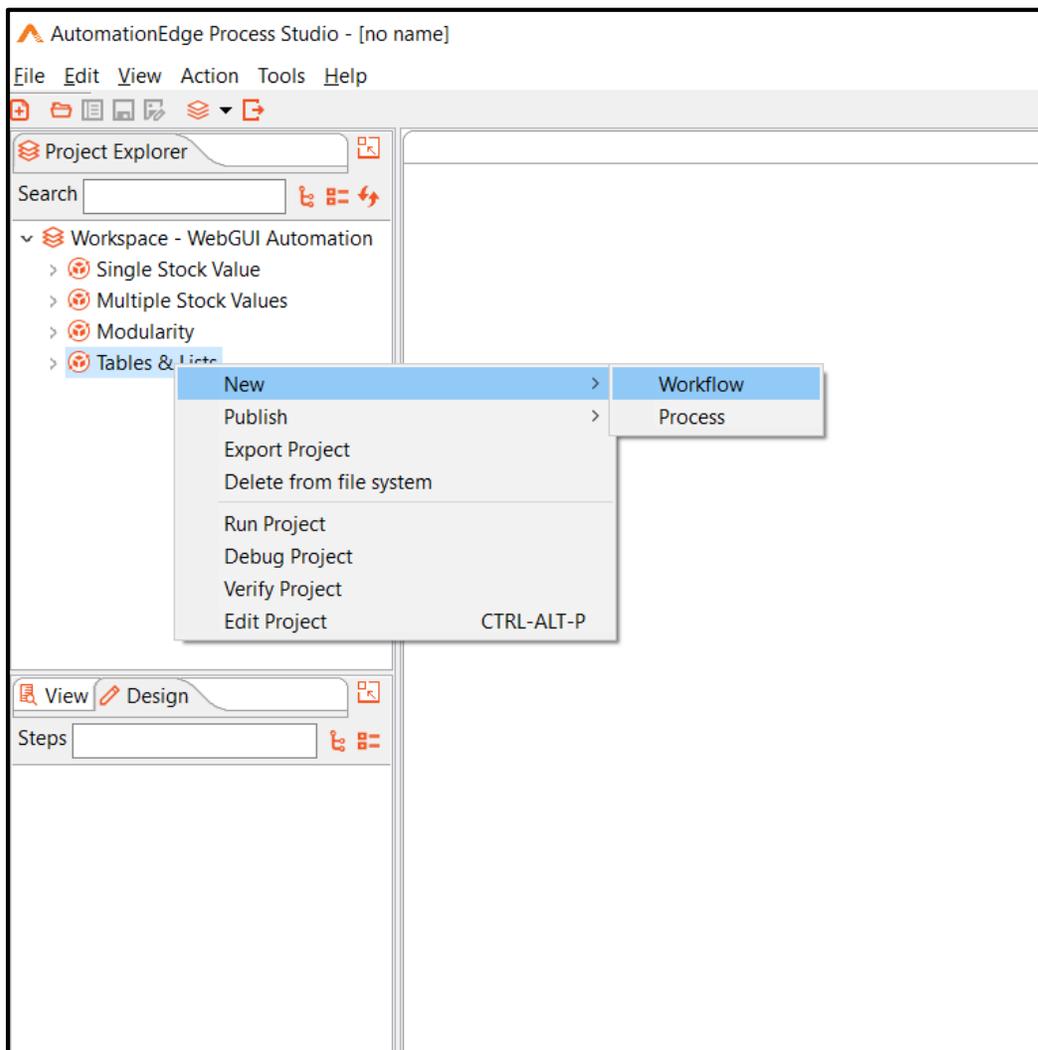
(More techniques for RPA Workflows)

5.1 Workflow: HelloWebLoopTable

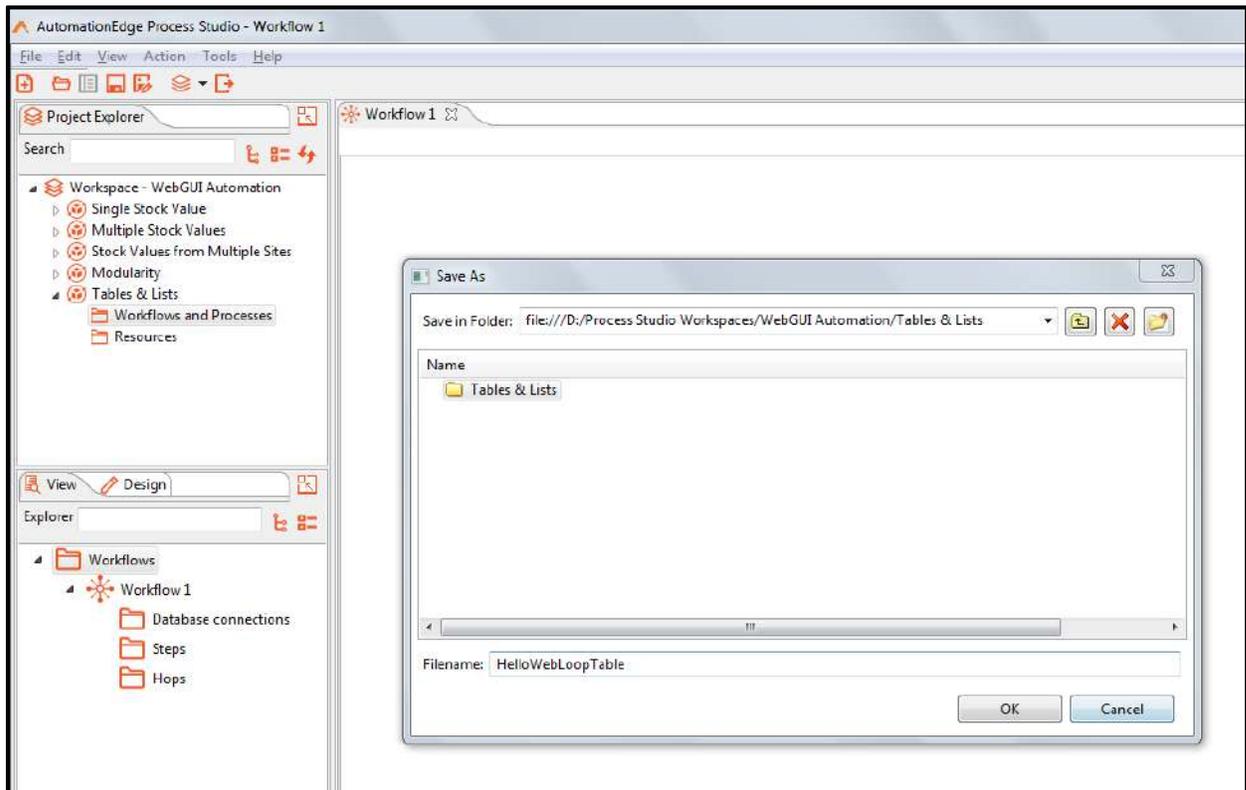
(Building GUI Automation for web workflow with Web Loop table: HelloWebLoopTable)

In this exercise will build a Web Loop Table workflow and call it, HelloWebLoopTable. In HelloWebLoopTable we will use Web Loop Table step to automate reading pages and fetching values in the AutomationEdge Requests menu.

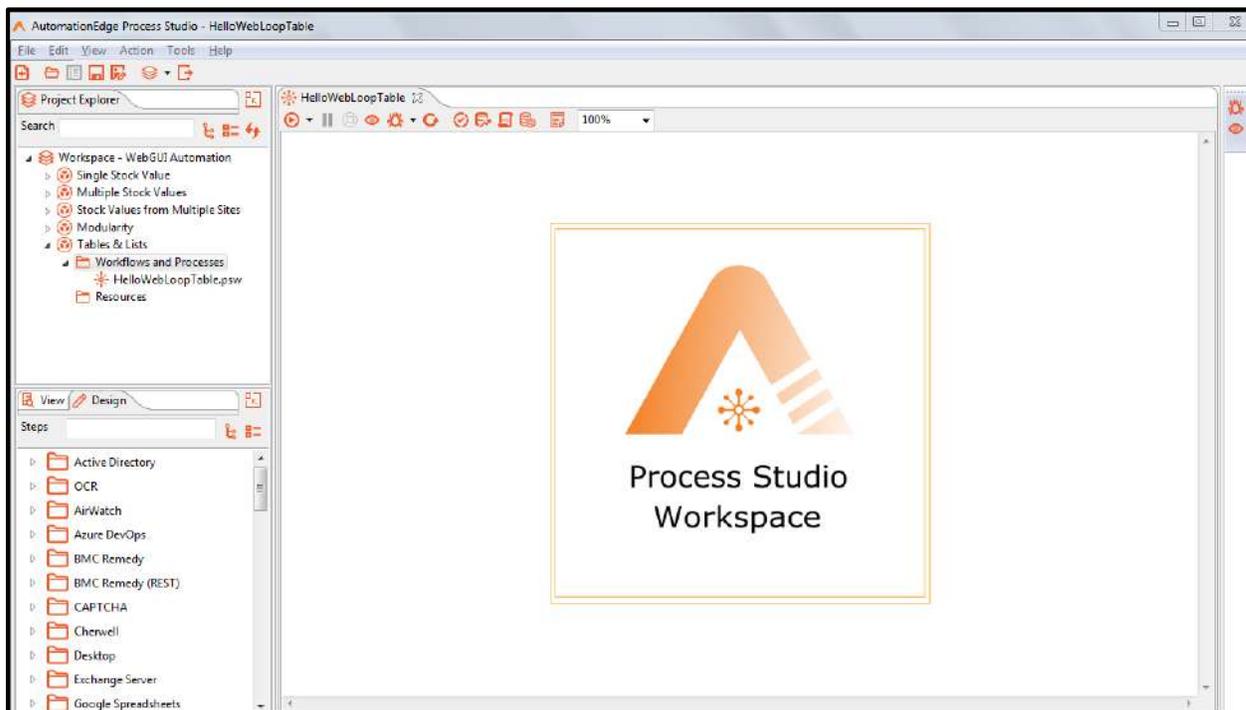
1. Open the workspace WebGUI Automation. Right click on the workspace and select New Project to create a new project – Tables & Lists.
2. Create a new workflow in the project.



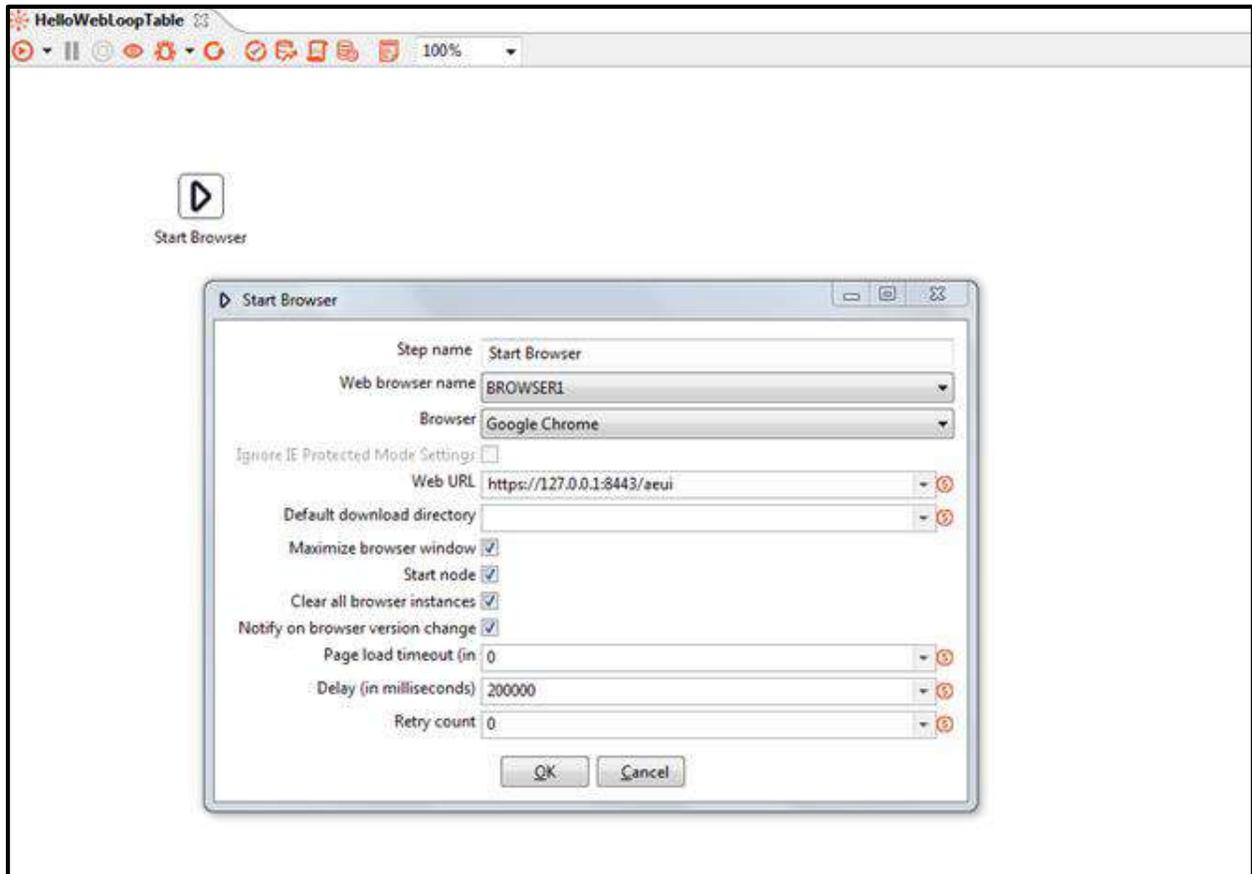
- Name of the workflow to **HelloWebLoopTable**. Workflow is stored in an XML file with extension psw which stands for Process Studio Workflow. It is stored in the directory path of the project.



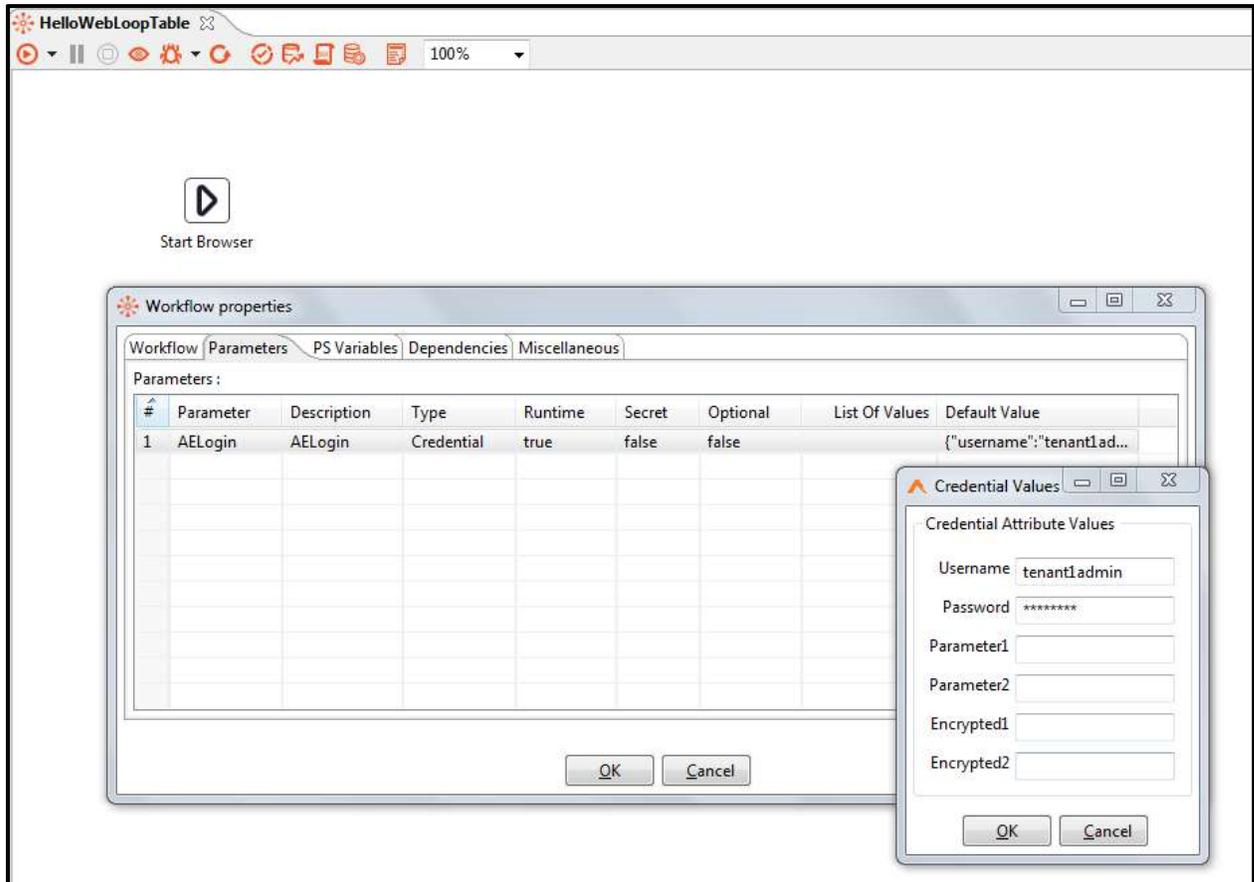
- Type **Start Browser** in the text box next to **Steps** in the left pane. All the steps having **Start Browser** in their name will be displayed.



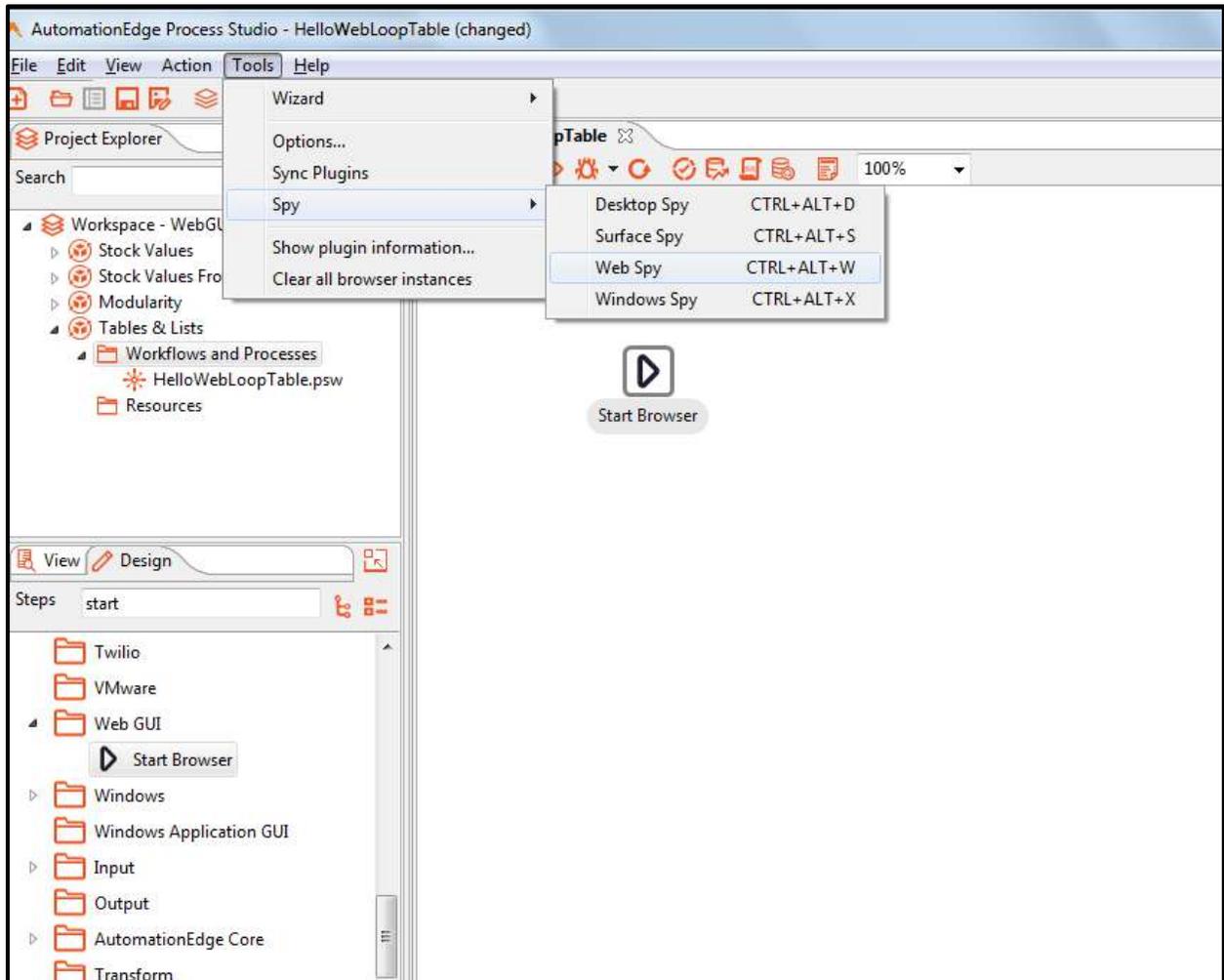
5. Drag **Start Browser** step from the left pane and drop it in the right pane. Set the maximum global delay for the entire session as 200000 milliseconds. Configure other options as shown below.



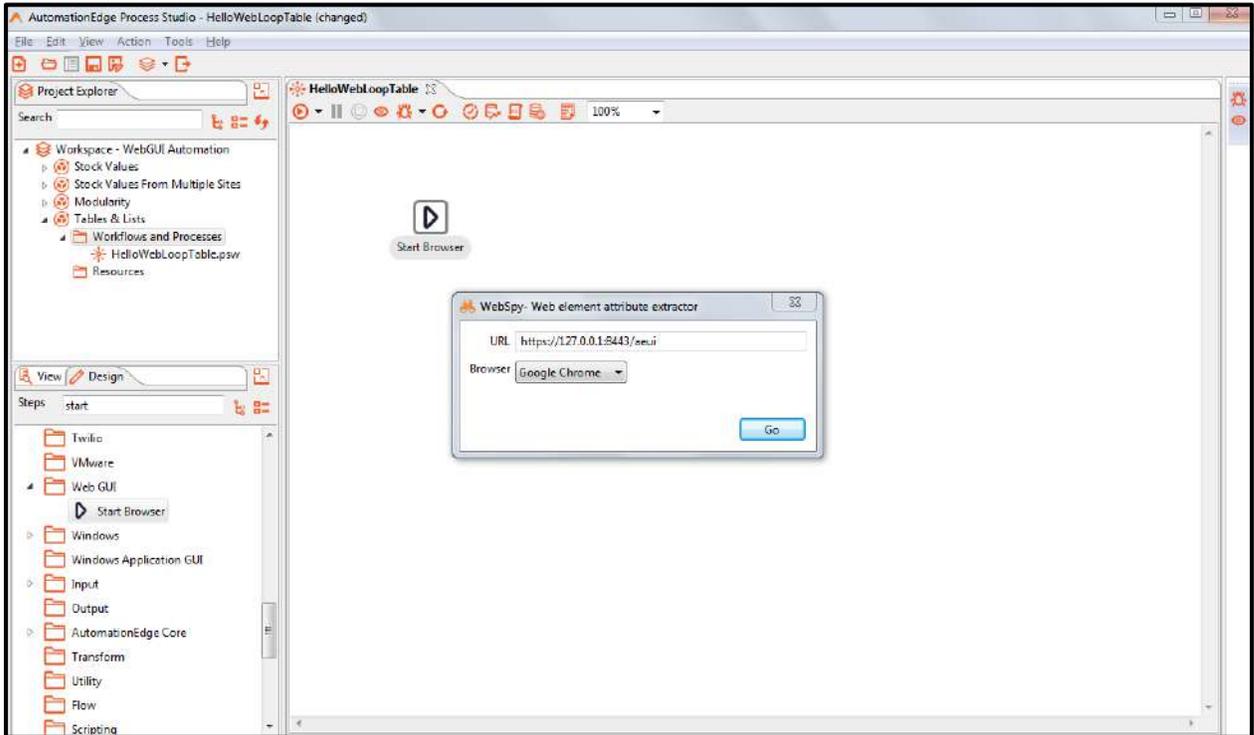
6. Double click in the workspace canvas in the right pane and Workflow properties dialog box appears. Create a parameter AELogin of type credential to store the username and password of the AutomationEdge cloud instance.



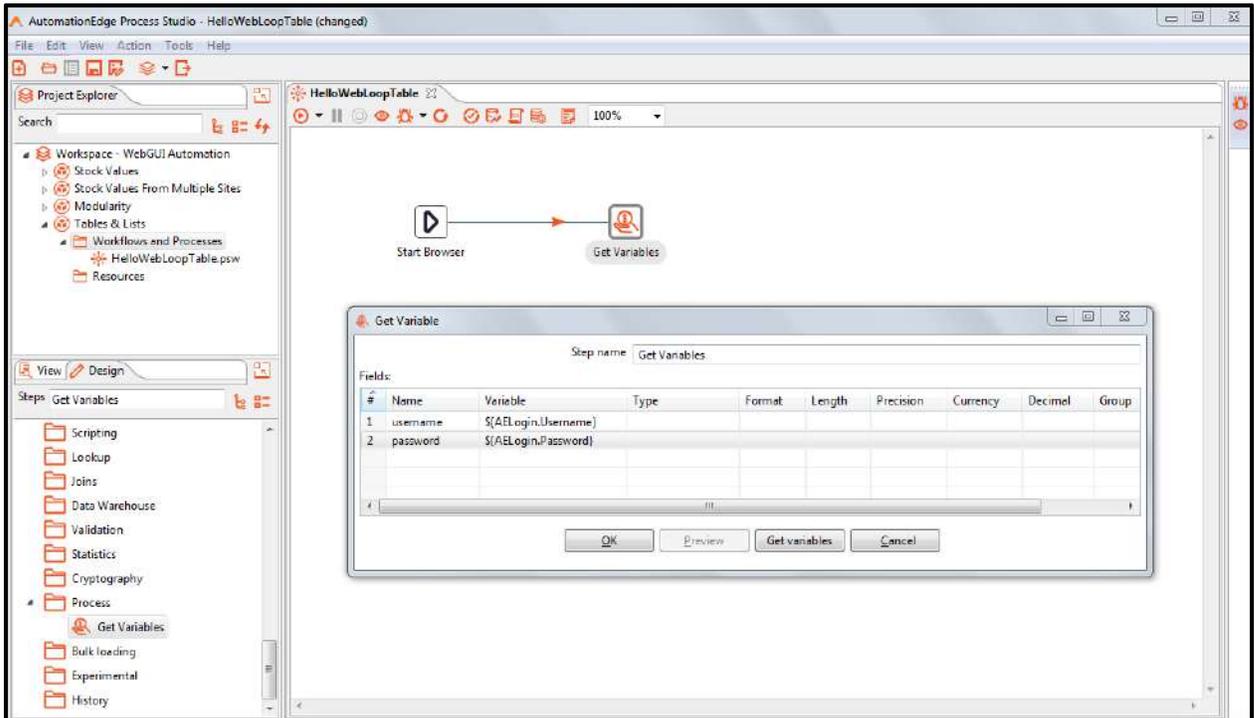
7. We shall use AutomationEdge GUI Spy to identify element locators. Click GUI Spy under Spy menu.



- Type the URL of the AutomationEdge instance to spy for locators (<https://www.t3.automationedge.com> in case of AutomationEdge Cloud Trial instance), it opens the URL in the browser selected. The GUI Spy menu also opens in the background.



- Drag and drop Get Variable step. Create two fields username and password and search for the corresponding AELogin variables defined earlier.

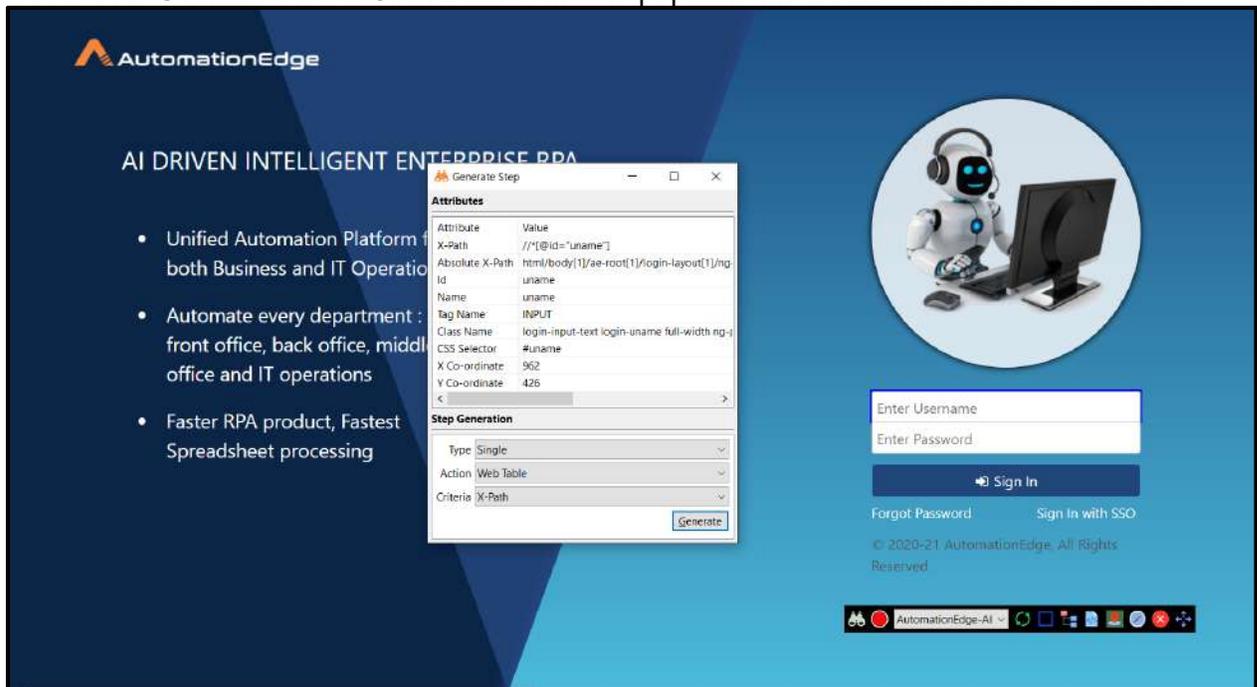


10. GUI Spy opens the AutomationEdge URL. Bring the GUI Spy window on top of the web page.



11. Make sure you are in Spy Mode. The Spy icon changes from orange to white.

12. Hover over Username field. Click Back Tick ‘`’ to populate the locator Attributes and values.



13. We will create a new Web Composite step. As a first entry in the Web Composite step we wish to set value of the username. Choose appropriate details. In this case chose criteria Id as seen below to create a new Web Composite step. Click Generate.
14. A message in green text appears at the bottom of the Generate step - Web Composite Step is added to the workflow is displayed.

AI DRIVEN INTELLIGENCE

- Unified Automation Platform for both Business and IT Operations
- Automate every department: front office, back office, middle office and IT operations
- Faster RPA product, Spreadsheet processes

Generate Step
✖

Attribute	Value
X-Path	//*[@id="username"]
Absolute X-Path	html/body[1]/ae-root[1]/login-layout[1]/ng-comp
Id	username
Name	username
Tag Name	INPUT
Class Name	login-input-text login-uname full-width ng-pristine
CSS Selector	#username
X Co-ordinate	962
Y Co-ordinate	426

Step Generation

Type: Composite

Step: New

Action: Set Value

Value: tenantcloud

Criteria: X-Path

[Generate](#)

Step [Web Composite] generated in workflow [HelloWebLoopTable]



Enter Username

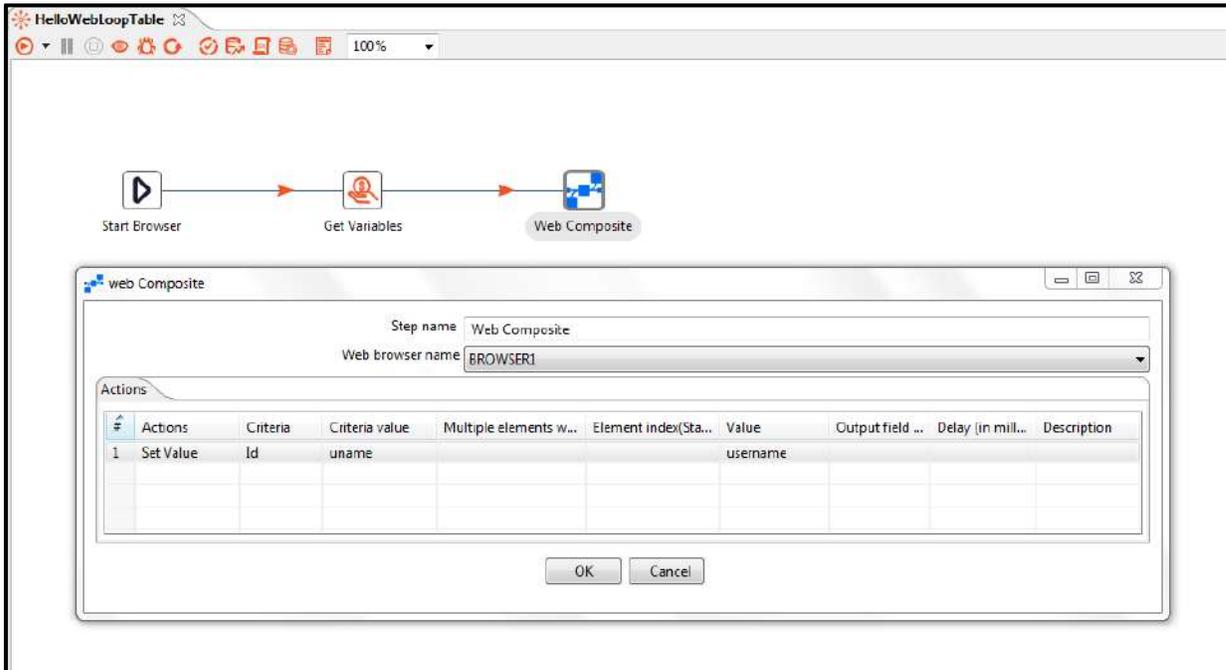
Enter Password

[Sign In](#)

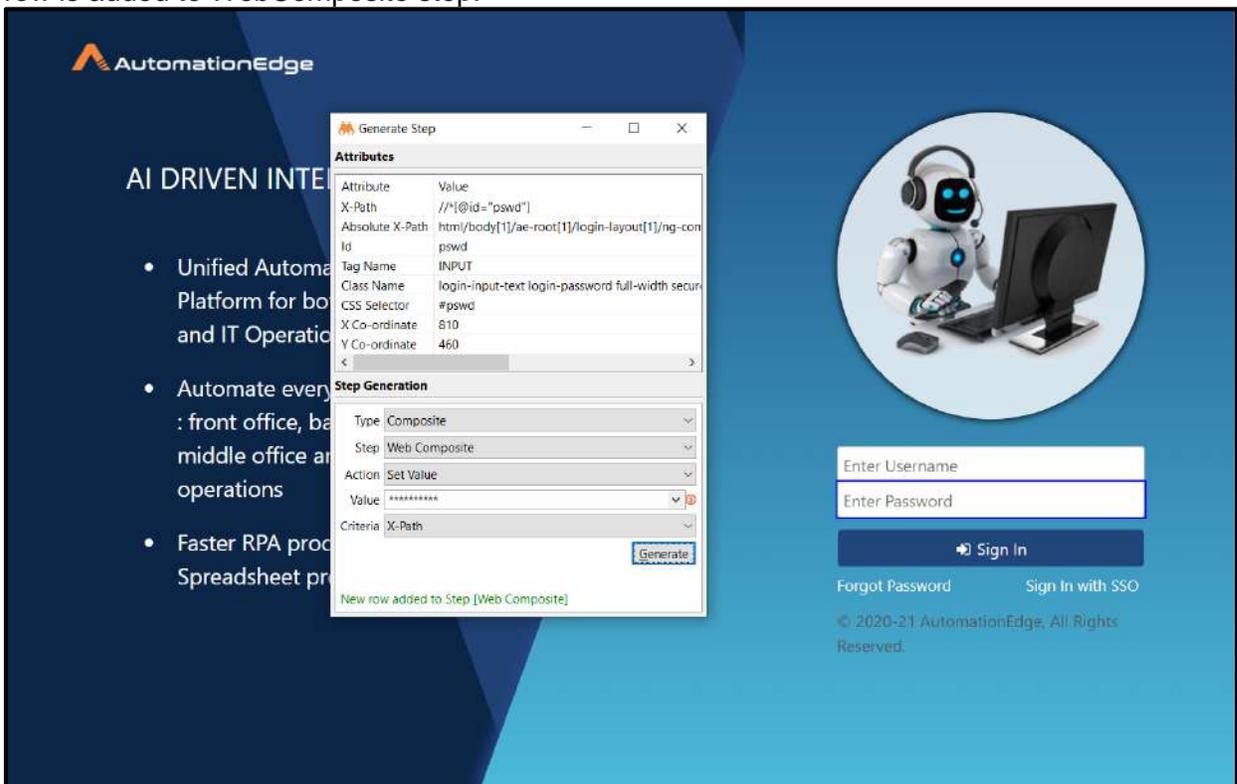
[Forgot Password](#) [Sign In with SSO](#)

© 2020-21 AutomationEdge, All Rights Reserved.

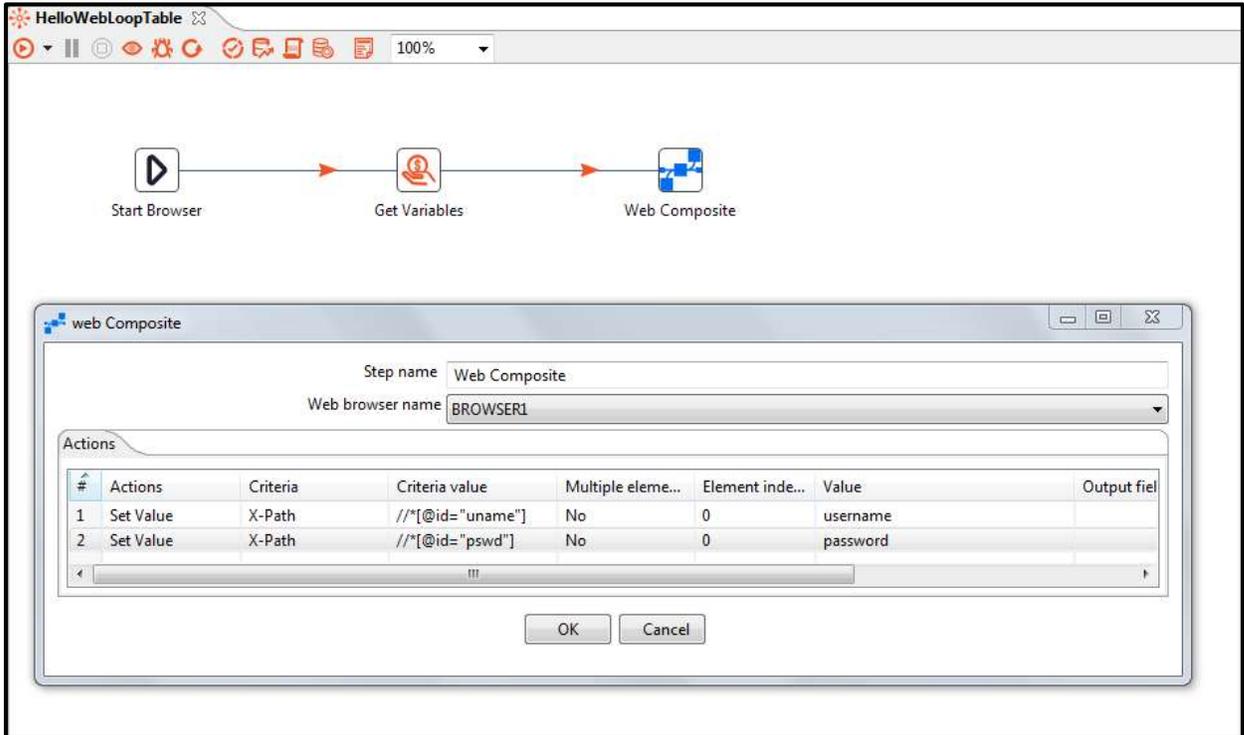
15. A Web Composite step is created as seen below. In this case you can also change the criteria to Id as seen below. Select username from the drop down list for Value.



16. Hover over Password to see the blue box. Add to the same Web Composite step. Choose Web Composite step from the Step drop down list as seen below. Choose other details appropriately. Click Generate. A message in green text appears at the bottom of the Generate step, that new row is added to WebComposite step.

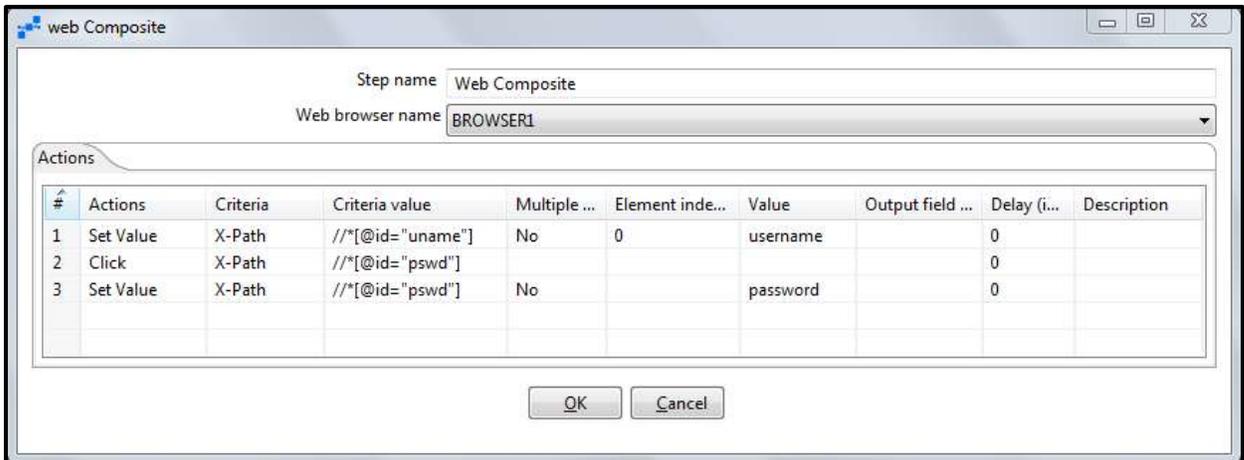


17. A Web Composite step is created as seen below. Select password from the drop down list for Value.

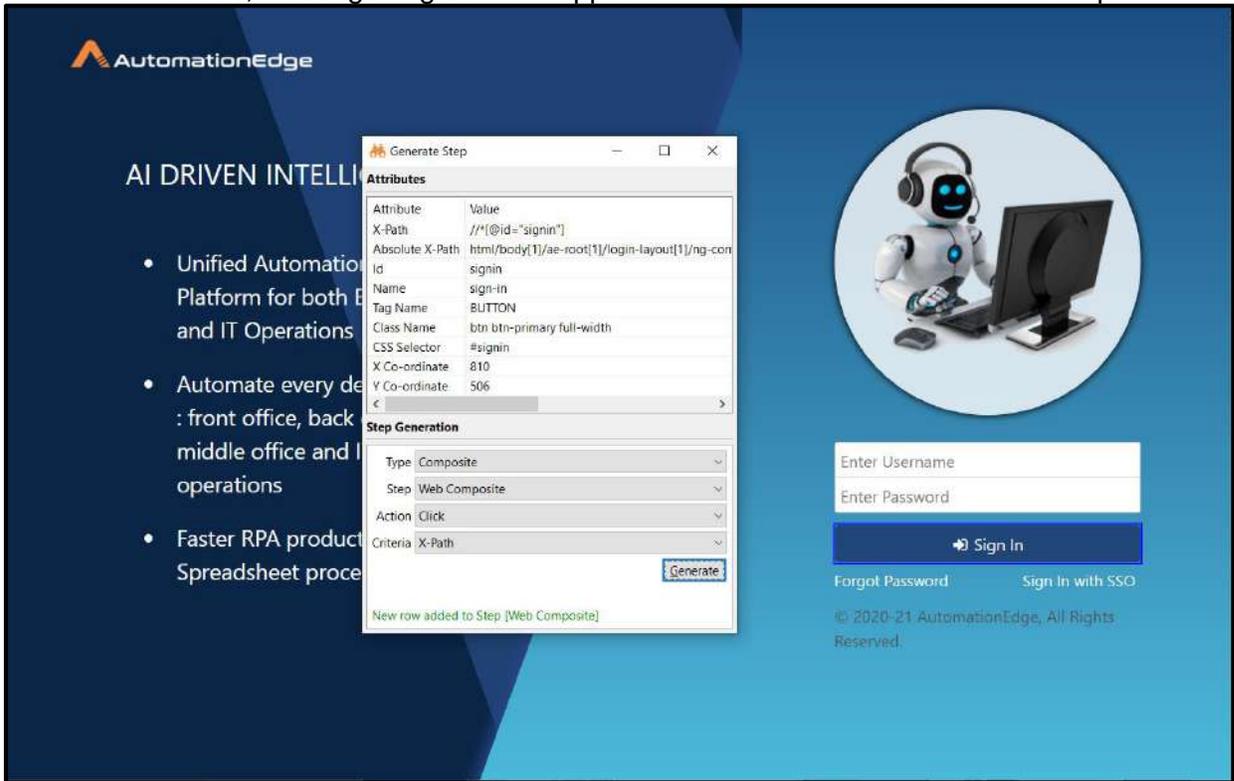


18. In case Web Set Value step cannot be used to set value in Password field, first click on the password field and then use Web Set Value. In this case redesign the Web Composite step with an additional Web Click step before Web Set Value for password as follows.

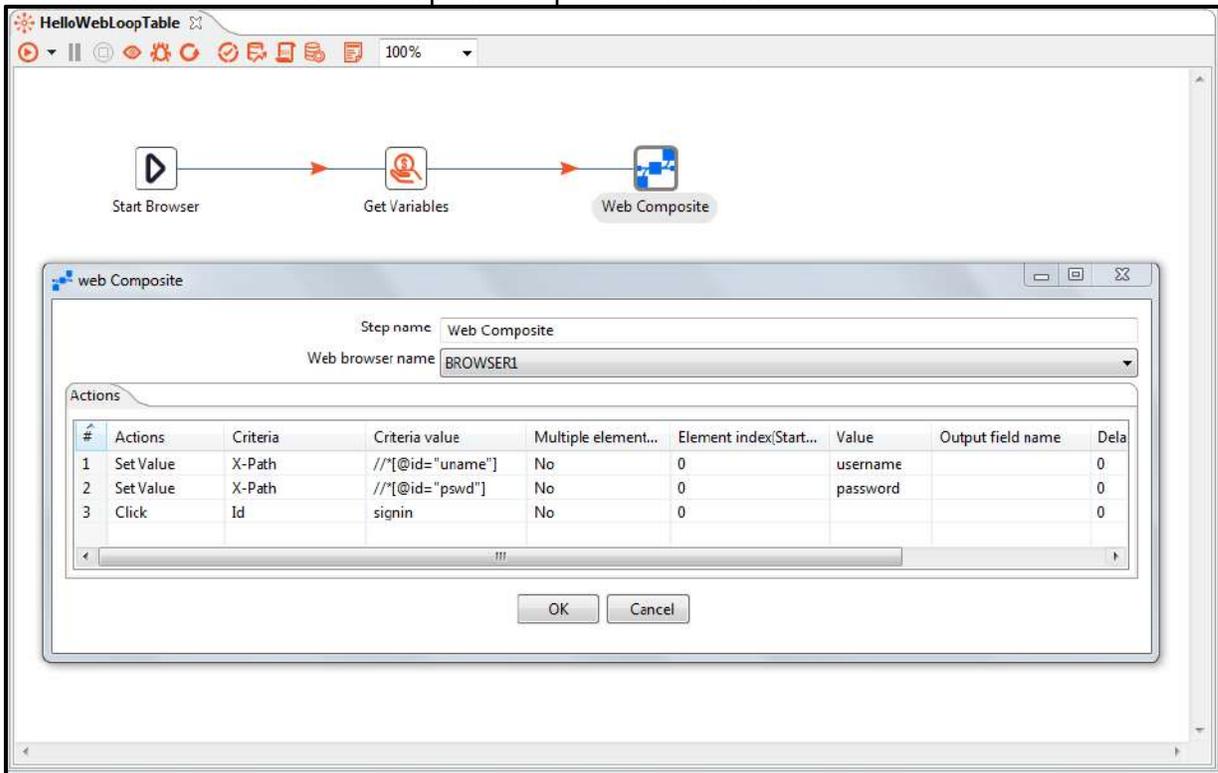
19. Alternately move to the password field using 'Tab' keyboard action with the Robot handling step and then Web Set Value for password.



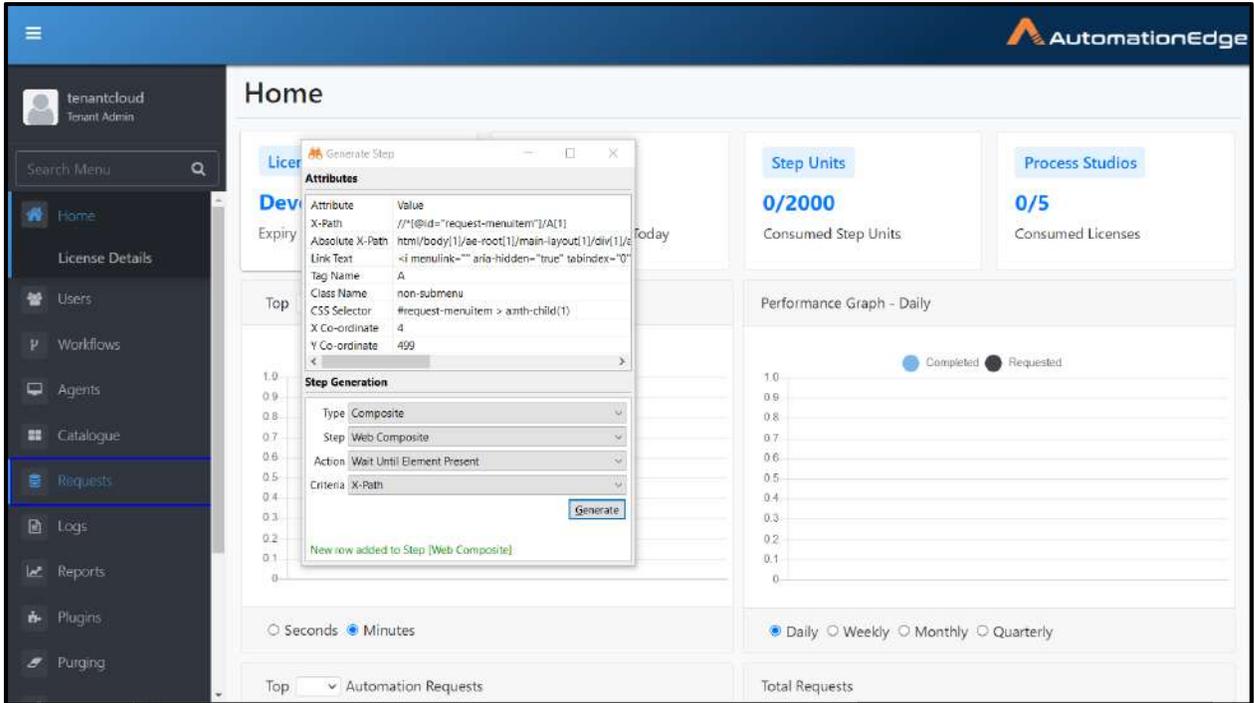
20. Spy the Sign In button. Choose appropriate details as shown and click Generate.
21. A New Row added, message in green text appears at the bottom in the Generate step.



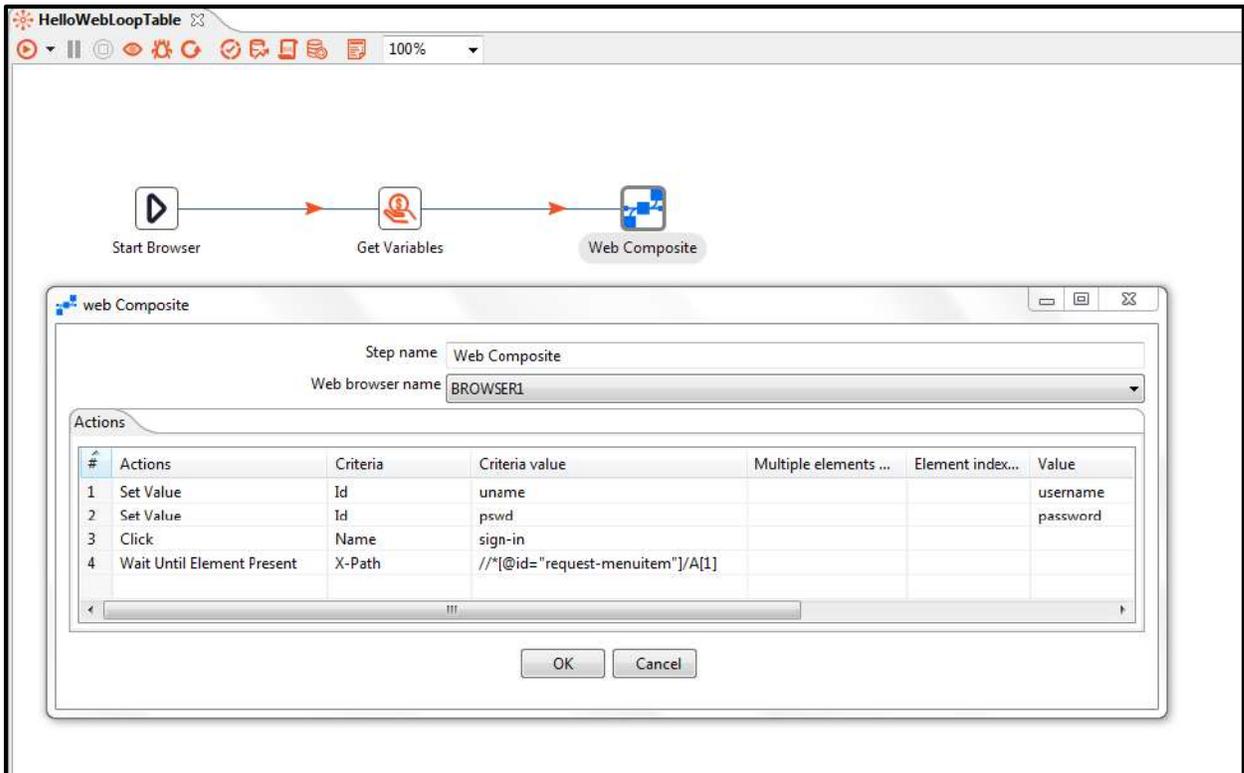
22. A new row is added to Web Composite step as seen below.



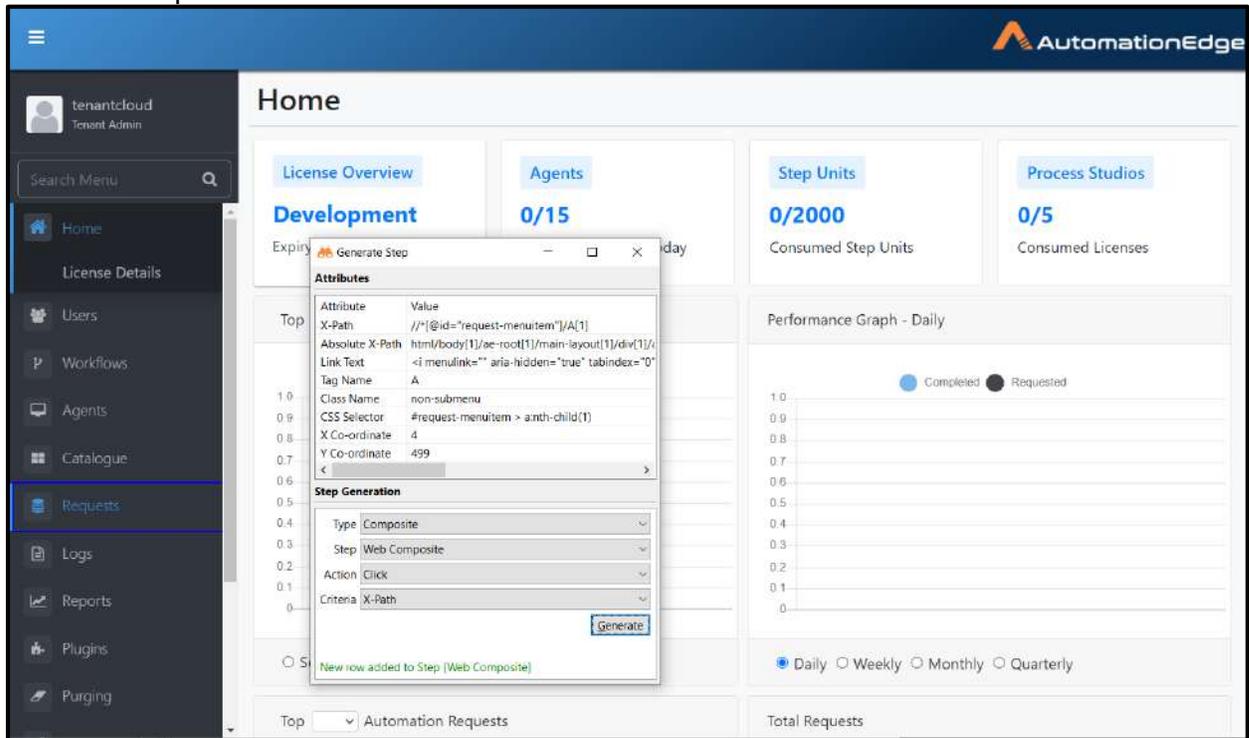
- Get the X-Path for Requests button using GUI Spy to create a Wait Until Element Present step. Configure as shown below and click Generate to add a row to the Web Composite step.



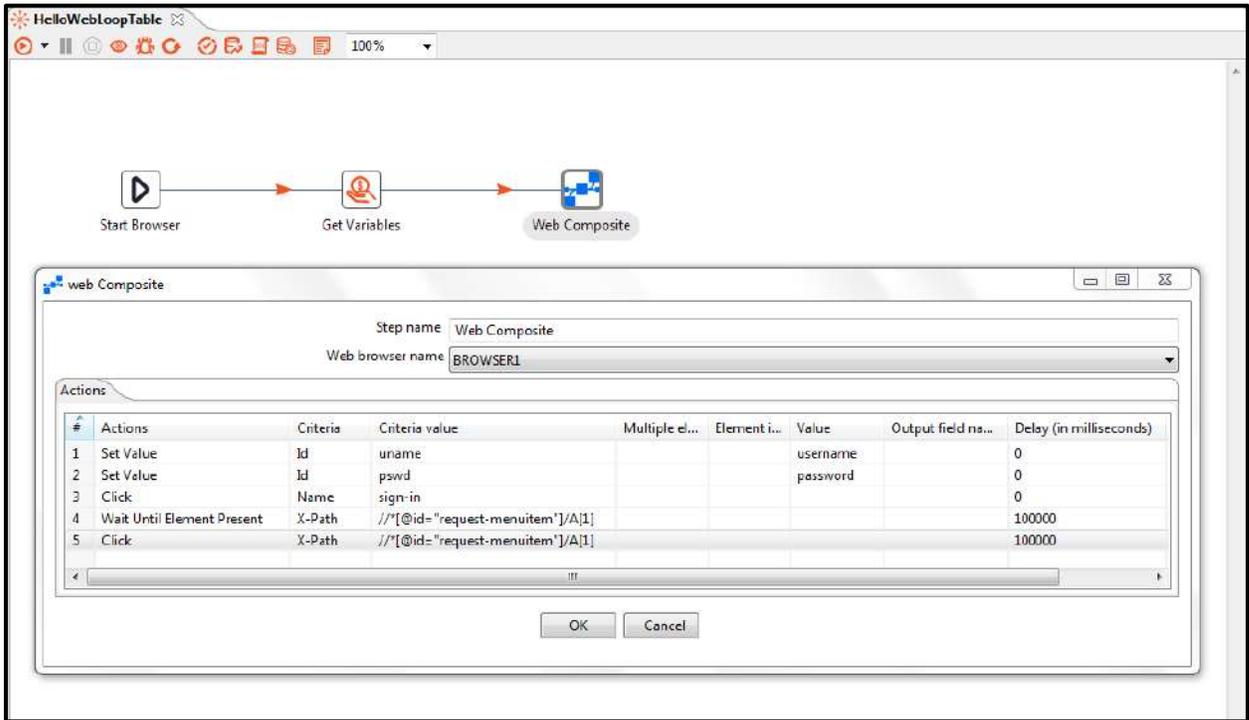
- A Web Composite step is created as seen below.



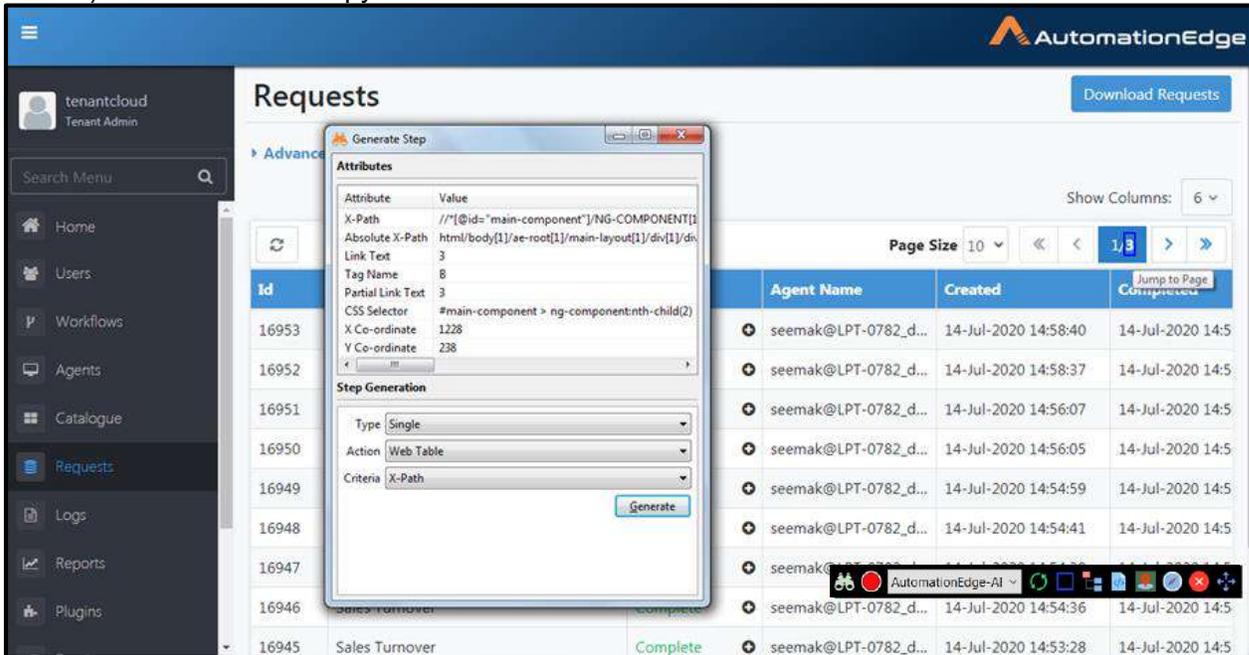
25. Once again Get the X-Path for Requests button using GUI Spy to create a Web Click step. Configure as shown below and click Generate to add a row to the Web Composite step.
26. A new row added to Web Composite step, message in green text appears at the bottom of the Generate step.



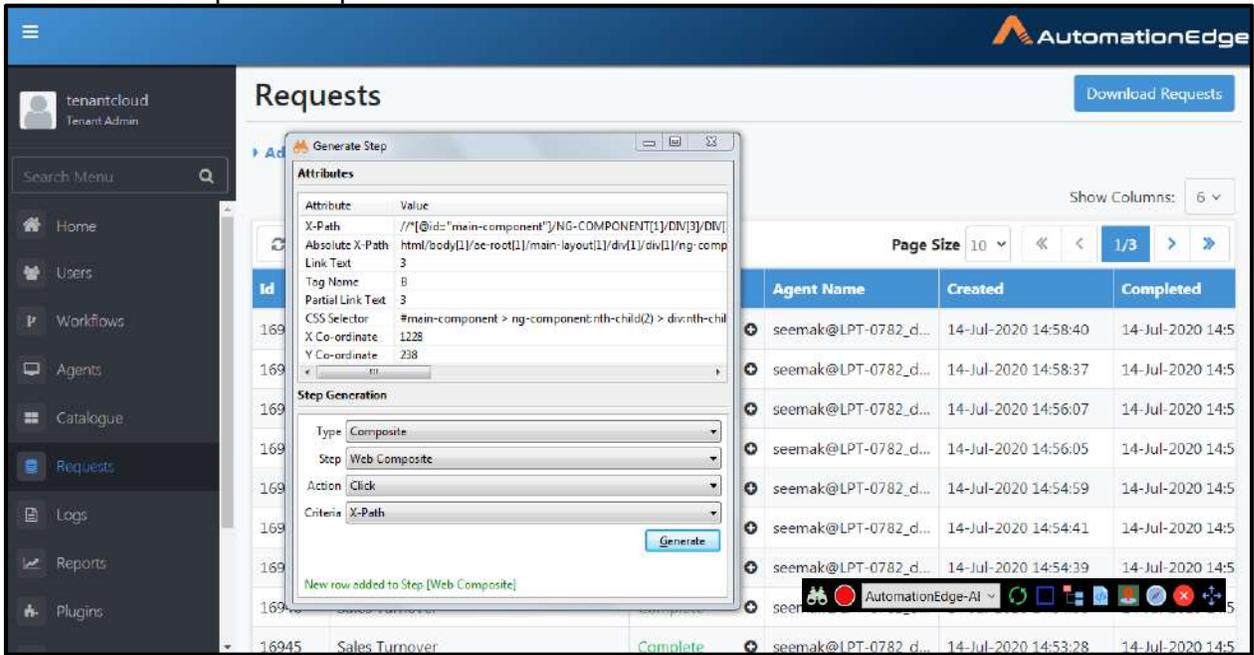
27. A Web Composite step is created as seen below. Add a Delay in milliseconds for both the steps as seen below.



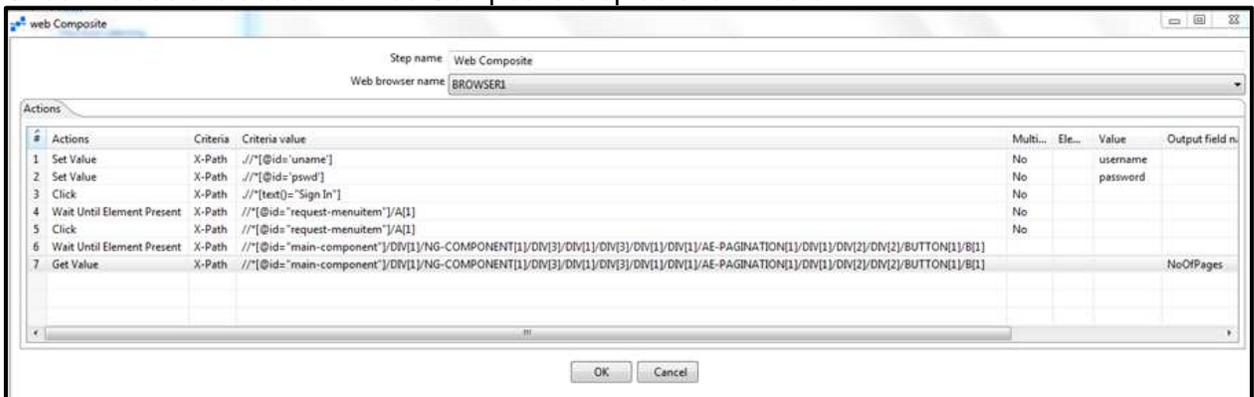
28. Get the X-Path for the total number of pages. Hover over the second part of the label (in this case 2). Double click to copy the X-Path value.



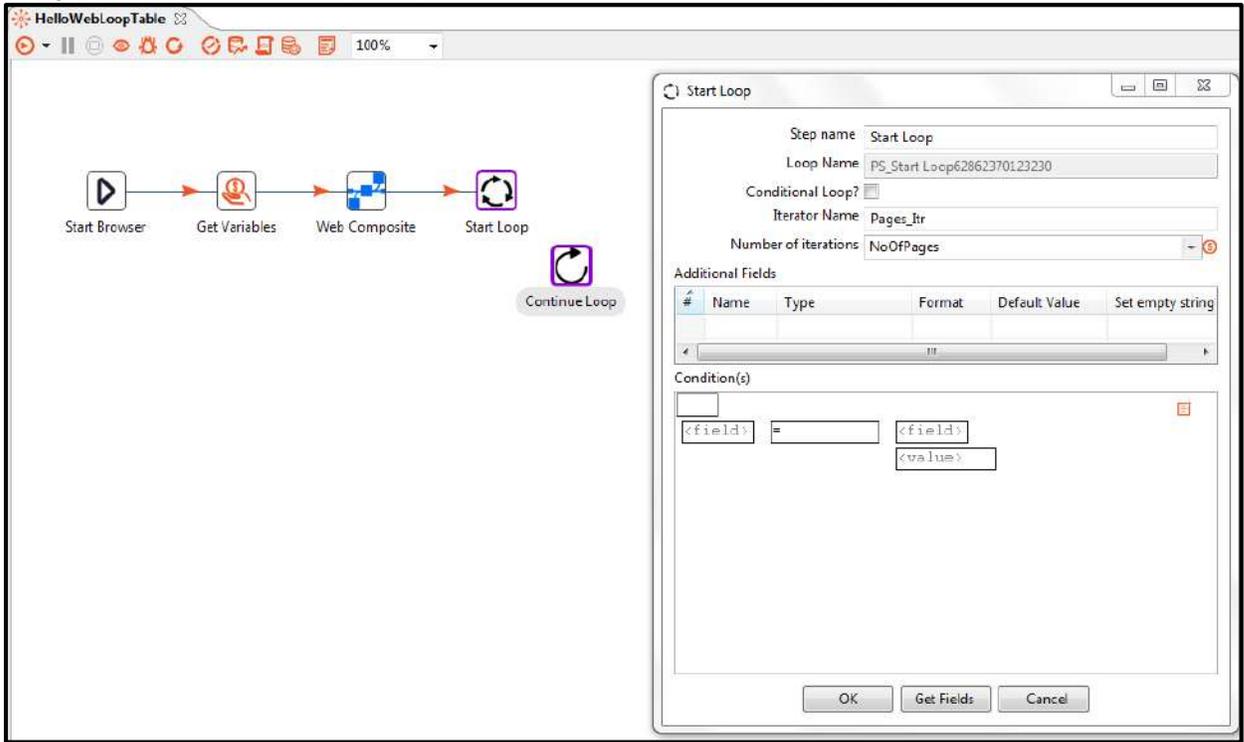
29. A message in green text appears at the bottom of the Generate step indicating a new row added to the Web Composite step.



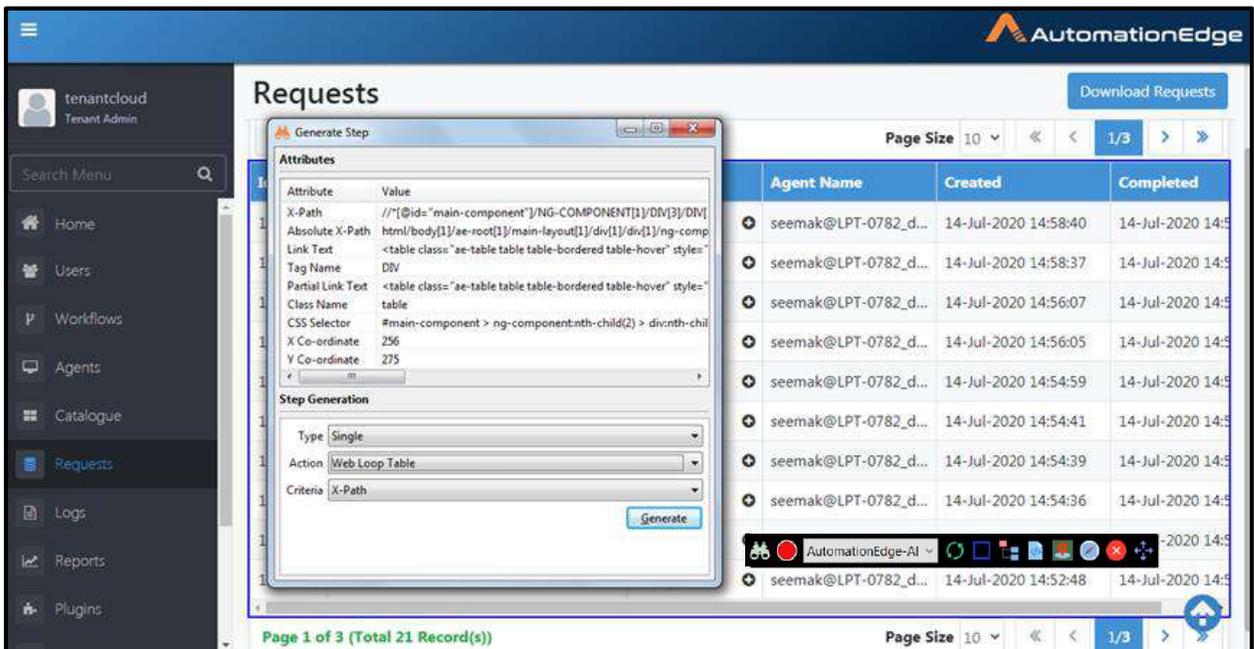
30. Check values are added to Web Composite step as shown below.



31. Drag and Drop a Start Loop Step. It comes with a Continue Loop step.



32. Hover over the bottom scrollbar for the Requests Table and click back tick “”. The web table is highlighted as seen below. Also Attribute[Xpath] value is copied to clipboard.

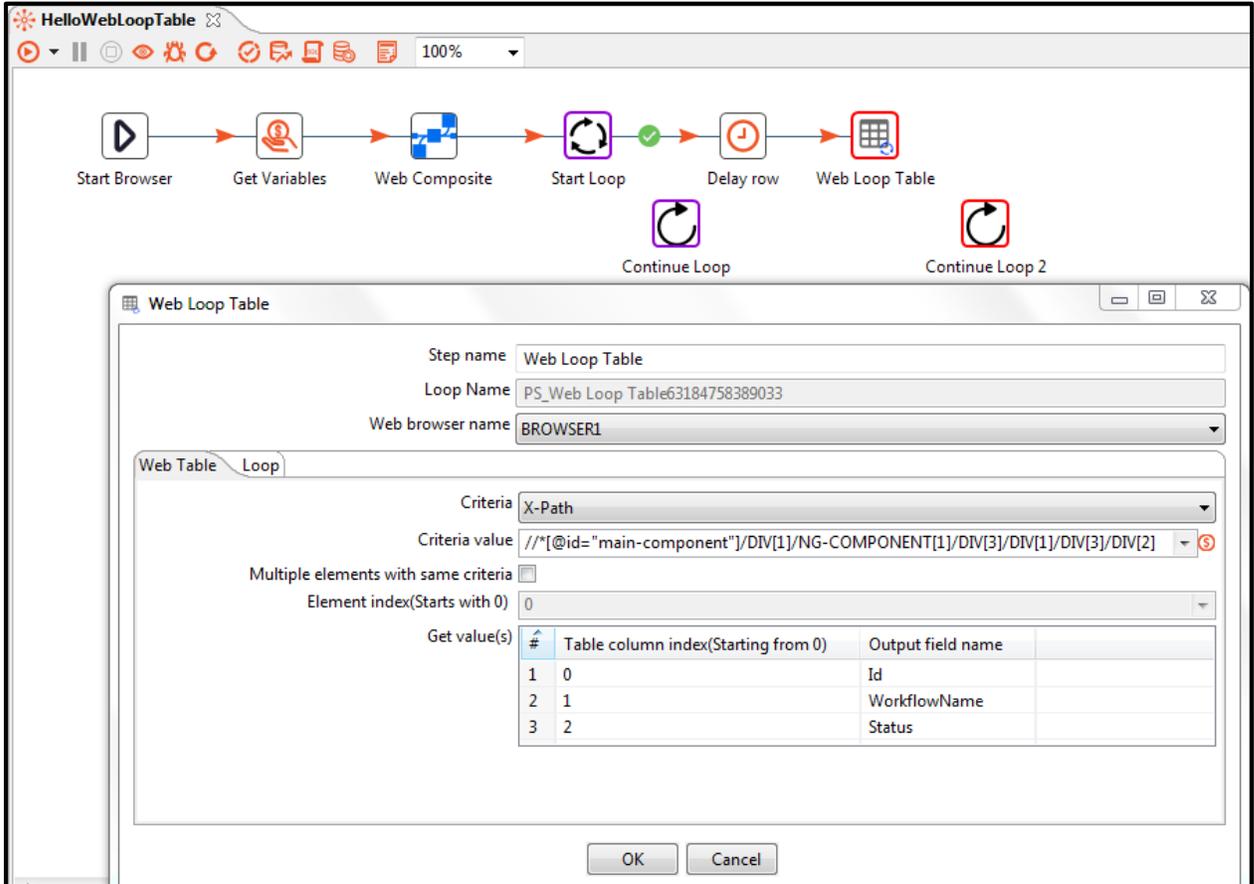


33. Alternately right click on X-Path in Generate step and Copy X-Path Value to clipboard.

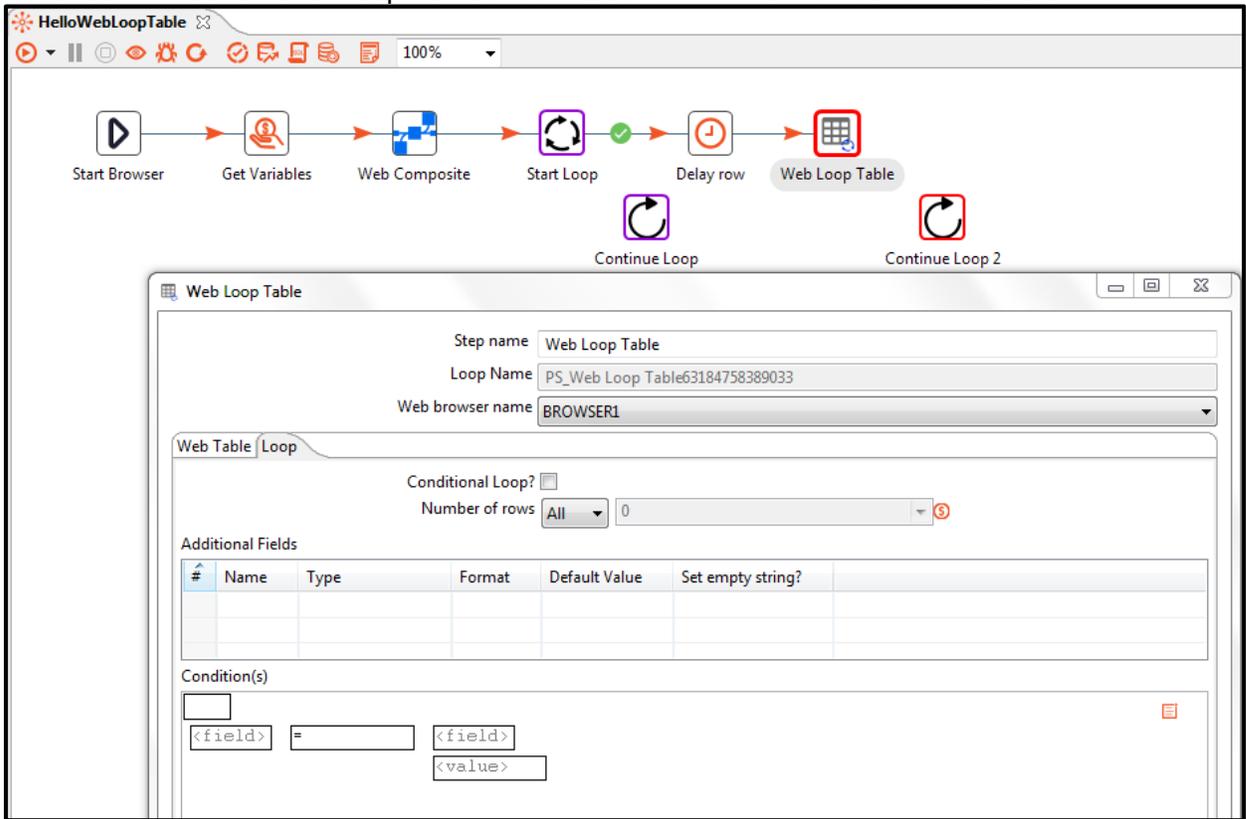
The screenshot displays the AutomationEdge interface. On the left is a navigation menu with options like Home, Users, Workflows, Agents, Catalogue, Requests, Logs, Reports, Plugins, Purging, and Process Studio. The main area is titled 'Requests' and shows a list of request items. A 'Generate Step' dialog box is open, showing the configuration for a step. The 'Attributes' section lists various properties for the X-Path step, such as X-Path, Absolute X-Path, Link Text, Tag Name, Partial Link Text, Class Name, CSS Selector, X Co-ordinate, and Y Co-ordinate. The 'Step Generation' section shows the step type as 'Single', action as 'Web Loop Table', and criteria as 'X-Path'. A 'Generate' button is visible. The background shows a table of requests with columns for Name, Created, and Completed.

Name	Created	Completed
@LPT-0782	14-Jul-2020 14:58:40	14-Jul-2020 14:58:40
@LPT-0782	14-Jul-2020 14:58:37	14-Jul-2020 14:58:37
@LPT-0782	14-Jul-2020 14:56:07	14-Jul-2020 14:56:07
@LPT-0782	14-Jul-2020 14:56:05	14-Jul-2020 14:56:05
@LPT-0782	14-Jul-2020 14:54:59	14-Jul-2020 14:54:59
@LPT-0782	14-Jul-2020 14:54:41	14-Jul-2020 14:54:41
@LPT-0782	14-Jul-2020 14:54:39	14-Jul-2020 14:54:39
@LPT-0782	14-Jul-2020 14:54:36	14-Jul-2020 14:54:36
@LPT-0782	14-Jul-2020 14:53:28	14-Jul-2020 14:53:28

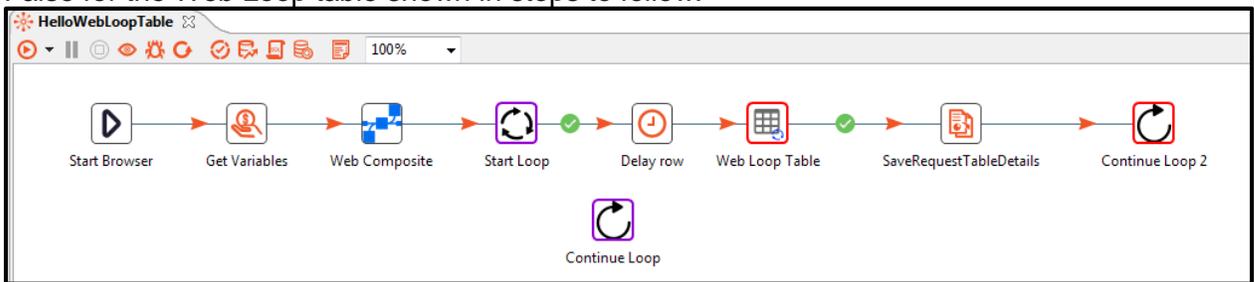
34. Drag and Drop a Web Loop Table step. It comes with a Continue Loop step. Paste the X-Path value in the clipboard to the Criteria value as seen below. Provide Output field names for the first three columns of the Requests Web Loop Table with index 0,1 and 2.
35. Also, Drag and Drop a Delay Row step with 1 second delay between Start Loop and Web Loop Table.



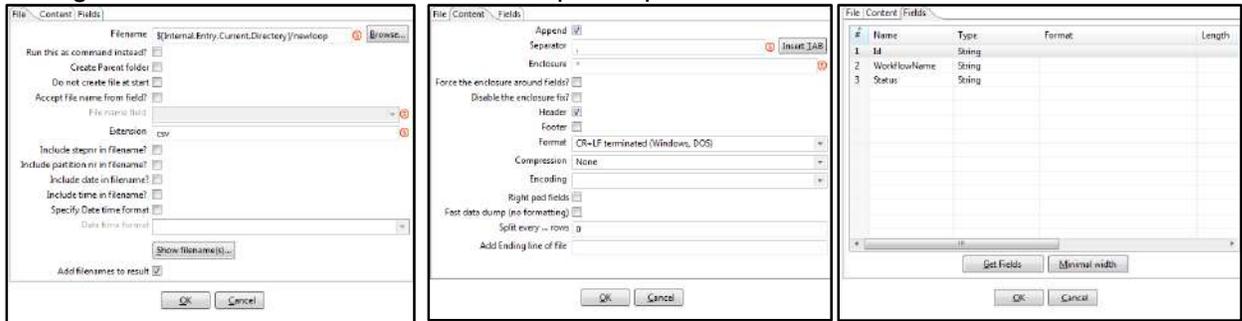
36. Un-check the Conditional Loop? Check box. Select All in the Number of rows field.



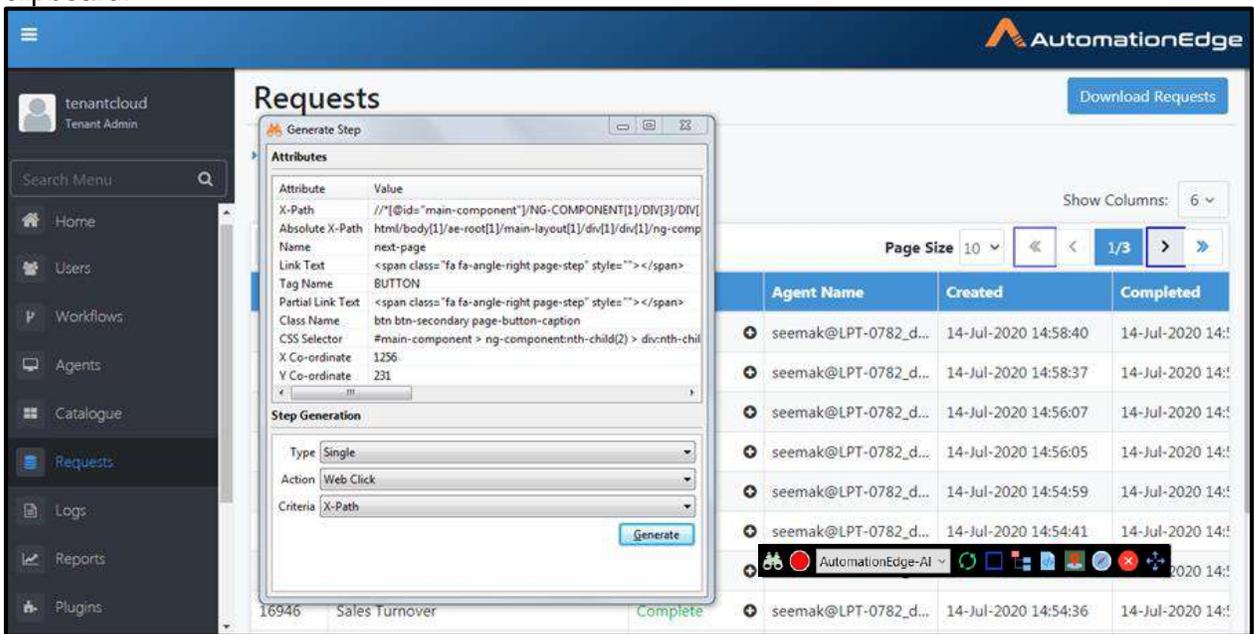
37. Drag and Drop a Text file output step. Join it to Continue Loop2 for Web Loop table. The Web Loop Table Loop reads all the rows of the current Requests page one by one and writes to a text output file in a loop. Once all the rows have been read and written to the output file the loop continues. However, since all the rows have been handled we need to configure a Condition: False for the Web Loop table shown in steps to follow.



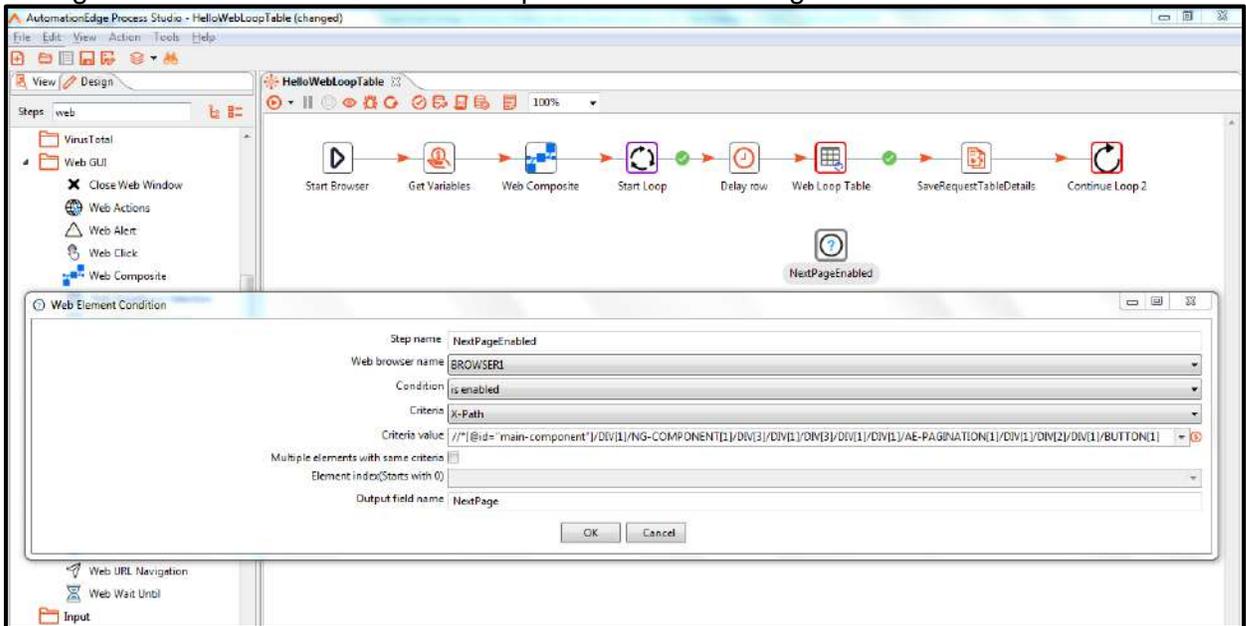
38. Configure the three tabs of the Text file output step as seen below.



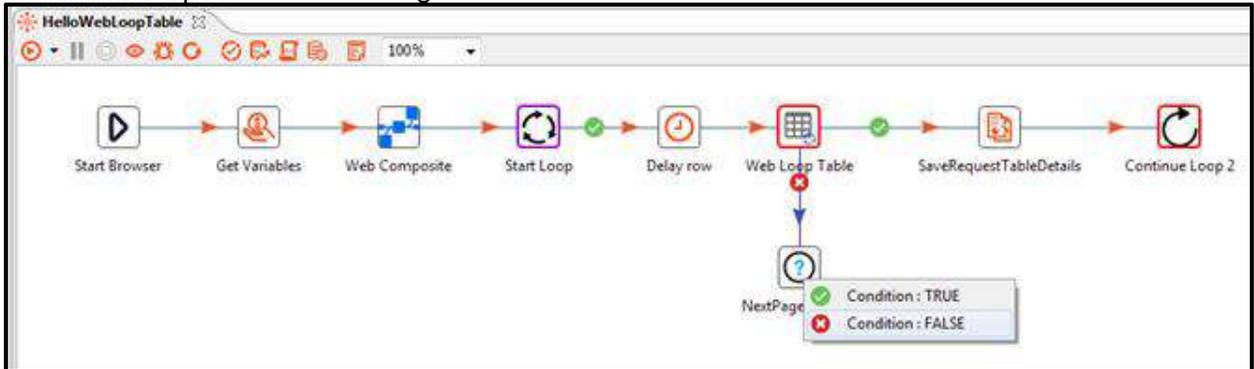
39. GUI Spy the Next button as seen below. Double click or right click and copy X-Path Value to clipboard.



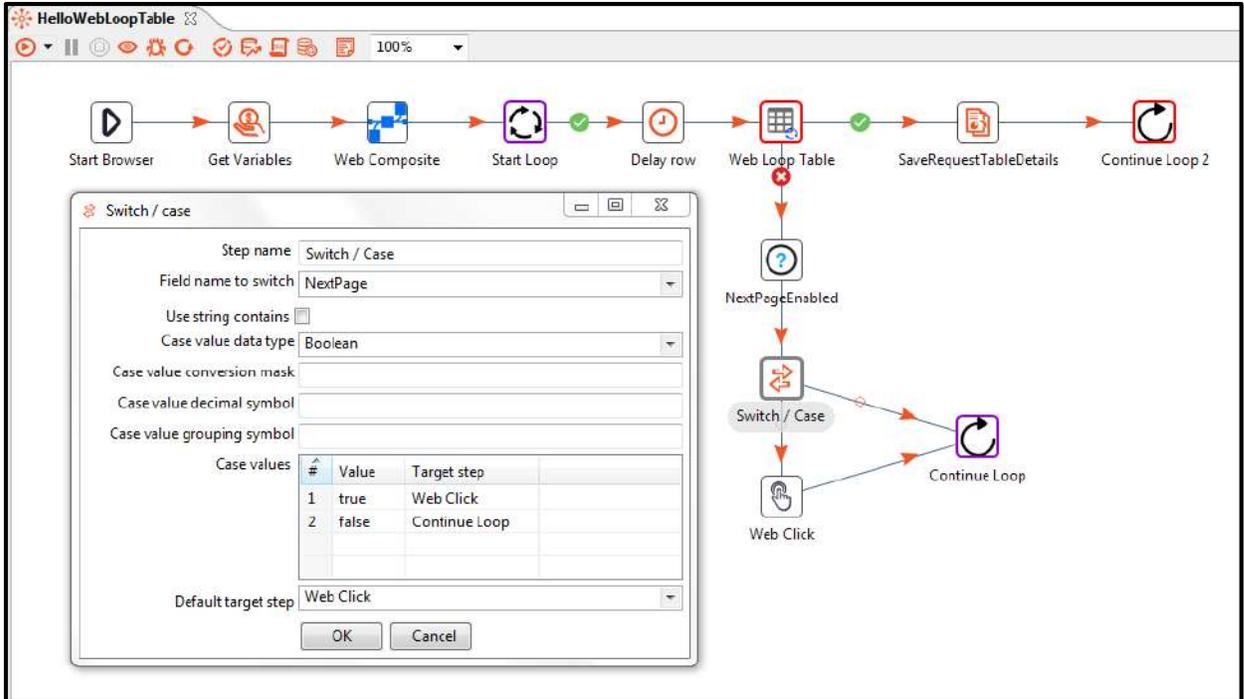
40. Configure a Web Element Condition Step to check if Next Page button is enabled.



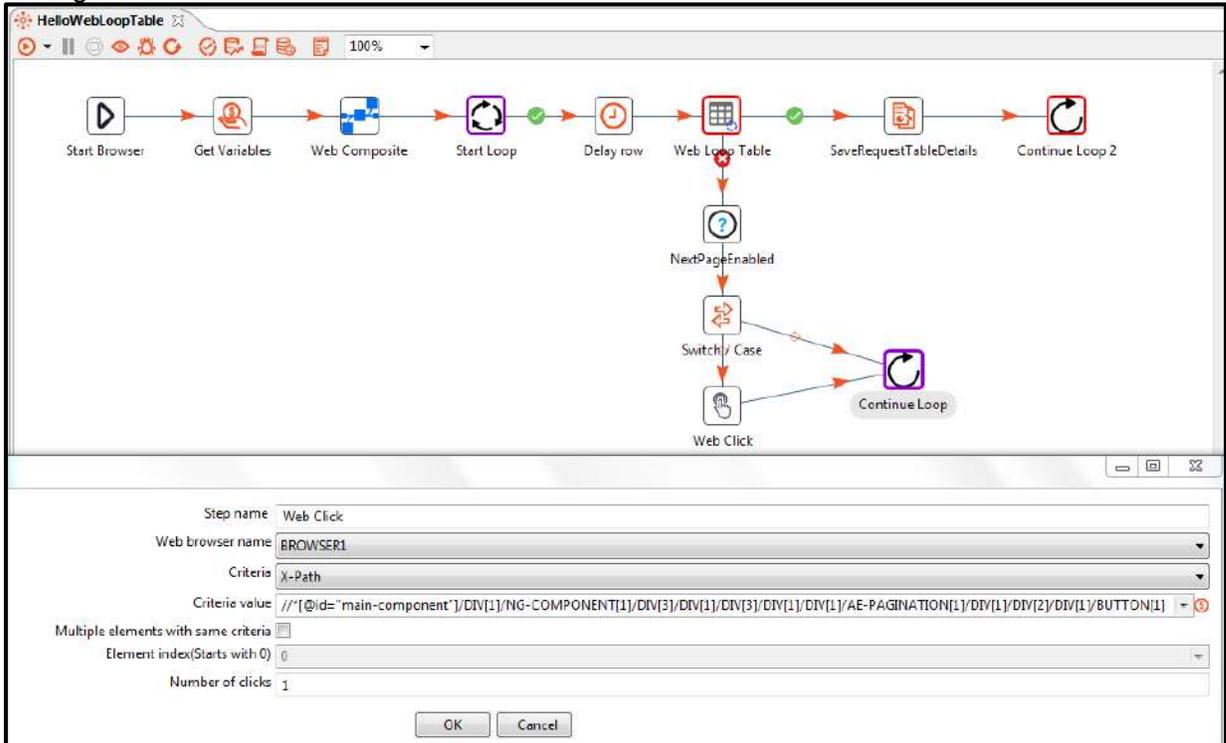
41. Join Web Loop Table to NextPageEnabled with Condition: False.



42. Configure a switch case. Join to Web Click if NextPageEnabled is true. Join Switch case to Continue Loop if NextPage Enabled is false. When all the rows in a particular page are read the loop enters the false condition and clicks the Next Button as shown below. Join Web Click to Continue Loop. The outer loop continues on every click of Next Page button until the number of iterations set in the Start Loop. In our case we have set the number of iterations equal to the number of pages for Requests.



43. Configure Web Click to click Next Button as shown below.



44. The following snapshot depicts a complete workflow executed successfully. It shows the partial view of data stored in the Text file output step in the Preview pane.

The screenshot displays a workflow titled "HelloWebLoopTable" in Process Studio. The workflow consists of the following steps: Start Browser, Get Variables, Web Composite, Start Loop, Delay row, Web Loop Table, SaveRequestTableDetails, and Continue Loop. A branch from Start Loop leads to Exit Browser. From Web Loop Table, a branch leads to NextPageEnabled, which then leads to a Switch/Case step. The Switch/Case step has two paths: one leading to Continue Loop 2 and another leading to Web Click, which then loops back to the Start Loop step.

Below the workflow diagram is the "Execution Results" section, which includes tabs for Logging, Step Metrics, Metrics, Preview data, and Debug. The "Preview data" tab is selected, showing a table of execution results:

#	id	wf	status
1	13283	Get Active Directory Users	Complete
2	13282	Google Stocks	Expired
3	13281	Generate Token	Complete
4	13280	Generate Key	Complete
5	13279	Fetch Deviation Score	Complete
6	13278	Fetch Transactions	Complete
7	13277	Get Google Stock Value	Complete
8	13276	Get Volkswagon Stock Value	Failure
9	13275	Get Wipro Stock Value	Complete

45. This completes the workflow with Web Loop Table.

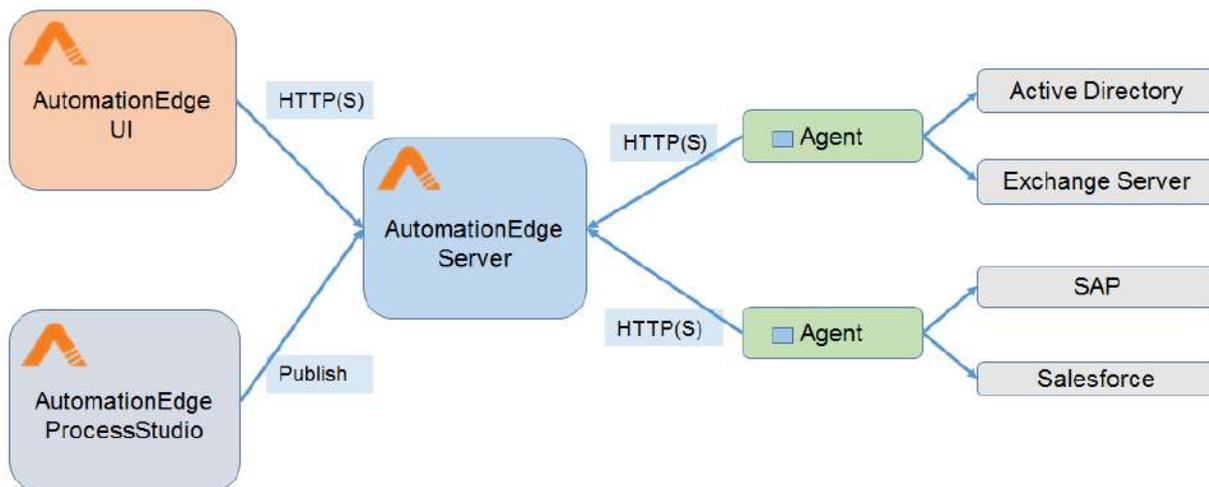
III Publishing Workflows on AE

6 Publishing Workflows on AE Overview

6.1 Introduction

Process Studio is a development environment. Process Studio workflows and processes need to be published to AutomationEdge server eventually, so that they can be executed by the Agents. Workflows can be published on AutomationEdge Servers with any of the licenses type: Trial, Development, UAT and Enterprise or Subscription. Servers with Enterprise and Subscription License are the production servers. Customers and partners generally begin with a Trial License. Eventually they usually go for an Enterprise license or Subscription License. Both Enterprise and Subscription License come with a Development and UAT license as well.

The following figure depicts the architecture of AutomationEdge.



6.2 Parameters considerations

When we run a process or workflow from Process Studio, we see a 'Run Options' dialog. Here we can set the values of both runtime and design parameters. However, there is no difference between runtime and design parameters when running workflows or processes from Process Studio. Significance of design time and run time parameters becomes evident after you publish a workflow or a process to AutomationEdge server.

We can execute a process or workflow after publishing it to AutomationEdge from a GUI based catalog. The design time parameters are not visible at runtime. However, in AutomationEdge the value of design time parameters can be set in workflow details under the section 'Configuration Parameters'. On the other hand, we are prompted to provide values of runtime parameters when we execute the workflow from AutomationEdge server.

The execution details can be viewed on the Requests tab. In AutomationEdge Server environment a target file directory is created storing workflow files, supporting files and any input/output files. Any output files that are generated by the workflow can be conveniently downloaded via a link that is available with the execution details of that request.

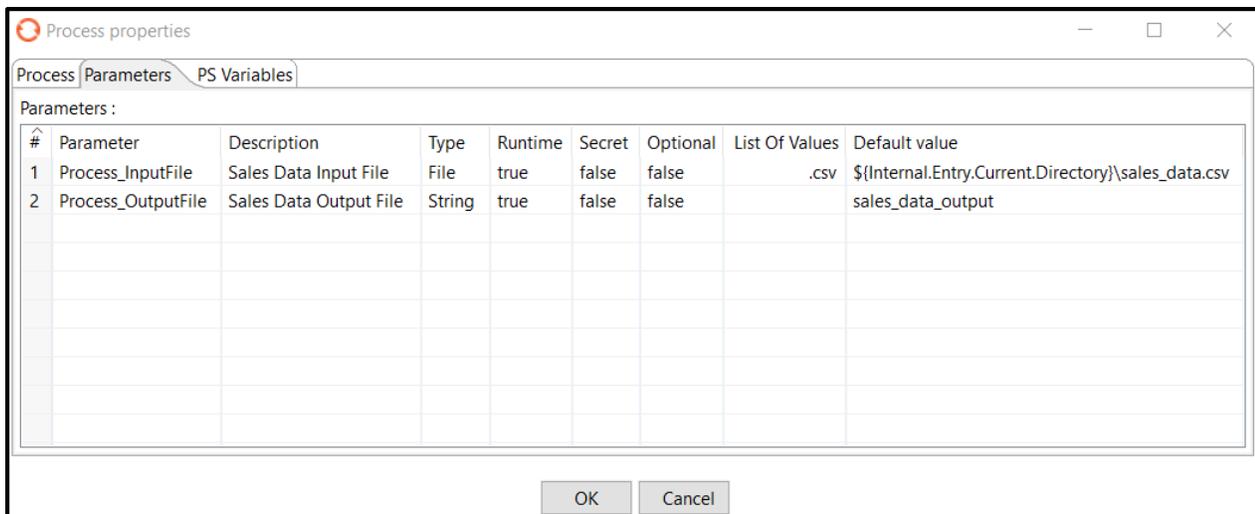
Note:

Process Studio allows an absolute path for input and output files identified by process parameters. However, these paths are specific to the machine where Process Studio is used. These workflows may be published by many individual tenants on various instances of AutomationEdge. Hence the absolute path is no longer relevant. We need to modify these using internal parameter `'${Internal.Entry.Current.Directory}'`. `'${Internal.Entry.Current.Directory}'` refers the 'File' directory on AutomationEdge server.

For example set input file parameter relative to `'${Project.Directory}'` or `'${Internal.Entry.Current.Directory}'` (e.g. `'${Internal.Entry.Current.Directory}\sales_data.csv'`). Set it as a runtime parameter so that user has an option to choose the input file.

For example set output file parameter as `'${Internal.Entry.Current.Directory}\sales_data_output'`. We shall set runtime as true as we want the end users to provide a file output name of their choice. The Output file type is String and not file because by default the output files are stored on AutomationEdge server. Output files can be easily downloaded from the link provided after execution on the Requests page. End user does not have to bother about the AutomationEdge file directory.

Please see figure below.



6.3 Publishing Workflows: Flow

6.3.1 Publishing Workflows: Flow for publishing to Trial instance

New workflows may be published on Trial instance in one of the follows:

1. Use Process Studio Publish option to directly publish new workflows to AutomationEdge or
2. Use Process Studio Export option to update workflows on AutomationEdge Trial Instance.

6.3.2 Publishing Workflows: Flow for publishing to Production instance

The flow for deploying workflows in production is as follows:

1. Publish to Development instance
 - Use Process Studio Publish option to directly publish new workflows to AutomationEdge or
 - Use Process Studio Export option to publish to update workflows on AutomationEdgeTrial Instance.
2. Import to Development Instance
3. Export from Development Instance
4. Import to UAT Instance
5. Export from UAT instance
6. Import to Production (Enterprise or Subscription) instance

7 Publish HelloWeb to AutomationEdge

7.1 Publish HelloWeb to Development Instance

Process Studio Processes/Workflows can be published using,

- Process Studio Publish option for Create or Update
- Export from an AutomationEdge instance followed by import to another AutomationEdge instance

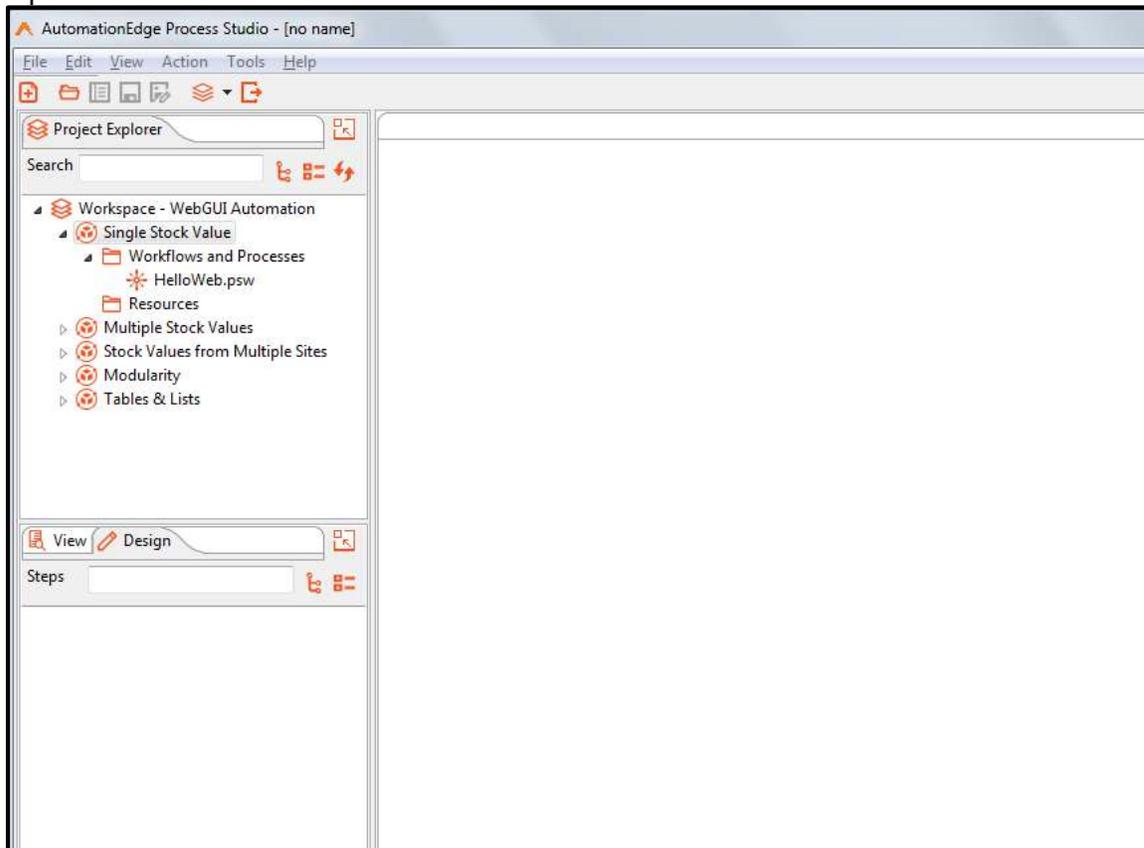
7.1.1 Workflows: Publish Project from Process Studio

In this section we will discuss Process Studio Publish→ Create and Update options to publish AutomationEdge workflow on Development instance.

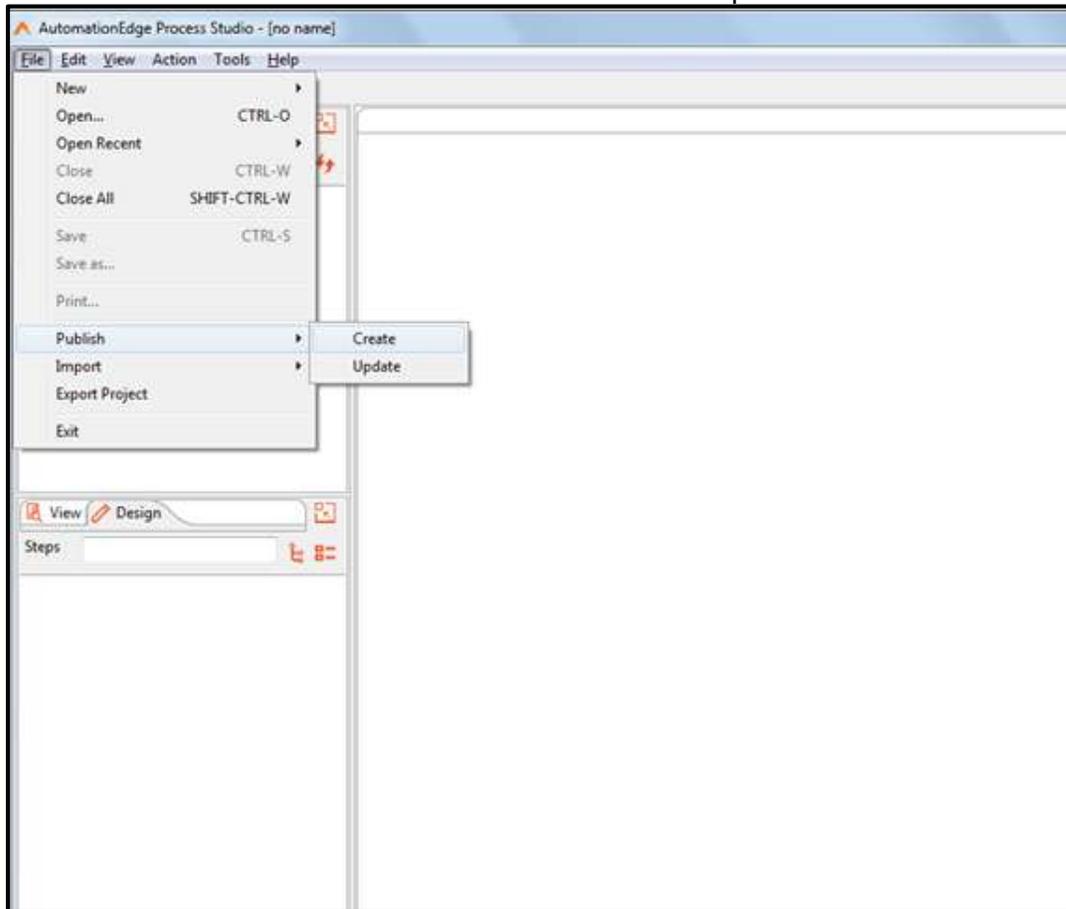
7.1.1.1 Publish: Create

In this exercise we wish to publish HelloWeb workflow for the first time using Process Studio Publish→Create option. Following are the steps to publish an AutomationEdge workflow using Process Studio Publish option,

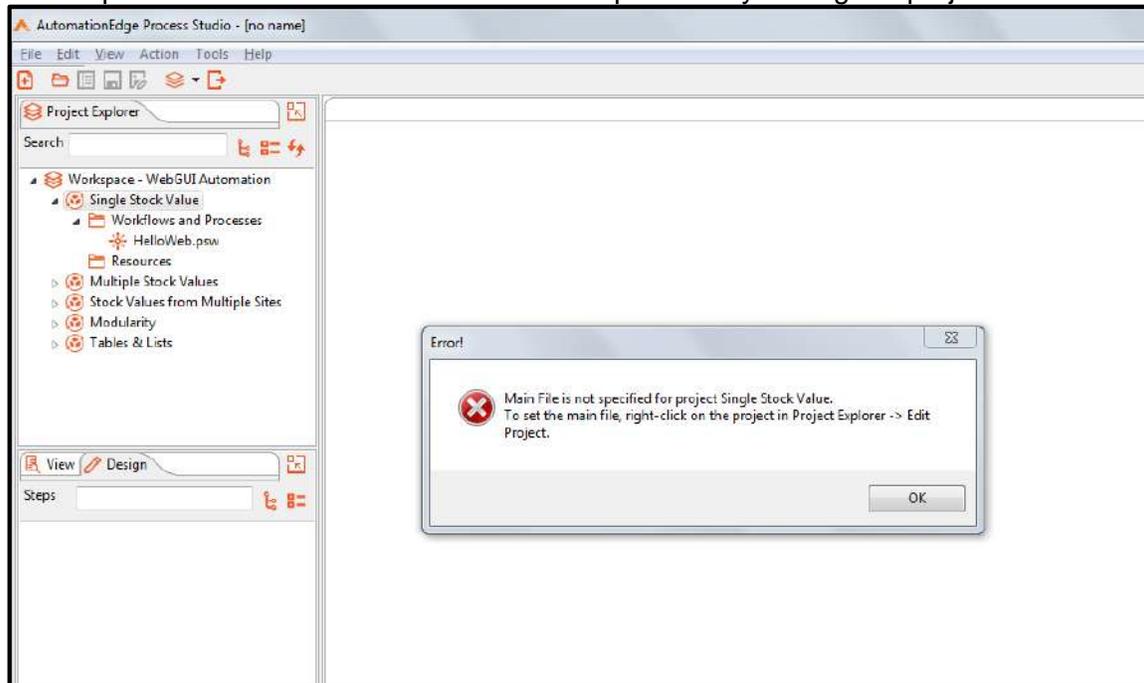
1. Open HelloWeb workflow in Process Studio as seen below.



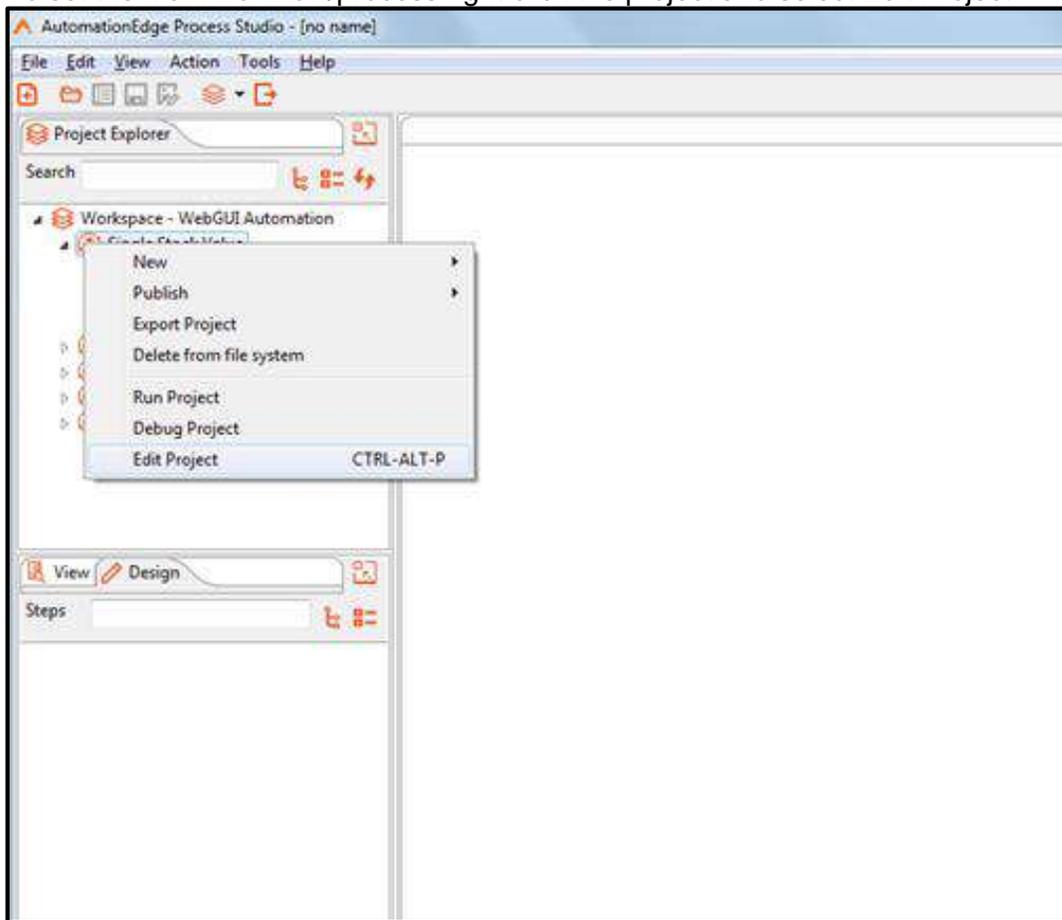
2. Select File menu and click Publish and then on Create option.



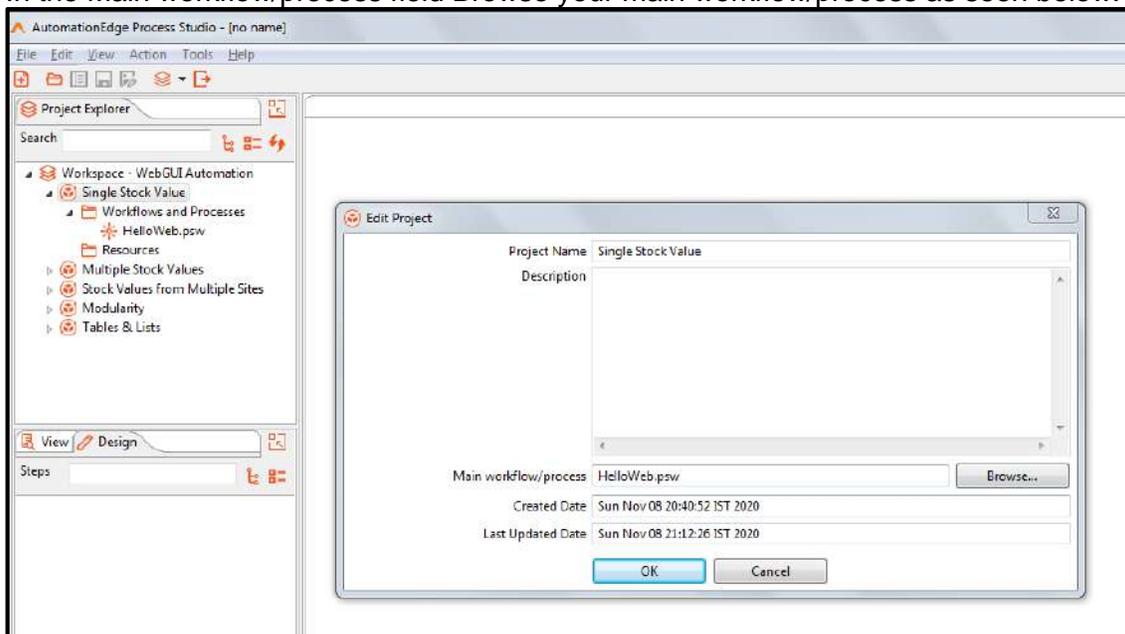
3. However, we receive an error message popup as seen below. Main File is not selected for the Project.
4. When you publish a project you need to set a main file by editing the project. This especially seems obvious when our project has multiple processes/workflows; where you need to set the Parent process/workflow as the main workflow/process by editing the project.



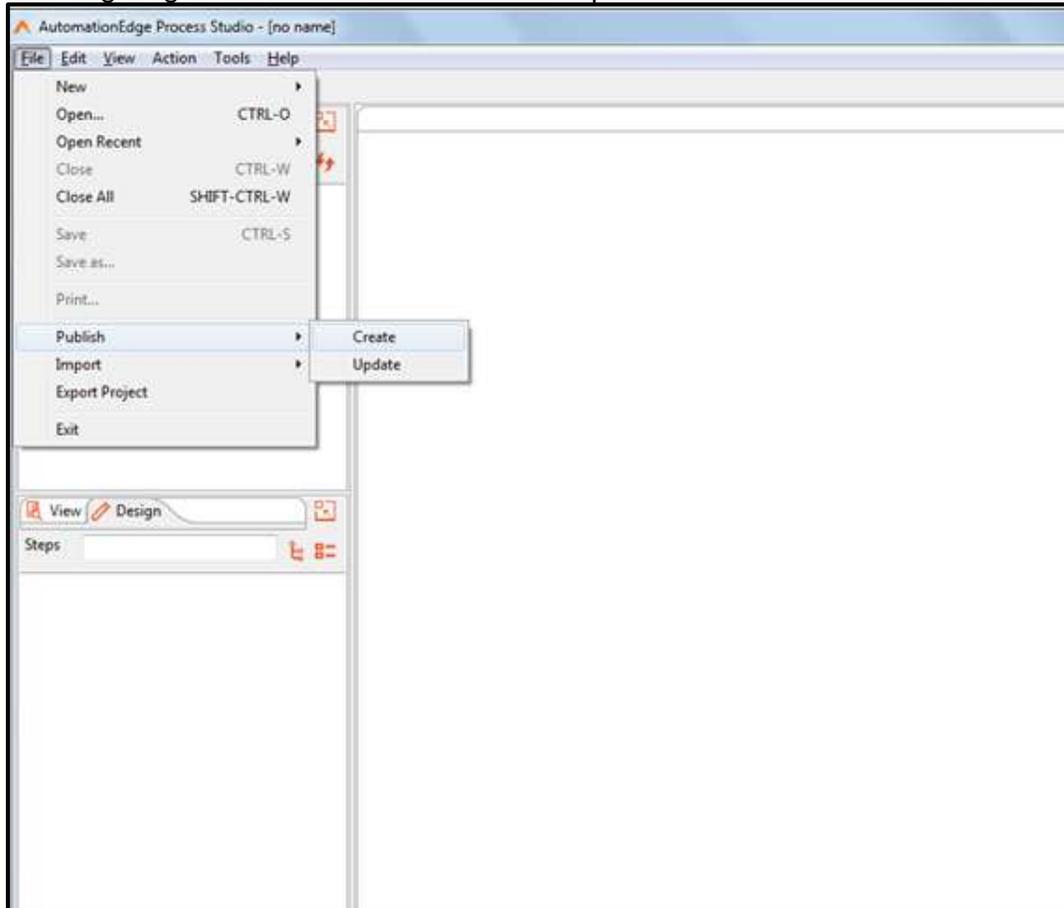
- To set the Main workflow/process right click the project and select Edit Project.



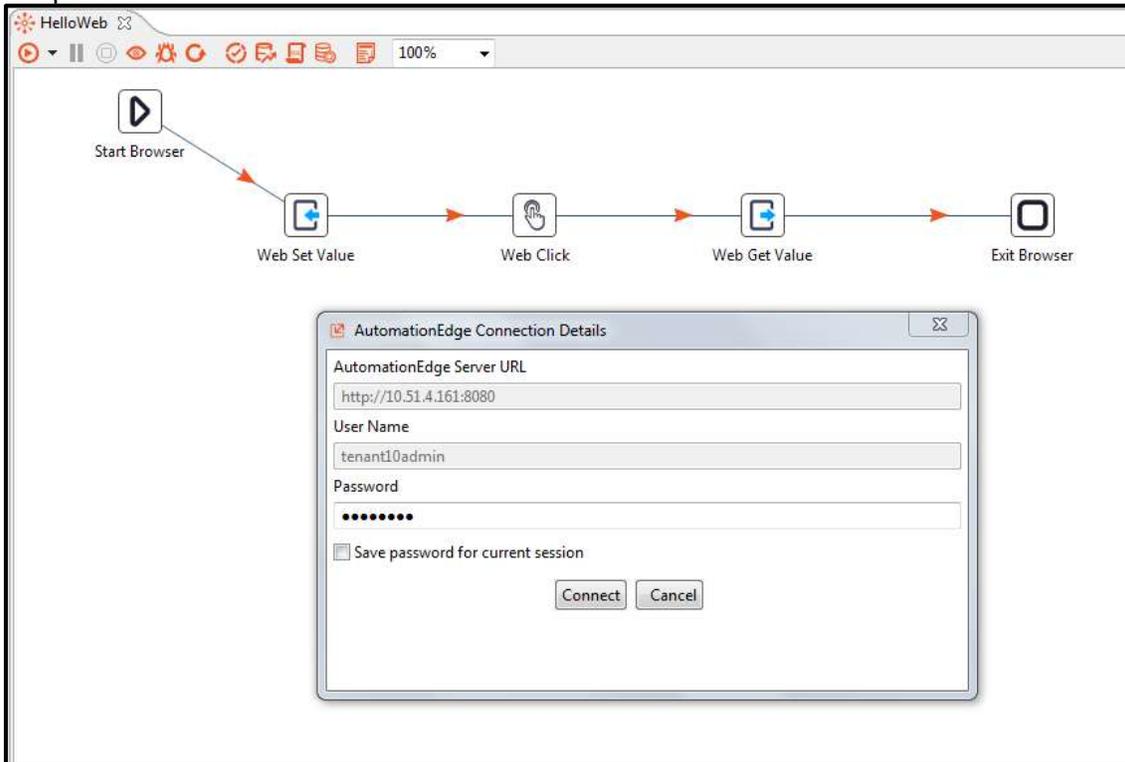
- In the Main workflow/process field Browse your main workflow/process as seen below.



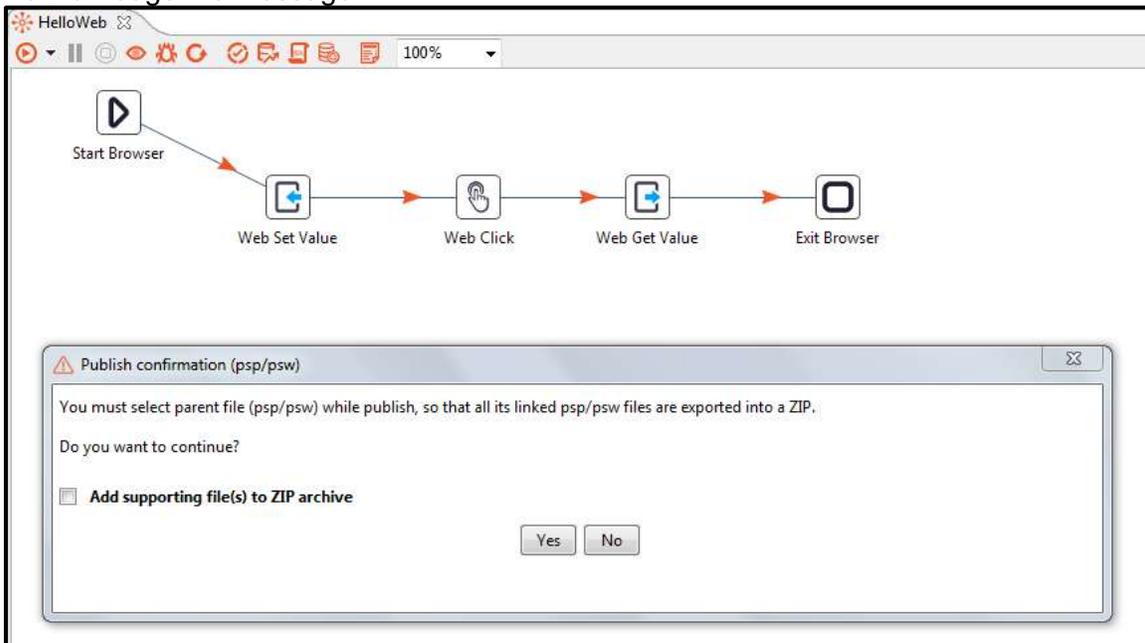
7. In case you have a Process and multiple workflows then open the parent Process or parent Workflow and go to the respective tab.
8. Once again go back to the Publish → Create option under File menu.



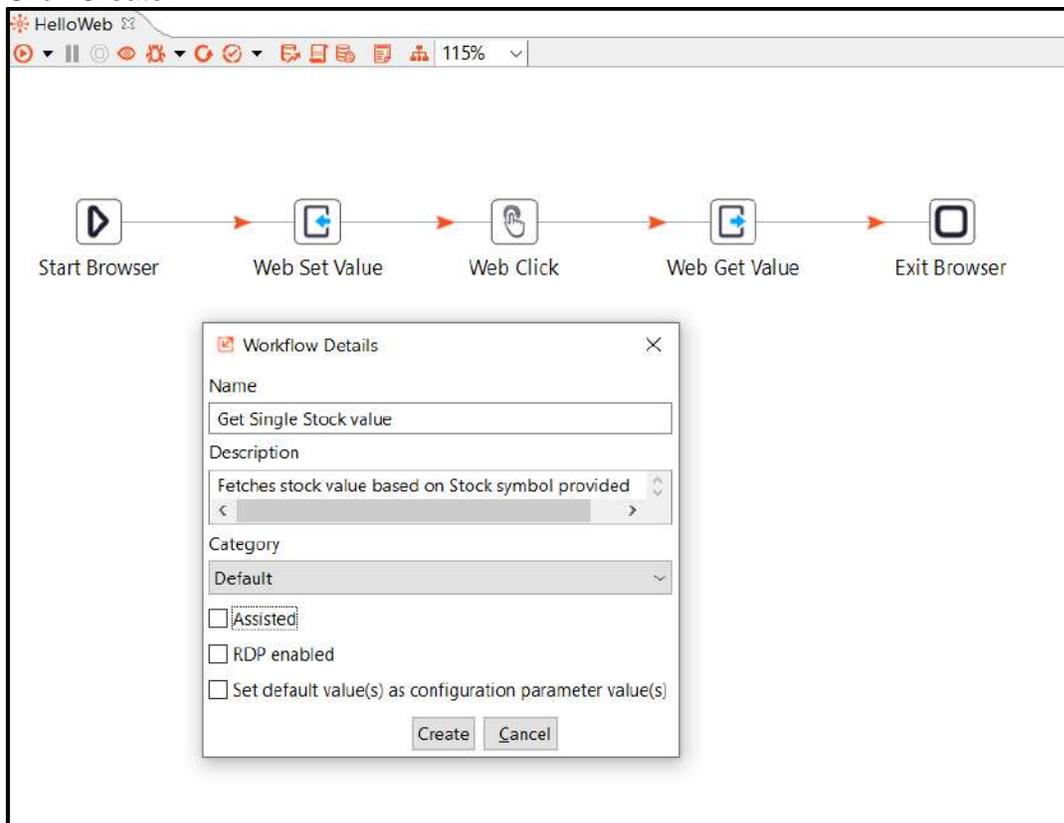
- If you have not saved the password AutomationEdge connection Details dialog appears. Provide the password.



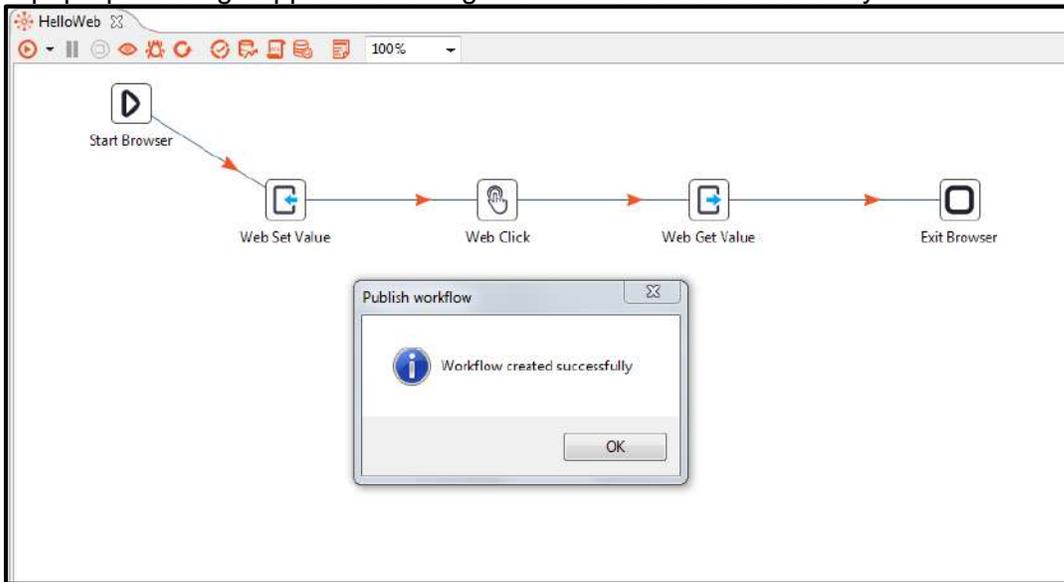
- A pop-up message appears warning that you must select a parent process(psp) or workflow(psw) as the case may be so that all the linked psp/psw files are exported as a zip. Acknowledge the message.



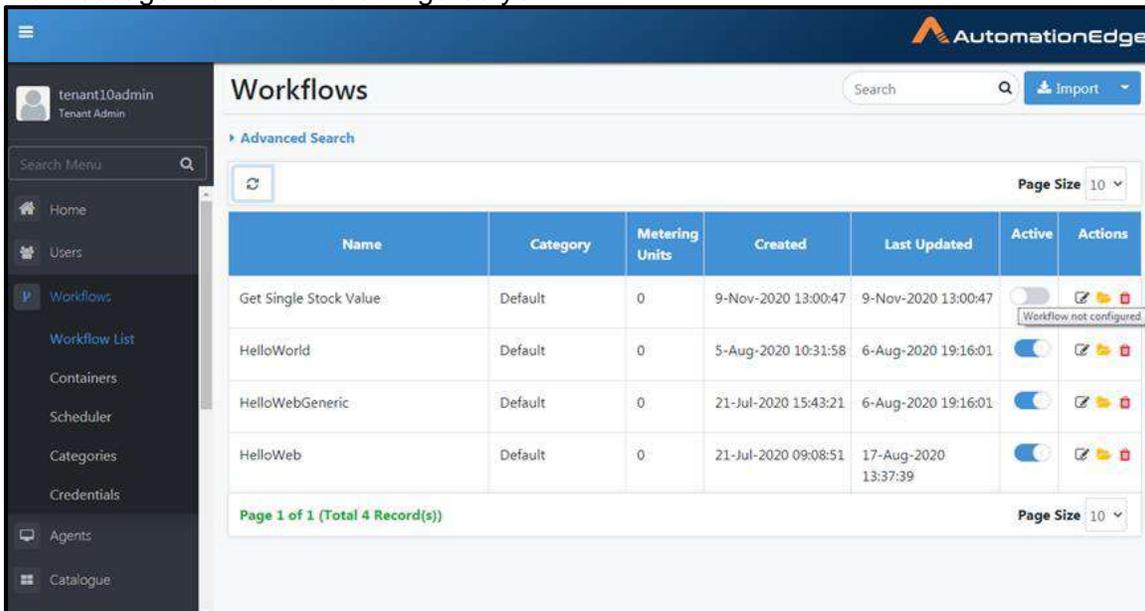
11. In case you check 'Add supporting files to zip archive' a corresponding popup appears. In this case we do not have any supporting .csv, .txt, xlsx or other files. Leave 'Add supporting files to zip archive' unchecked and click Yes.
12. A Workflow Details pop up appears with details as seen below.
 - Enable Assisted for Attended workflows
 - It is internally detected if the workflow is sequential; after publishing it is reflected on AutomationEdge UI (Else if one of the GUI automation workflows running as non-sequential workflow is terminated, then all the subsequent requests remain in New state). This workflow will not be detected as a sequential workflow as it does not have any of the following steps – Robot Handling step, Capture Screenshot step, Desktop steps, GUI Automation Windows steps, Surface Automation steps, AS400, Auto IT which decide if a workflow is sequential.
 - Enable RDP enabled if working on Controller Agent machine.
 - If you wish to Set default values as Configuration parameter values enable check box.
13. Click Create.



- A pop-up message appears showing Workflow created successfully.



- This completes the publish process of a process or workflow from Process Studio.
- Now logon to AutomationEdge instance on which you created the workflow. In the Workflow List menu, you can see the newly published Get Single Stock Value workflow.
- At this point you cannot yet activate the workflow. Hover over the Activate toggle switch to see the message Workflow not configured yet.



18. Click the edit icon in the Actions column to complete workflow configuration. As seen below most of the basic details were already provided during publish in Process Studio.
19. Since this workflow was not detected as sequential internally, the Enable Sequential Automation Checkbox is unchecked. You can check the Enable Sequential Execution if you desire to make it sequential, although it is not mandatory.
20. Additional configure Expected Completion Time, Maximum Completion Time, Cleanup Requests older than and Manual Execution time.

Configure Workflow Details

▼ Basic Details

Workflow Name: **Get Single Stock Value**

Workflow Description (Maximum 128 Characters): *

Fetches stock value based on Stock symbols provided

Workflow Category:

Default

Workflow Icon: 

Assisted Workflow : **false**

Enable Sequential Execution

Enable RDP

Enable Input Attributes

Workflow Priority:

Default

Expected Completion Time(Seconds): *

5

Maximum Completion Time(Seconds): *

20

Cleanup Requests older than(Hours):

36

Manual Execution Time:

5 Minutes

▶ Email Notification Setting

No Configuration Parameters



21. Configure Email Notification Setting as seen below.
22. Click Save.

▼ Email Notification Setting

Notify On Workflow Failure

Notify On Exceeding Time Limit

Select Users*:

By Role: Tenant Admin Workflow Admin

By Username:

By Email:

Request Creator

Failure Message:

No Configuration Parameters



23. Workflow updated successfully message appears.

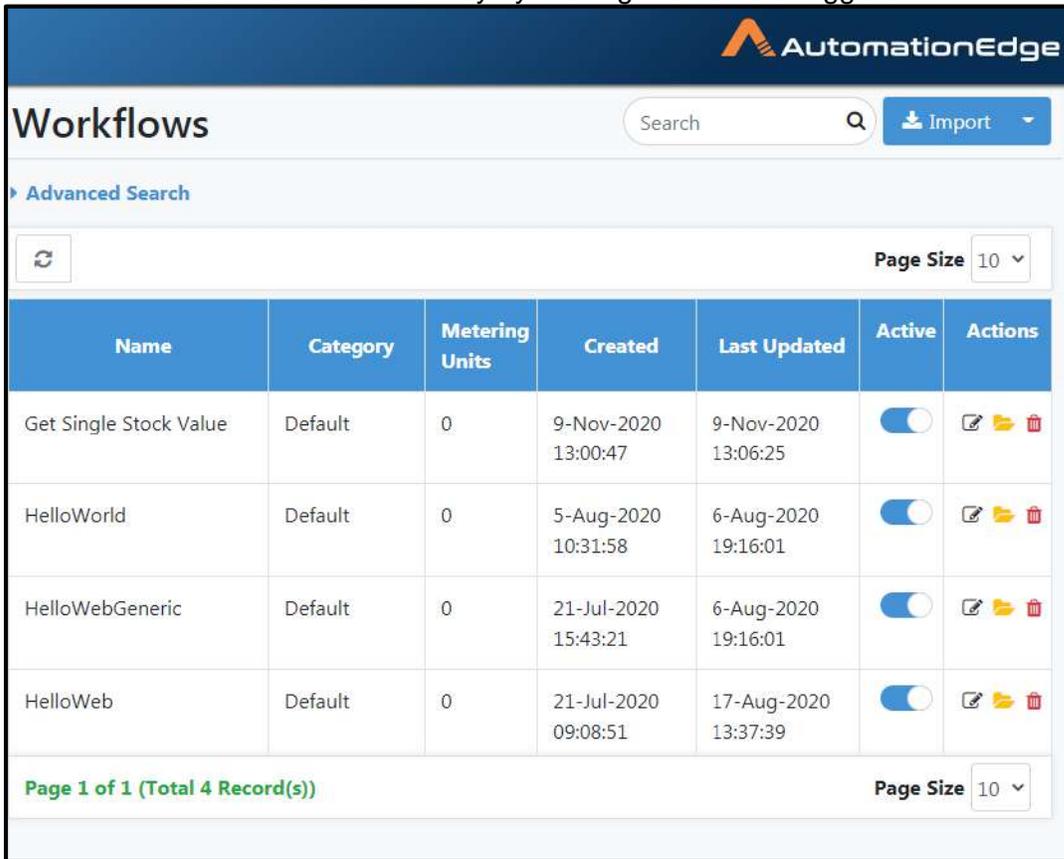
The screenshot shows the AutomationEdge Workflows interface. At the top, there is a search bar and an 'Import' button. Below the search bar is an 'Advanced Search' link. The main content area features a table with the following columns: Name, Category, Metering Units, Created, Last Updated, Active, and Actions. The table contains four rows of workflow data. At the bottom of the table, there is a pagination bar showing 'Page 1 of 1 (Total 4 Record(s))' and a 'Page Size' dropdown set to 10. A green success message box is displayed at the bottom right of the interface, containing a checkmark icon, the word 'Success', and the text 'Workflow updated successfully'.

Name	Category	Metering Units	Created	Last Updated	Active	Actions
Get Single Stock Value	Default	0	9-Nov-2020 13:00:47	9-Nov-2020 13:06:25	<input type="checkbox"/>	
HelloWorld	Default	0	5-Aug-2020 10:31:58	6-Aug-2020 19:16:01	<input checked="" type="checkbox"/>	
HelloWebGeneric	Default	0	21-Jul-2020 15:43:21	6-Aug-2020 19:16:01	<input checked="" type="checkbox"/>	
HelloWeb	Default	0	21-Jul-2020 09:08:51	17-Aug-2020 13:37:39	<input checked="" type="checkbox"/>	

Page 1 of 1 (Total 4 Record(s)) Page Size 10

Success
Workflow updated successfully

24. You can now activate the workflow by clicking the activate toggle switch.



The screenshot shows the AutomationEdge Workflows management interface. At the top, there is a search bar and an 'Import' button. Below the search bar is an 'Advanced Search' section with a refresh icon and a 'Page Size' dropdown set to 10. The main content is a table with the following columns: Name, Category, Metering Units, Created, Last Updated, Active, and Actions. The 'Active' column contains toggle switches, and the 'Get Single Stock Value' workflow has its toggle switch turned on. The bottom of the table shows 'Page 1 of 1 (Total 4 Record(s))' and another 'Page Size' dropdown set to 10.

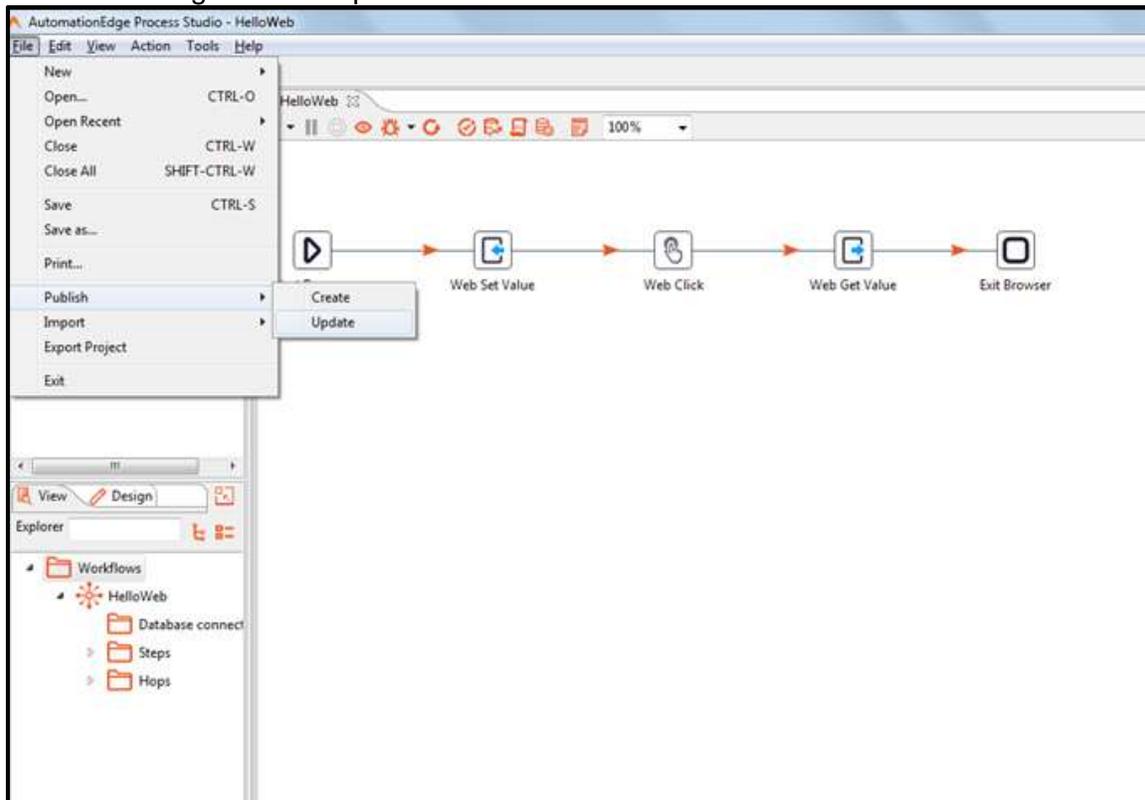
Name	Category	Metering Units	Created	Last Updated	Active	Actions
Get Single Stock Value	Default	0	9-Nov-2020 13:00:47	9-Nov-2020 13:06:25	<input checked="" type="checkbox"/>	  
HelloWorld	Default	0	5-Aug-2020 10:31:58	6-Aug-2020 19:16:01	<input checked="" type="checkbox"/>	  
HelloWebGeneric	Default	0	21-Jul-2020 15:43:21	6-Aug-2020 19:16:01	<input checked="" type="checkbox"/>	  
HelloWeb	Default	0	21-Jul-2020 09:08:51	17-Aug-2020 13:37:39	<input checked="" type="checkbox"/>	  

25. This completes the Process of publishing workflow to AutomationEdge (Development instance) using the Publish→Create option in Process Studio.
26. In further sections we will complete the lifecycle of exporting from Development instance, importing to UAT instance, exporting from UAT and finally importing to Production instance.

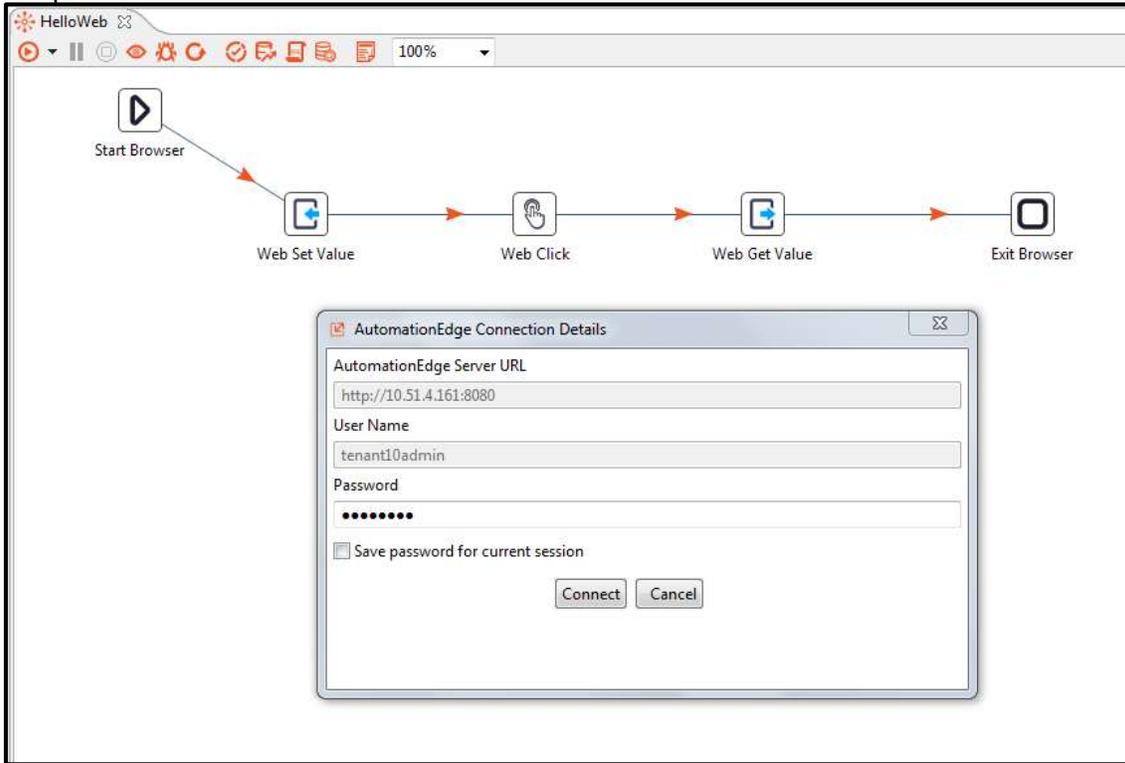
7.1.2 Publish: Update

In the previous section we completed the process of publishing workflow HelloWeb, for the first time with Publish→Create option, to AutomationEdge. In case we wish to make updates to this workflow we can use the Publish→Update option. Following are the steps to publish an AutomationEdge workflow using Process Studio Publish→Update option,

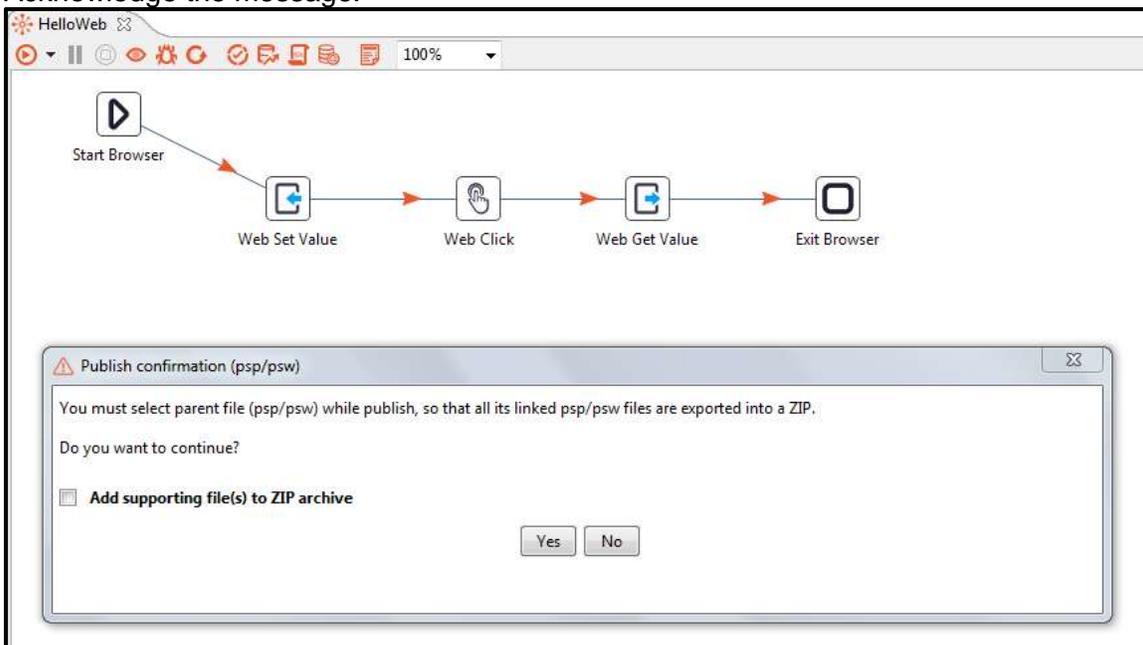
1. Open HelloWeb workflow in Process Studio as seen below.
2. In case you have a Process and multiple workflows then open the parent Process or parent Workflow and go to the respective tab.



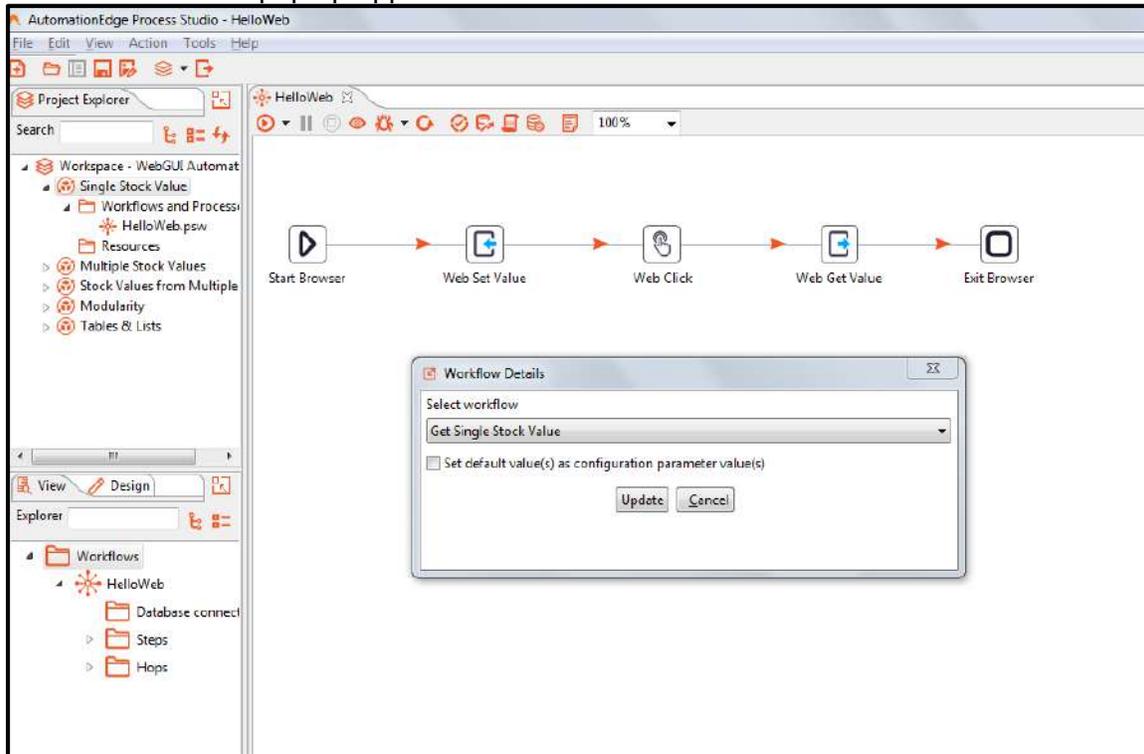
3. Select File menu and click Publish and then on Update option.
4. If you have not saved the password AutomationEdge connection Details dialog appears. Provide the password.



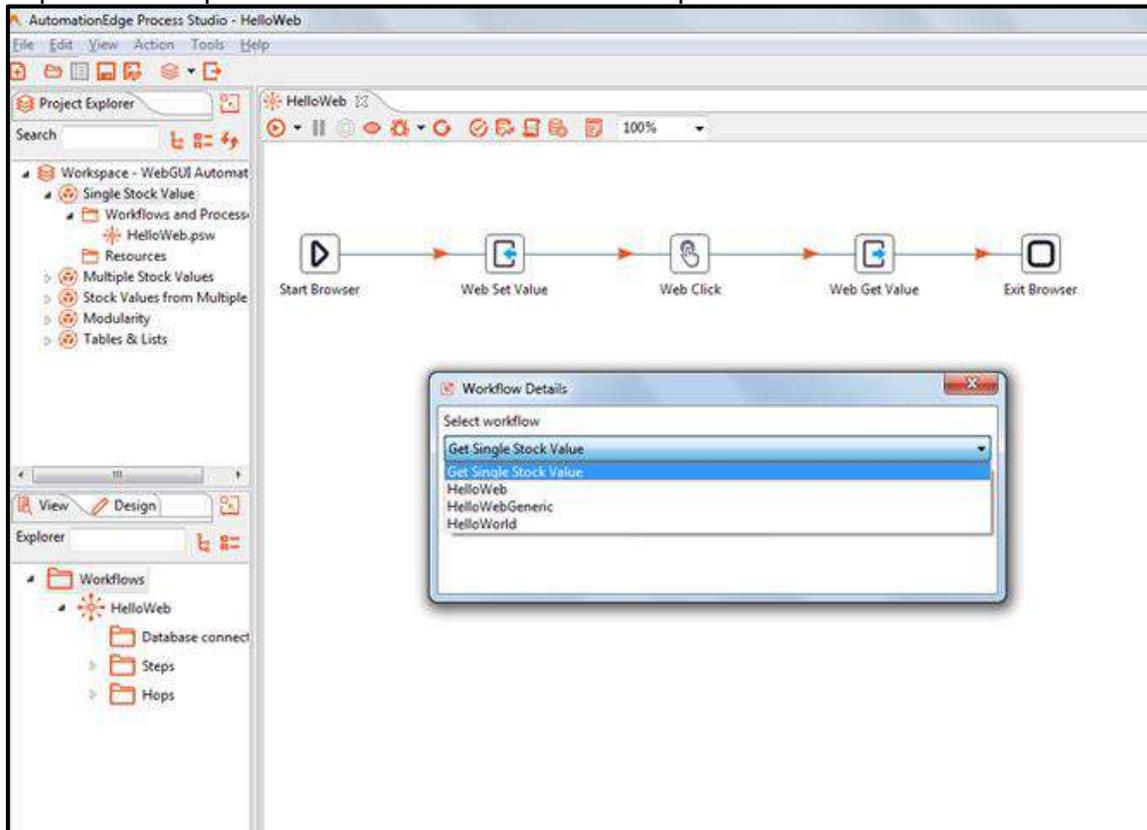
5. A pop-up message appears warning that you must select a parent process(psp) or workflow(psw) as the case may be so that all the linked psp/psw files are exported as a zip. Acknowledge the message.



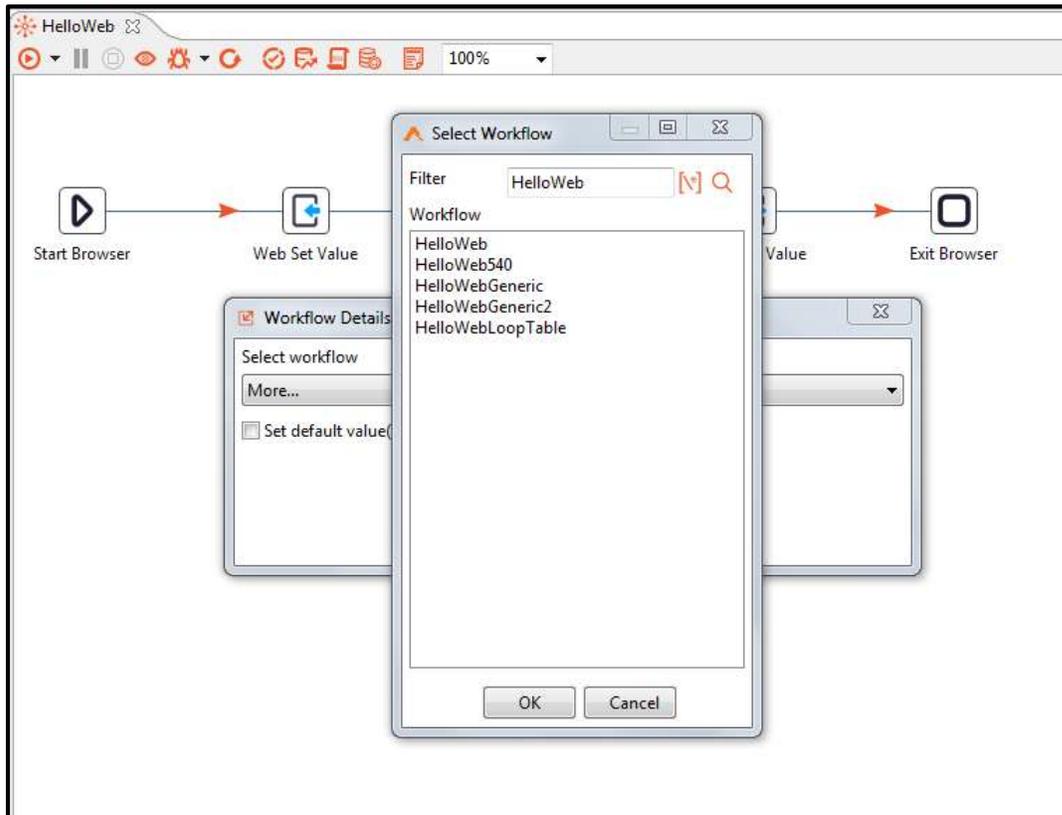
6. In case you check 'Add supporting files to zip archive' a corresponding popup appears.
7. In this case we do not have any supporting .csv, .txt, xlsx or other files. Leave 'Add supporting files to zip archive' unchecked and click Yes.
8. A Workflow Details pop up appears with details as seen below.



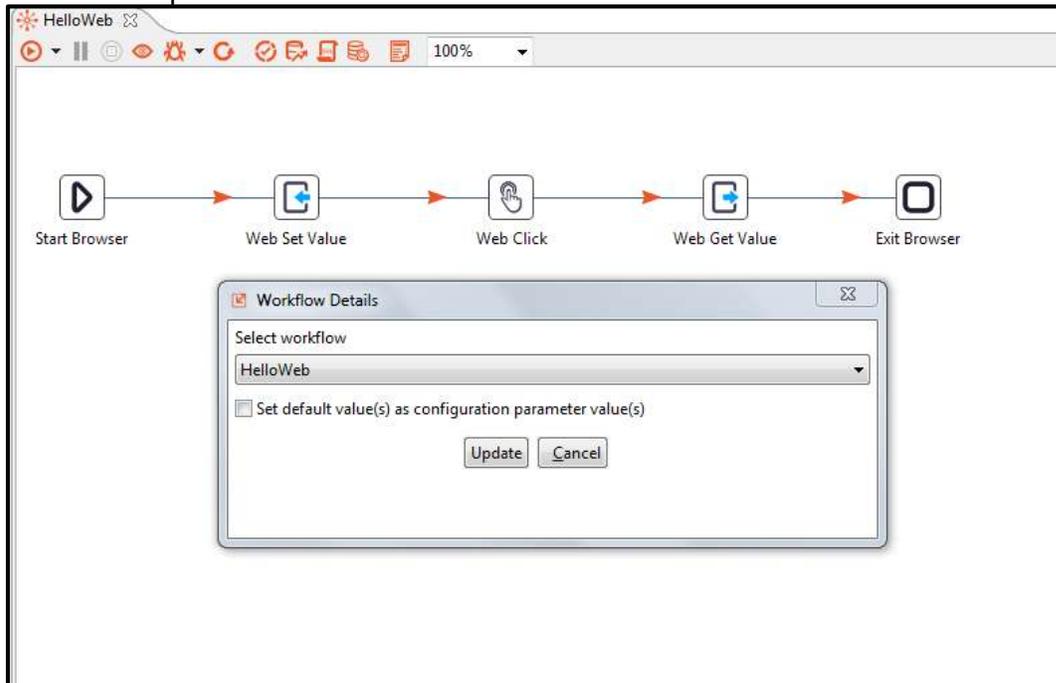
9. Expand the drop down list and Select Workflow to Update. Choose a workflow from the list.



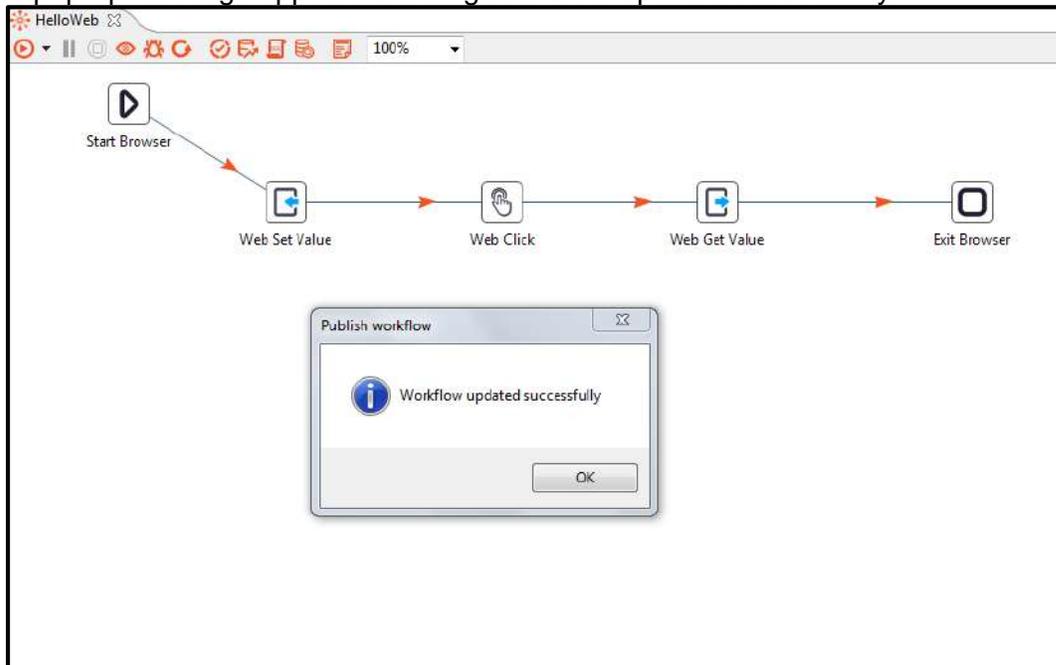
10. In case you do not see your workflow in this list, click More... in the drop down list, and locate your workflow. You may use Filter box to search narrow the list.
11. Click OK.



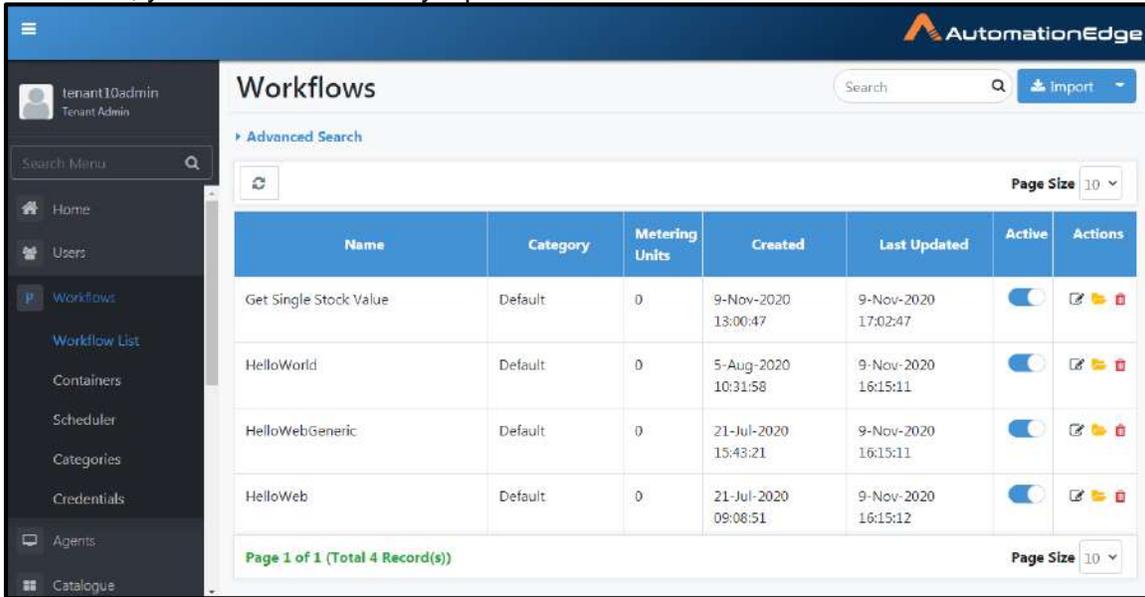
12. Parameters such as Assisted, Sequential, RDP enabled are inherited from the published workflow. If you wish to 'Set default value(s) as configuration parameter value(s)' enable check box. Click Update.



13. A pop-up message appears showing Workflow Updated successfully.



- This completes the Publish→Update process of workflow from Process Studio.
- Now logon to AutomationEdge instance that you used to publish the workflow. In the Workflow List menu, you can see the newly Updated HelloWorld workflow.



The screenshot displays the AutomationEdge Workflows management interface. The left sidebar shows the user 'tenant10admin' and a navigation menu with options like Home, Users, Workflows, Workflow List, Containers, Scheduler, Categories, Credentials, Agents, and Catalogue. The main content area is titled 'Workflows' and features a search bar, an 'Import' button, and a table of workflow records. The table has columns for Name, Category, Metering Units, Created, Last Updated, Active, and Actions. The 'HelloWorld' workflow is highlighted, showing it was last updated on 9-Nov-2020 at 16:15:11. The interface also shows a 'Page Size' dropdown set to 10 and a status bar indicating 'Page 1 of 1 (Total 4 Record(s))'.

Name	Category	Metering Units	Created	Last Updated	Active	Actions
Get Single Stock Value	Default	0	9-Nov-2020 13:00:47	9-Nov-2020 17:02:47	<input checked="" type="checkbox"/>	  
HelloWorld	Default	0	5-Aug-2020 10:31:58	9-Nov-2020 16:15:11	<input checked="" type="checkbox"/>	  
HelloWebGeneric	Default	0	21-Jul-2020 15:43:21	9-Nov-2020 16:15:11	<input checked="" type="checkbox"/>	  
HelloWeb	Default	0	21-Jul-2020 09:08:51	9-Nov-2020 16:15:12	<input checked="" type="checkbox"/>	  

16. Click the edit icon in the Actions column to make any updates to workflow configuration. As seen below most of the basic details were already provided during Publish→Create in Process Studio. Update as desired. Select Enable Sequential Execution if desired although initially it was not selected.

Configure Workflow Details

▼ Basic Details

Workflow Name: **Get Single Stock Value**

Workflow Description (Maximum 128 Characters): *

Fetches stock value based on Stock symbols provided

Workflow Category:

Default

Workflow Icon: 

Assisted Workflow : **false**

Enable Sequential Execution

Enable RDP

Enable Input Attributes

Workflow Priority:

Default

Expected Completion Time(Seconds): *

5

Maximum Completion Time(Seconds): *

20

Cleanup Requests older than(Hours):

36

Manual Execution Time:

5 Minutes

▶ Email Notification Setting

No Configuration Parameters



17. Update Email Notification Setting if desired.

18. Click Save.

▼ Email Notification Setting

Notify On Workflow Failure

Notify On Exceeding Time Limit

Select Users*:

By Role: Tenant Admin Workflow Admin

By Username:

By Email:

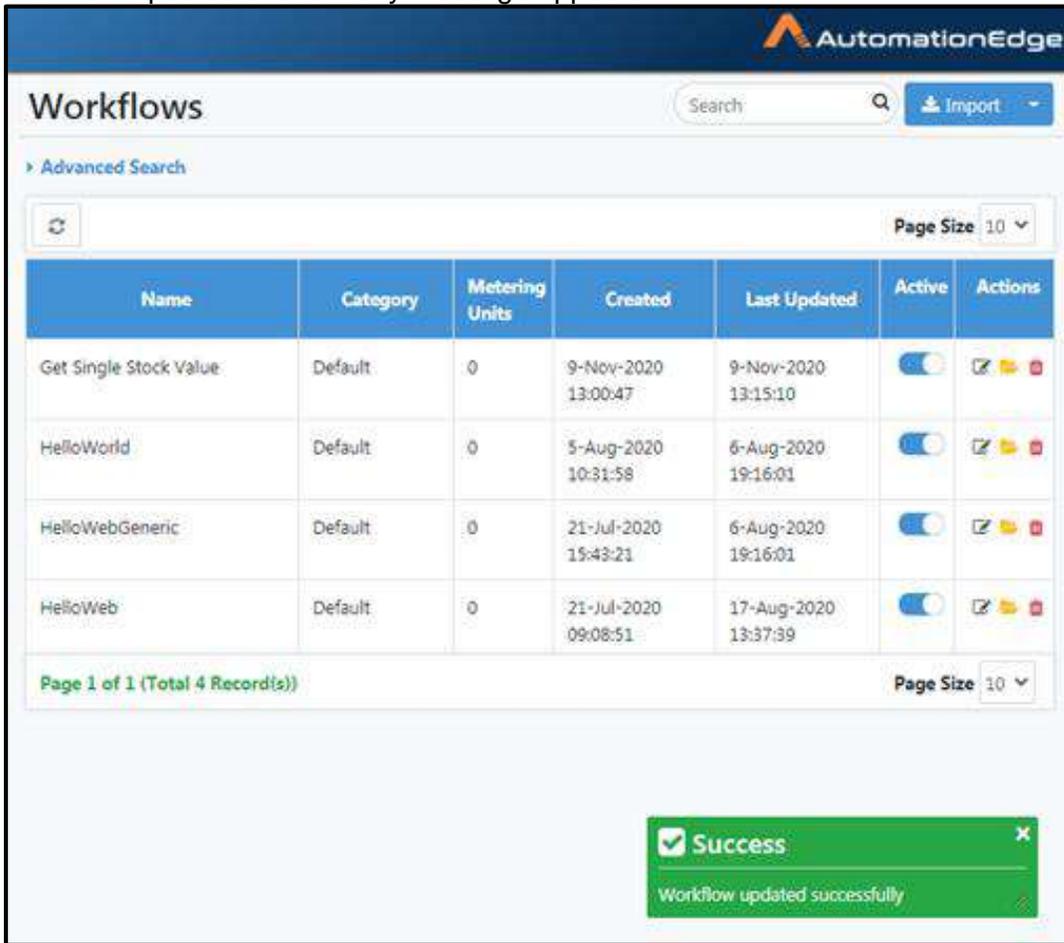
Request Creator

Failure Message:

No Configuration Parameters



19. Workflow updated successfully message appears.



The screenshot displays the AutomationEdge Workflows management interface. At the top, there is a search bar and an 'Import' button. Below the search bar is an 'Advanced Search' link and a refresh icon. The main content is a table with the following columns: Name, Category, Metering Units, Created, Last Updated, Active, and Actions. The table contains four rows of workflow data. At the bottom of the table, there is a pagination indicator showing 'Page 1 of 1 (Total 4 Record(s))' and a 'Page Size' dropdown set to 10. A green success message box is visible in the bottom right corner, stating 'Success' and 'Workflow updated successfully'.

Name	Category	Metering Units	Created	Last Updated	Active	Actions
Get Single Stock Value	Default	0	9-Nov-2020 13:00:47	9-Nov-2020 13:15:10	<input checked="" type="checkbox"/>	  
HelloWorld	Default	0	5-Aug-2020 10:31:58	6-Aug-2020 19:16:01	<input checked="" type="checkbox"/>	  
HelloWebGeneric	Default	0	21-Jul-2020 15:43:21	6-Aug-2020 19:16:01	<input checked="" type="checkbox"/>	  
HelloWeb	Default	0	21-Jul-2020 09:08:51	17-Aug-2020 13:37:39	<input checked="" type="checkbox"/>	  

20. This completes the process of publishing workflow to AutomationEdge using the Publish→Update option in Process Studio.

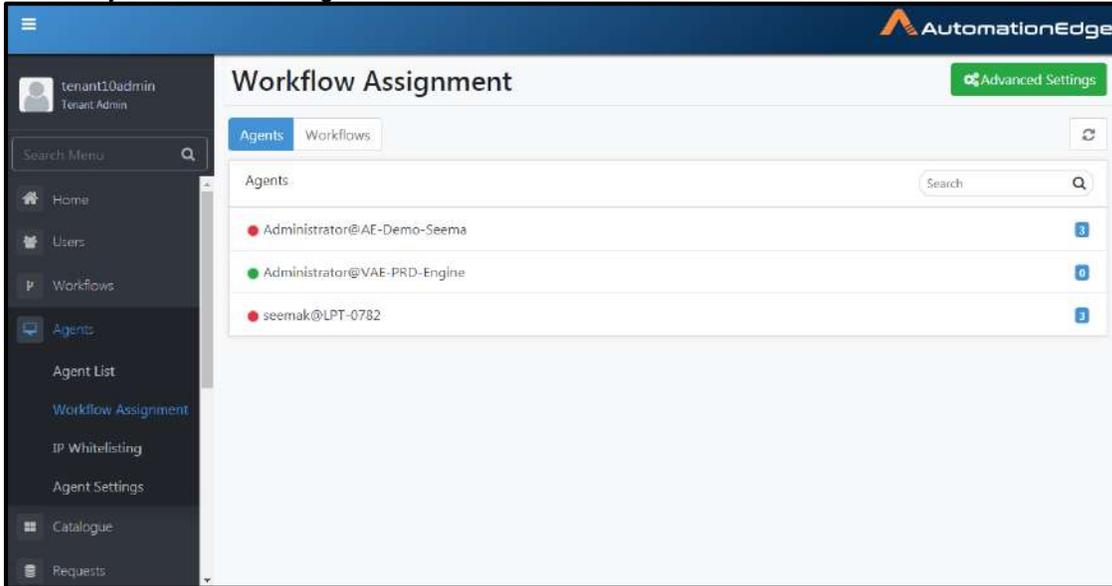
In the next section we shall setup and execute HelloWeb on this instance (Development instance); and in further sections we shall complete the life cycle of exporting from Development instance, importing to UAT instance, exporting from UAT and finally importing to Production instance.

7.2 Setup, Execute HelloWeb on Development Instance

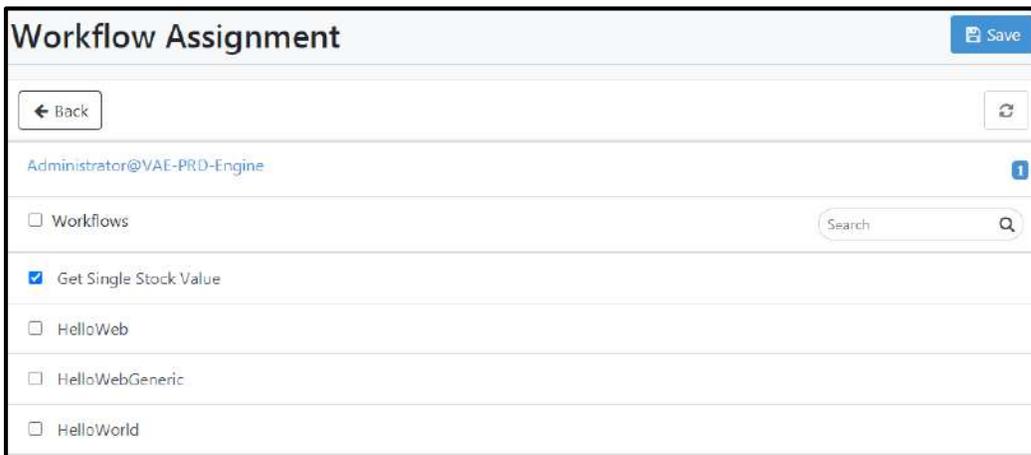
7.2.1 Assign Workflow to Agent

Following are steps to setup workflow – HelloWeb on AutomationEdge,

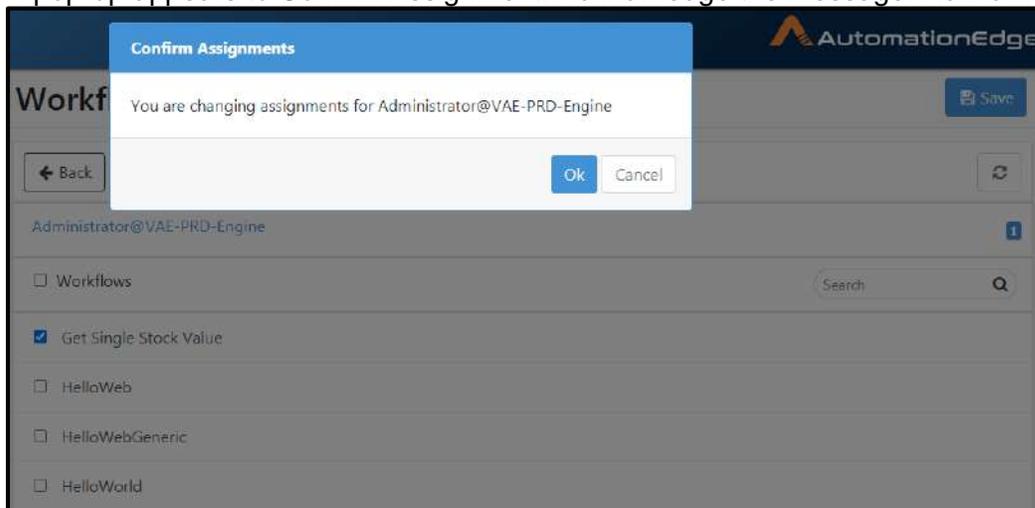
1. Navigate to Agents menu. Click Workflow Assignment sub-menu.
2. Click anywhere on the Agent Name row.



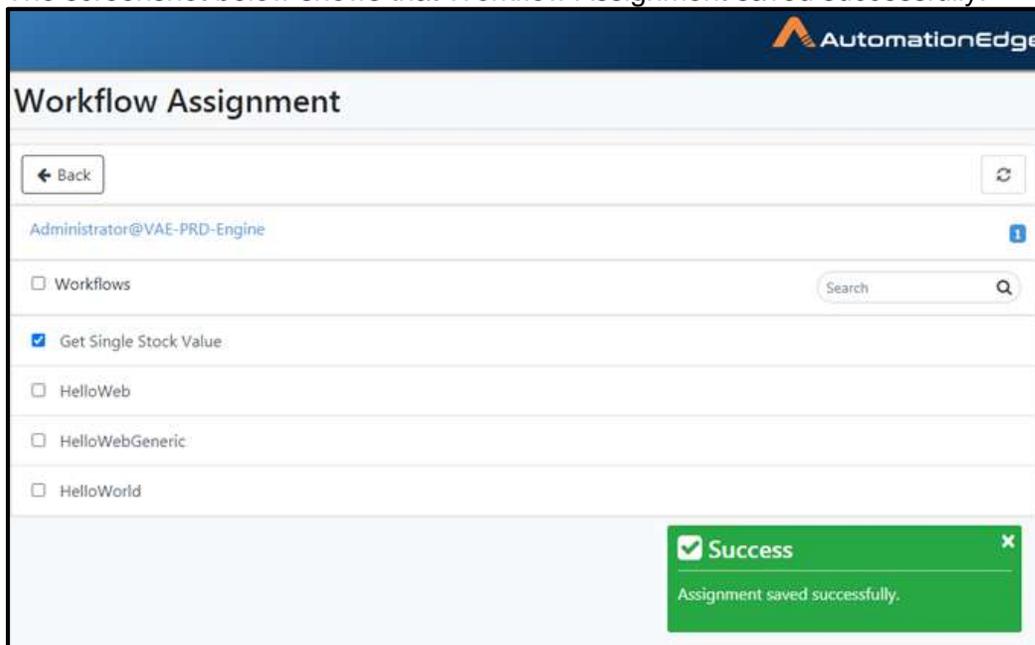
3. On the left hand side select the Agent.
4. On the right side enable checkbox for the workflows to be assigned to the Agent.
5. Click Save.
6. Enable checkbox next to HelloWebGeneric.



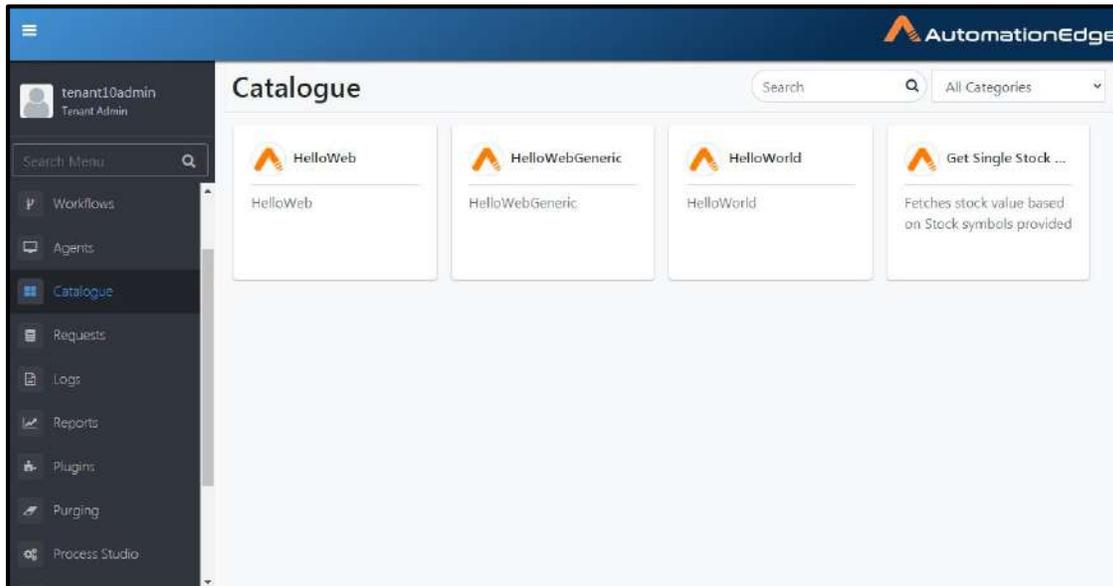
7. A pop-up appears to Confirm Assignment. Acknowledge the message. Acknowledge message.



8. The screenshot below shows that Workflow Assignment saved successfully.



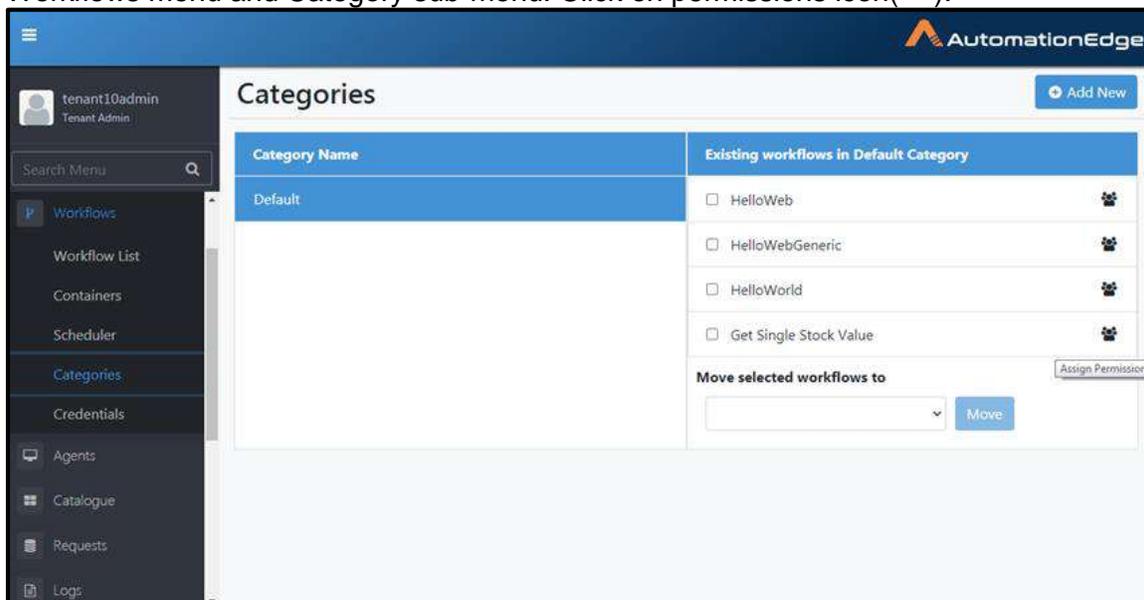
9. Go to Catalogue menu.
10. The workflow is now visible in the catalogue.
11. Click HelloWorld workflow icon.



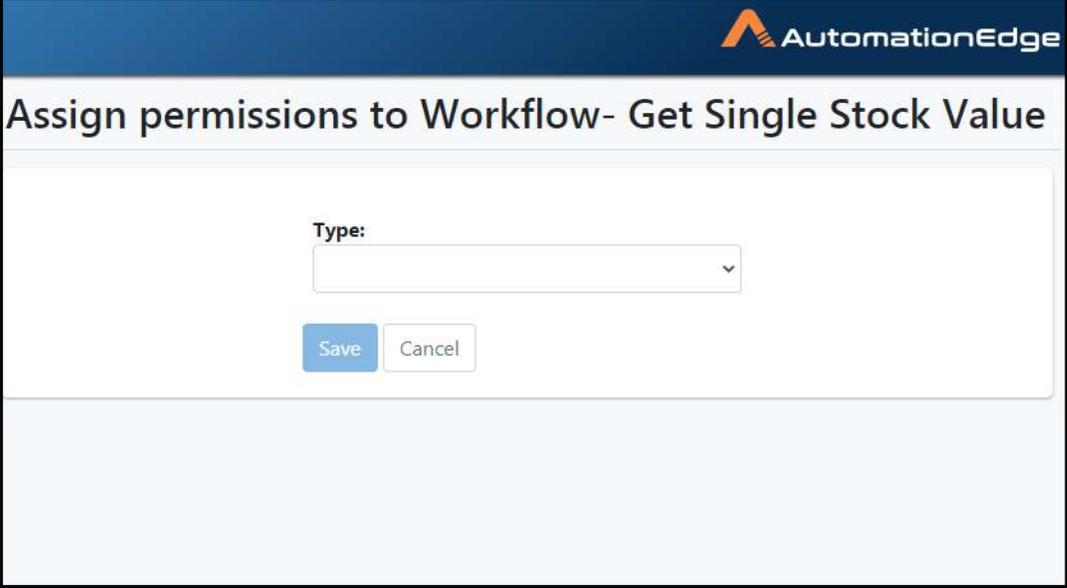
7.2.2 Assigning Permissions to Workflow

We need to assign workflow related permissions to users for them to be able to perform workflow operations.

1. necessary Permissions to Workflow to users or groups. To see the workflow as a catalog item for other users provide necessary Read/Write/Execute permissions to Users/Groups under Workflows menu and Category sub-menu. Click on permissions icon().



2. A pop-up to Assign permissions to workflow appears. Choose the permission recipient type from the drop down list.



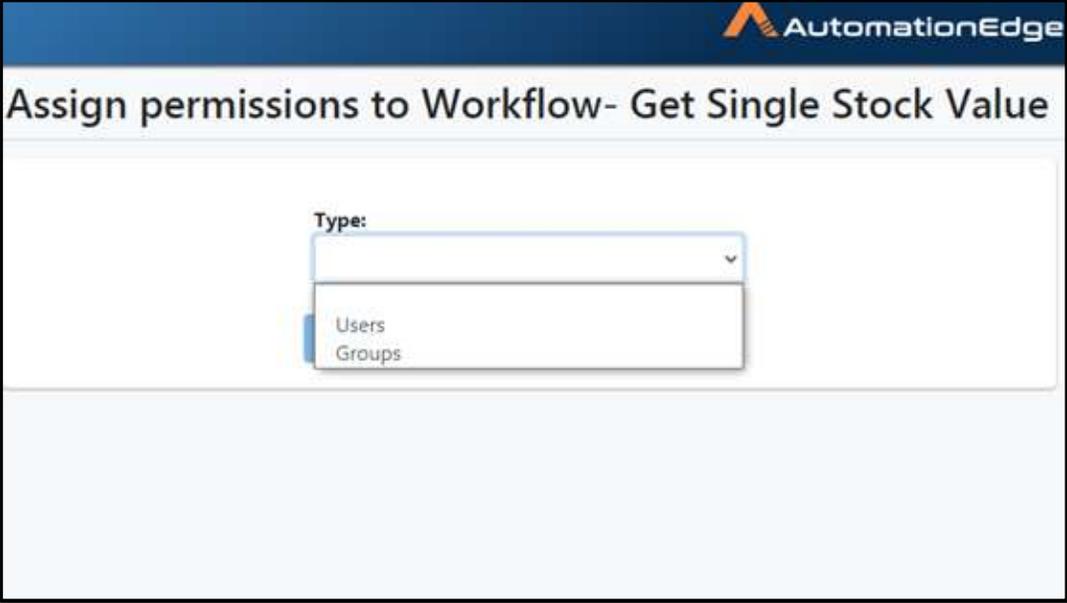
AutomationEdge

Assign permissions to Workflow- Get Single Stock Value

Type:

Save Cancel

3. Select from Users or Groups.



AutomationEdge

Assign permissions to Workflow- Get Single Stock Value

Type:

Users
Groups

- In this snapshot below Type **User** is chosen. Click the Members drop down list and select member **Tenant 1 User** for assigning Workflow permissions to the user.

AutomationEdge

Assign permissions to Workflow- Get Single Stock Value

Type:
Users

Members:
Tenant10 User x

Select All

Search

Tenant10 User

Save Cancel

- Click on the Members dropdown list to minimize it. Choose permissions to be assigned. In this case Read, Write and Execute permissions are assigned to Tenant 1 User.
- Click Save.

AutomationEdge

Assign permissions to Workflow- Get Single Stock Value

Type:
Users

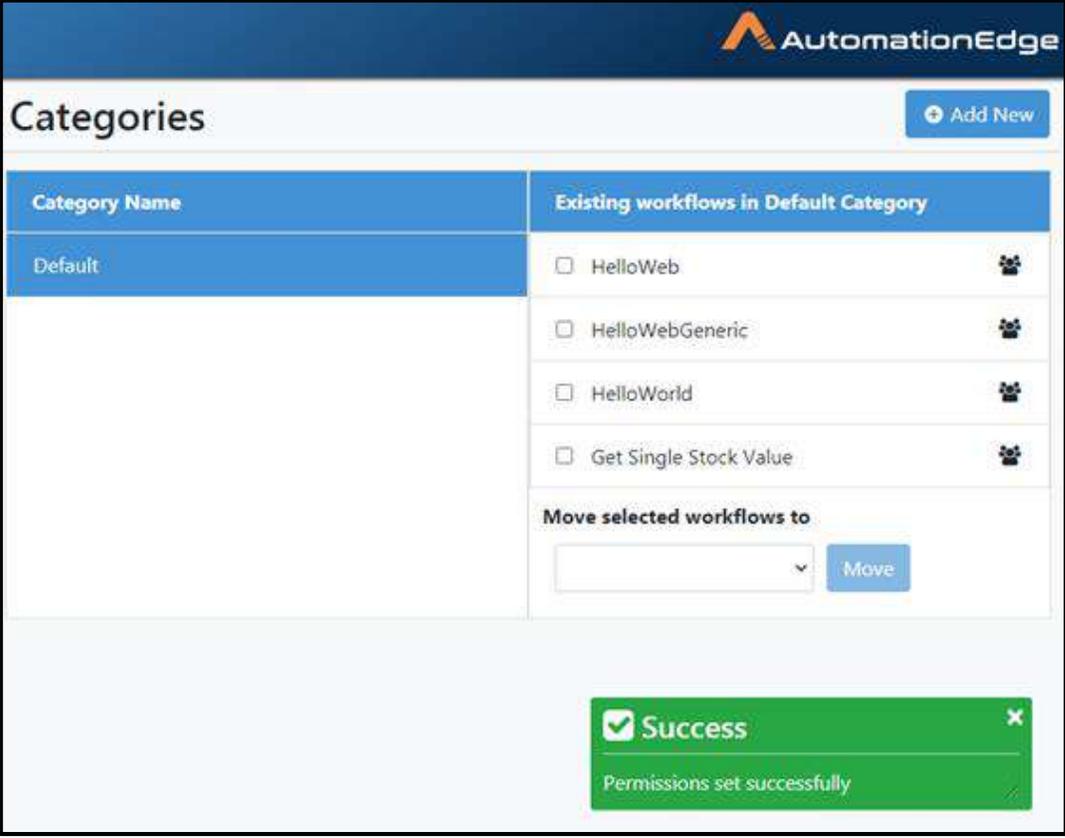
Members:
Tenant10 User x

Permissions

Users/Groups	Read	Execute	Write
Tenant10 User	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Save Cancel

7. Note the Permissions Set successfully message.



The screenshot shows the 'Categories' page in AutomationEdge. The page has a dark blue header with the AutomationEdge logo and the text 'AutomationEdge'. Below the header, the title 'Categories' is displayed on the left, and an 'Add New' button is on the right. The main content area contains a table with two columns: 'Category Name' and 'Existing workflows in Default Category'. The 'Default' category is selected, and it lists four workflows: 'HelloWeb', 'HelloWebGeneric', 'HelloWorld', and 'Get Single Stock Value'. Each workflow has a checkbox and a group icon. Below the table, there is a section titled 'Move selected workflows to' with a dropdown menu and a 'Move' button. At the bottom right, a green success message box reads 'Success' and 'Permissions set successfully'.

Category Name	Existing workflows in Default Category
Default	<input type="checkbox"/> HelloWeb 
	<input type="checkbox"/> HelloWebGeneric 
	<input type="checkbox"/> HelloWorld 
	<input type="checkbox"/> Get Single Stock Value 

Move selected workflows to

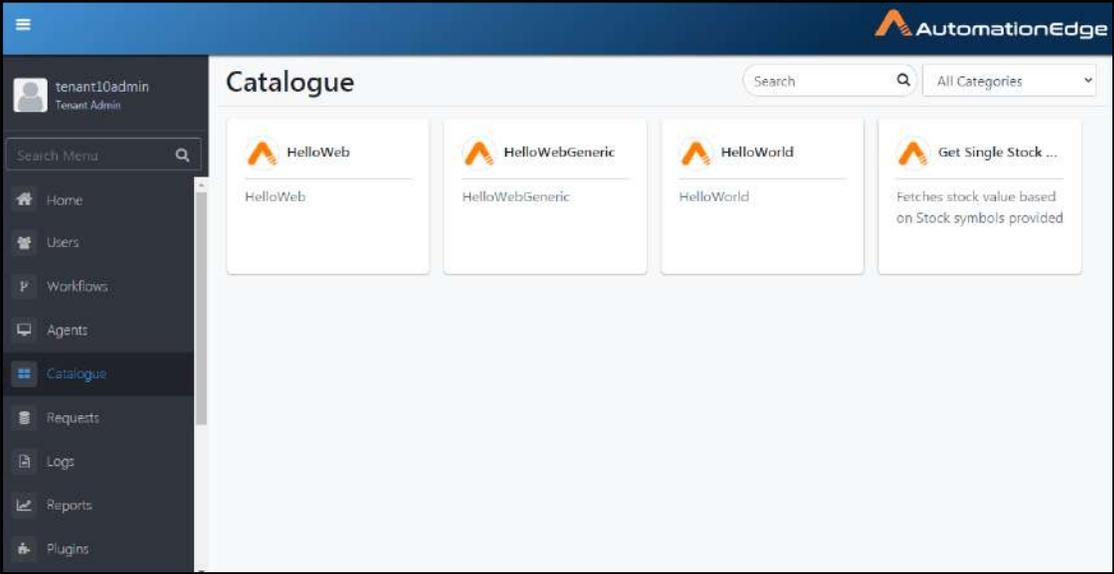
 **Success**

Permissions set successfully

7.2.3 Executing Workflows

The workflow can now be executed from the Catalogue.

1. Navigate to Catalogue menu.
2. Click on HelloWeb workflow.



The screenshot shows the 'Catalogue' page in AutomationEdge. The page has a dark blue header with the AutomationEdge logo and the text 'AutomationEdge'. Below the header, the title 'Catalogue' is displayed on the left, and a search bar and a dropdown menu are on the right. The main content area contains four workflow cards: 'HelloWeb', 'HelloWebGeneric', 'HelloWorld', and 'Get Single Stock ...'. Each card has a group icon and a description. The 'Catalogue' menu item is highlighted in the left sidebar.

tenant10admin
Tenant Admin

Search Menu

Home
Users
Workflows
Agents
Catalogue
Requests
Logs
Reports
Plugins

Catalogue

Search All Categories

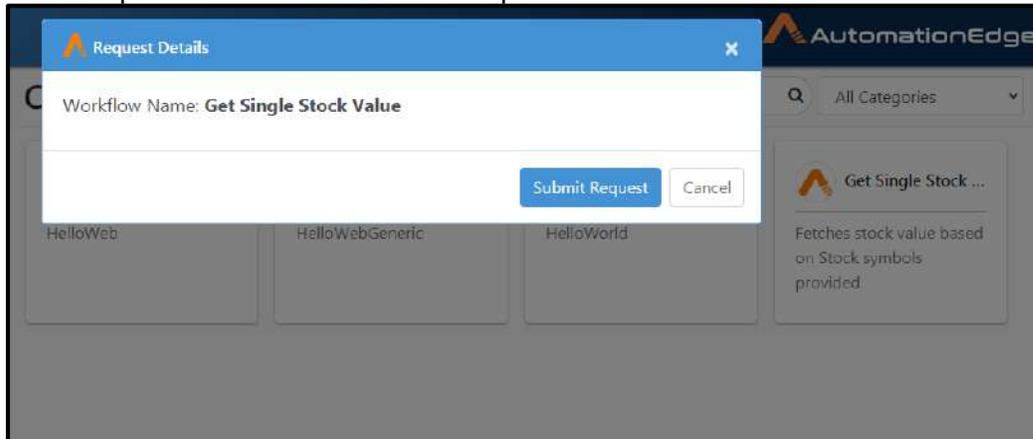
 HelloWeb
HelloWeb

 HelloWebGeneric
HelloWebGeneric

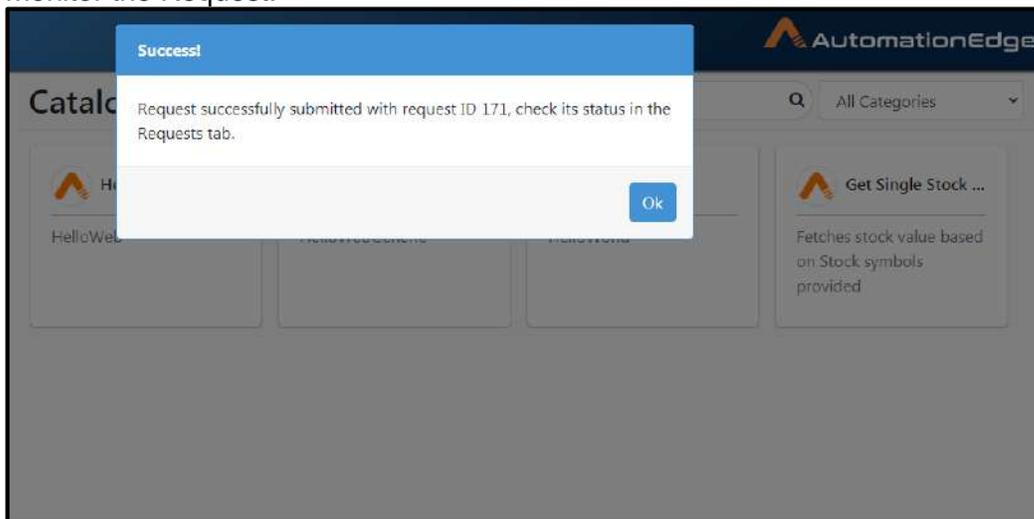
 HelloWorld
HelloWorld

 Get Single Stock ...
Fetches stock value based on Stock symbols provided

12. Acknowledge the pop-up for successful request submission. This workflow does not have any run time parameters. Click Submit Request.



13. A popup appears showing that message was submitted successfully. Note the Request ID and monitor the Request.



7.2.4 Monitor Requests

1. Click Requests menu. Identify your Request Id.
2. The screen shot below shows HelloWeb is in ExecutionStarted status.

- Expand the + sign next to execution started to see the red square. You may click on the red square to terminate the request.

Id	Workflow Name	Status	Agent Name	Created	Completed
171	Get Single Stock Value	ExecutionStarted	Administrator@VAE-PRD-...	9-Nov-2020 14:31:07	9-Nov-2020 14:32:00
12	HelloWebGeneric	Complete	seemak@LPT-0782_d1596...	29-Jul-2020 09:58:21	29-Jul-2020 09:58:43
10	HelloWeb	Complete	seemak@LPT-0782_d1596...	29-Jul-2020 09:57:49	29-Jul-2020 09:58:07
9	HelloWebGeneric	Complete	seemak@LPT-0782_d1596...	29-Jul-2020 08:01:16	29-Jul-2020 08:03:07
7	HelloWeb	Complete	seemak@LPT-0782_d1596...	29-Jul-2020 07:59:26	29-Jul-2020 08:02:34
5	HelloWebGeneric	Complete	Administrator@VAE-PRD-...	21-Jul-2020 17:31:30	21-Jul-2020 17:31:54
3	HelloWeb	Complete	Administrator@VAE-PRD-...	21-Jul-2020 10:55:18	21-Jul-2020 10:55:37

- In the screenshot below the status of request is now completed successfully.

Id	Workflow Name	Status	Agent Name	Created	Completed
171	Get Single Stock Value	Complete	Administrator@VAE-PR...	9-Nov-2020 14:31:07	9-Nov-2020 16:04:20
Message : Execution Successful					
12	HelloWebGeneric	Complete	seemak@LPT-0782_d15...	29-Jul-2020 09:58:21	29-Jul-2020 09:58:43
10	HelloWeb	Complete	seemak@LPT-0782_d15...	29-Jul-2020 09:57:49	29-Jul-2020 09:58:07
9	HelloWebGeneric	Complete	seemak@LPT-0782_d15...	29-Jul-2020 08:01:16	29-Jul-2020 08:03:07
7	HelloWeb	Complete	seemak@LPT-0782_d15...	29-Jul-2020 07:59:26	29-Jul-2020 08:02:34
5	HelloWebGeneric	Complete	Administrator@VAE-PR...	21-Jul-2020 17:31:30	21-Jul-2020 17:31:54
3	HelloWeb	Complete	Administrator@VAE-PR...	21-Jul-2020 10:55:18	21-Jul-2020 10:55:37

- This completes publishing and executing HelloWeb Workflow on AutomationEdge server.

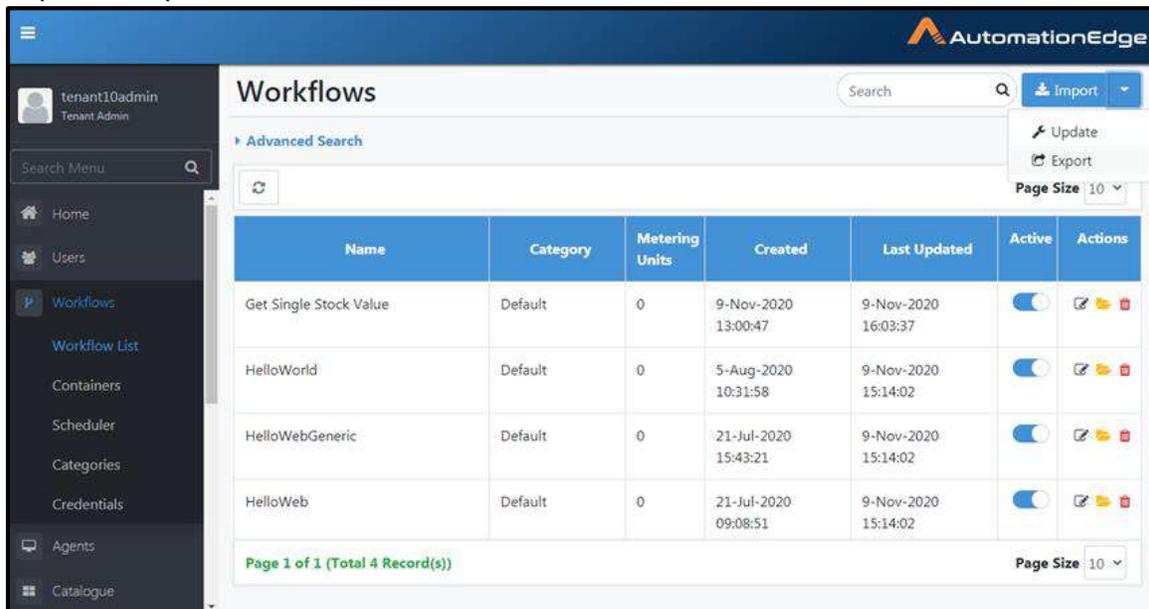
7.3 Workflows: Export from Development Instance

Once workflow is found to be working satisfactorily on Development instance, it's usually exported and then uploaded on UAT. A workflow can be exported from development instance and then imported to UAT or Production (Enterprise/Subscription) instances, although it is usually imported to UAT server.

The exported workflow is self-contained i.e. it has all the .psp and .psw it needs, and all the supporting files that are required for execution. The exported zip file from AutomationEdge also contains a signature and a manifest file. The signature and a manifest file are used for verification during import to other instances.

Following are the steps to export one or more Workflows,

1. Go the Workflows menu. Workflow List sub-menu is selected by default.
2. Click the arrow next to the Import button on the top right corner. From the drop down list click Export to Export Process Studio Workflow.

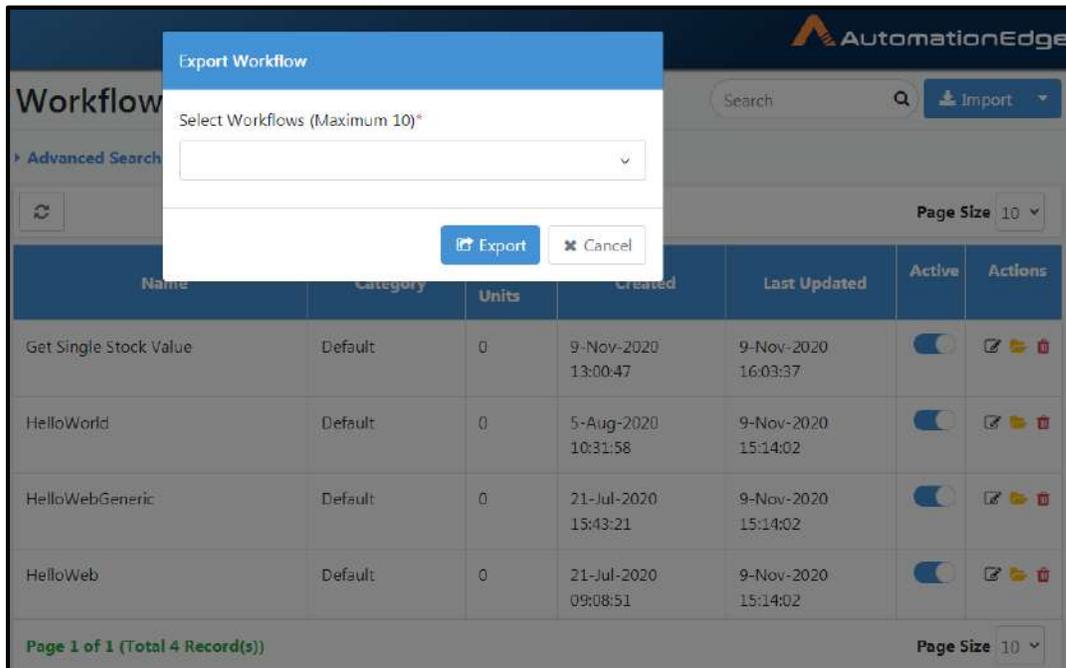


The screenshot displays the AutomationEdge Workflows management interface. The left sidebar shows the navigation menu with 'Workflows' selected. The main content area shows a table of workflows. The 'Actions' column for each workflow includes an 'Export' icon. A dropdown menu is open next to the 'Import' button, showing 'Update' and 'Export' options.

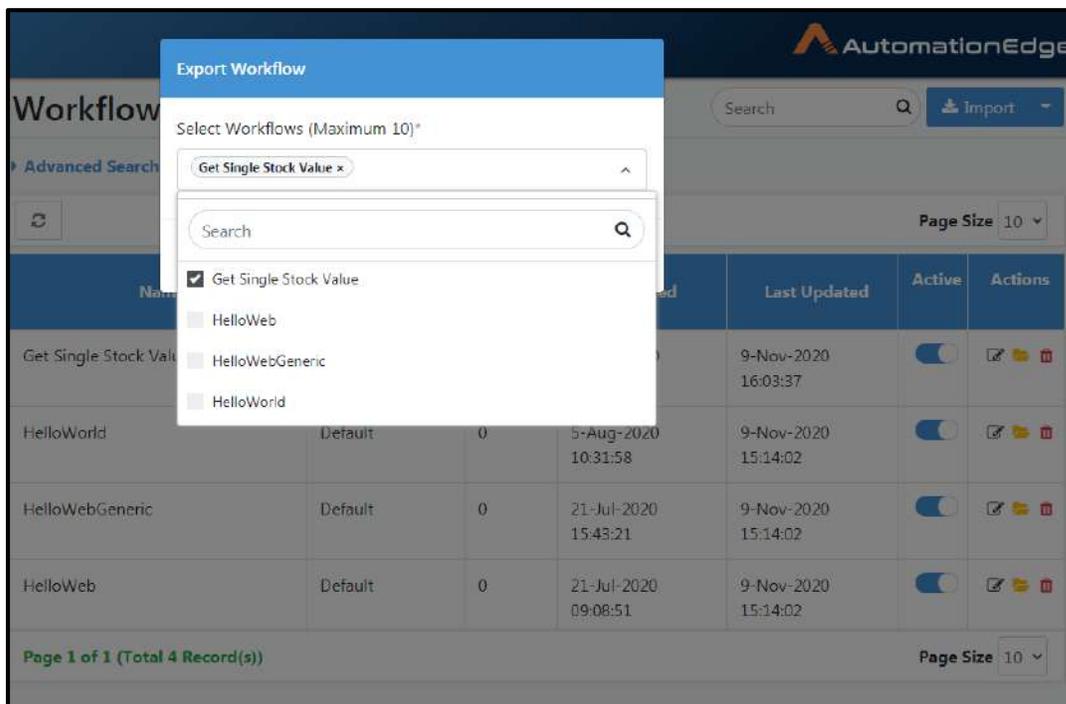
Name	Category	Metering Units	Created	Last Updated	Active	Actions
Get Single Stock Value	Default	0	9-Nov-2020 13:00:47	9-Nov-2020 16:03:37	<input checked="" type="checkbox"/>	  
HelloWorld	Default	0	5-Aug-2020 10:31:58	9-Nov-2020 15:14:02	<input checked="" type="checkbox"/>	  
HelloWebGeneric	Default	0	21-Jul-2020 15:43:21	9-Nov-2020 15:14:02	<input checked="" type="checkbox"/>	  
HelloWeb	Default	0	21-Jul-2020 09:08:51	9-Nov-2020 15:14:02	<input checked="" type="checkbox"/>	  

Page 1 of 1 (Total 4 Record(s))

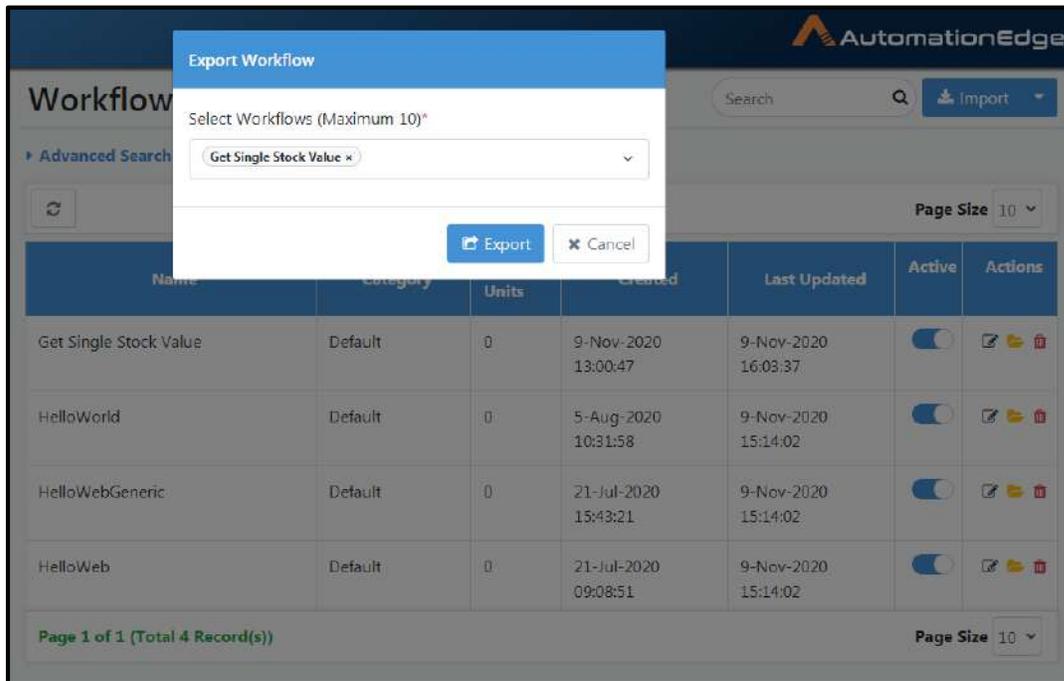
3. 'Export Workflow' window appears.



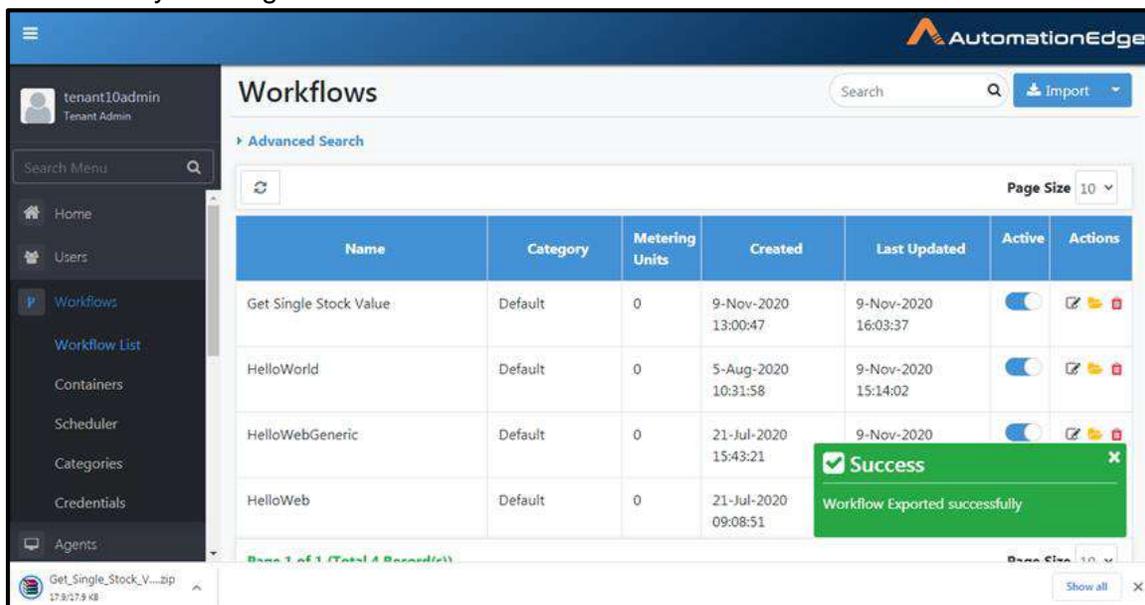
4. Select a workflow or workflows to export. Enable check boxes next to workflows to be exported. Enable checkbox next to HelloWeb.



5. HelloWeb is selected.



6. The below snapshot shows the exported zip file at the bottom left corner and Workflow Exported Successfully Message.



7.4 Workflows: Import to UAT Instance

Import button on Workflow menu supports import of Process Studio Processes and Workflows. It can be used to import the zip file exported from Process Studio. Adapter workflows are for backward compatibility only. There is no provision to import Adapter workflows.

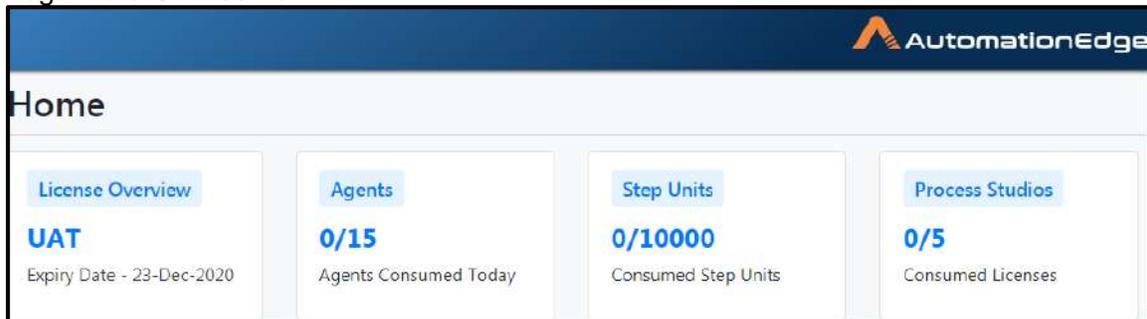
Once workflow is found to be working satisfactorily on Development instance, it's usually uploaded on UAT. A workflow can be exported from development instance and then imported to UAT or Production (Enterprise/Subscription) instances, although it is usually imported to UAT server.

AE server checks the exported zip file from any AutomationEdge instance - that zip file is not tampered with rejects if not valid.

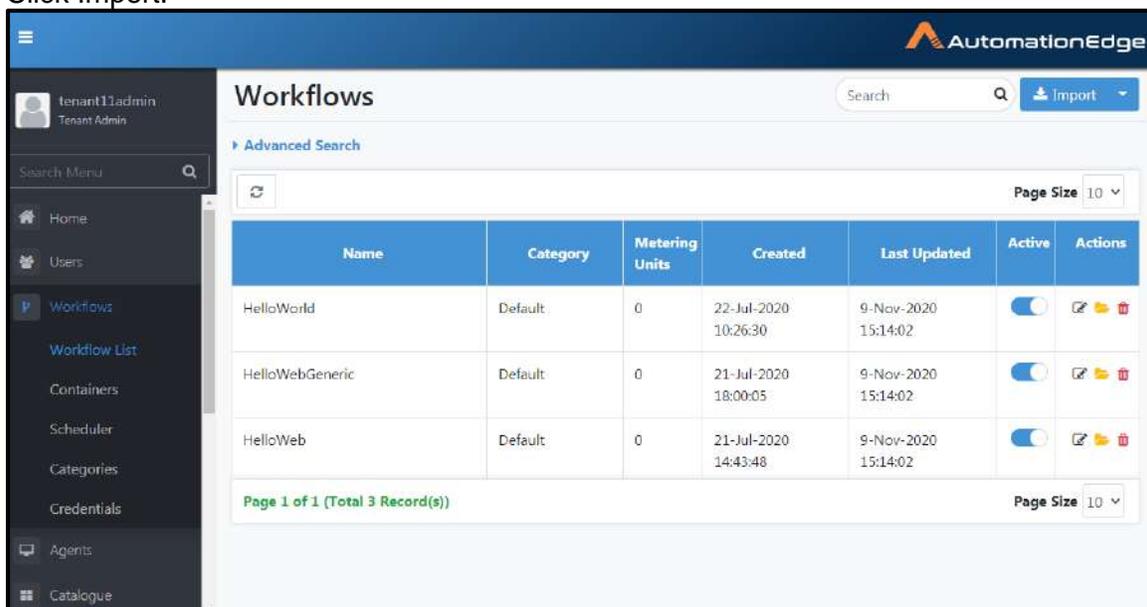
Import the workflow as exported from Development Instance. The process to import the workflow is exactly same as Import to Development instance. Detailed steps to import a workflow have been shown during the process of import to Development instance.

Following are the steps to import the workflow UAT in short,

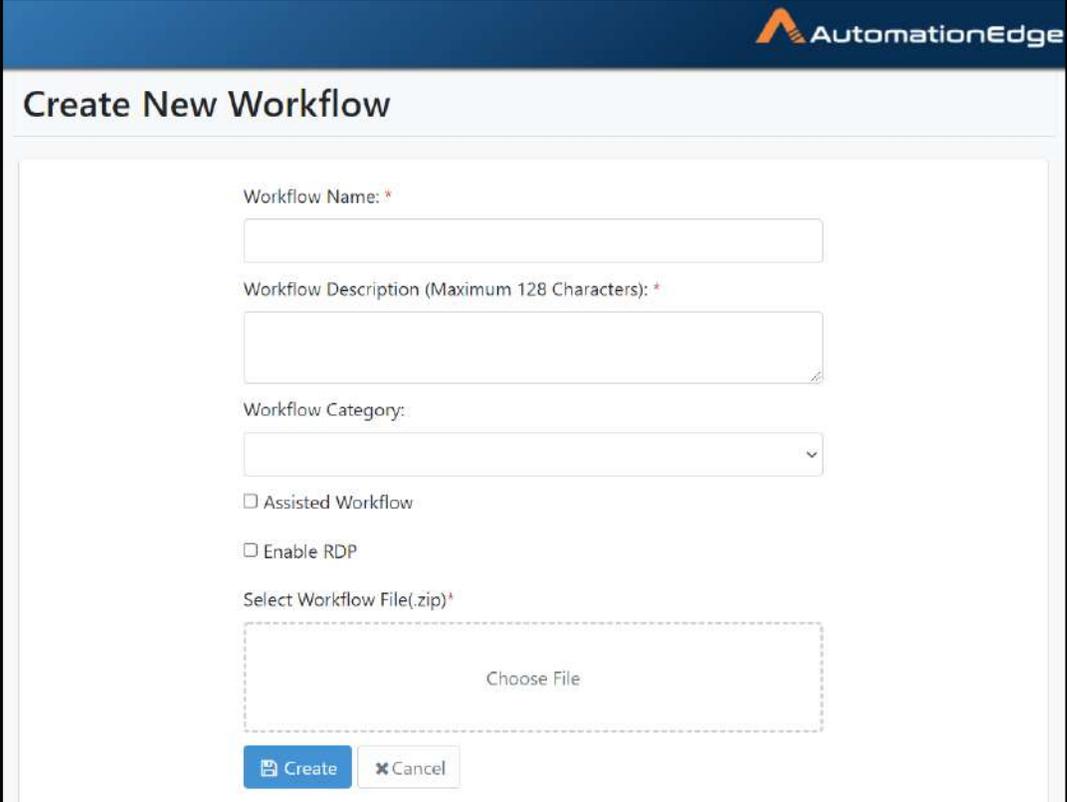
1. Logon to a UAT server.



2. Import the workflow as exported from Development Instance
3. Navigate to the Workflows→Workflow List menu.
4. Click Import.



5. Create a New Workflow screen appears.



AutomationEdge

Create New Workflow

Workflow Name: *

Workflow Description (Maximum 128 Characters): *

Workflow Category:

Assisted Workflow

Enable RDP

Select Workflow File(.zip)*

Choose File

Create Cancel

6. Provide details for the new workflow. Browse and select the exported zip file from Development instance.
7. Click Show File Contents to see the contents of the zip file.
8. Click Create.

AutomationEdge

Create New Workflow

Workflow Name: *

Workflow Description (Maximum 128 Characters): *

Workflow Category:

Default
▼

Assisted Workflow

Enable RDP

Select Workflow File(.zip)*

Get_Single_Stock_Value.zip

Show file content

Create
✖ Cancel

Content:

- ▣
Get_Single_Stock_Value.zip
 - signature.txt
- ▣
Get_Single_Stock_Value.zip
 - HelloWeb.psw
 - workflow.json

Create
✖ Cancel

↑

- The Configure Workflow Details page appears. Provide details in **Basic Details** section as shown below. Enable Sequential Execution is displayed as true and is uneditable in the case of a GUI automation workflow. In this case since the workflow is not detected as sequential, the checkbox is editable.

Configure Workflow Details

▼ **Basic Details**

Workflow Name: **Get Single Stock Value**

Workflow Description (Maximum 128 Characters): *

Fetches stock value based on Stock symbols provided

Workflow Category:

Default ▼

Workflow Icon: Choose File

Assisted Workflow : **false**

Enable Sequential Execution

Enable RDP

Enable Input Attributes

Workflow Priority:

Default ▼

Expected Completion Time(Seconds): *

5

Maximum Completion Time(Seconds): *

20

Cleanup Requests older than(Hours):

36

Manual Execution Time:

5

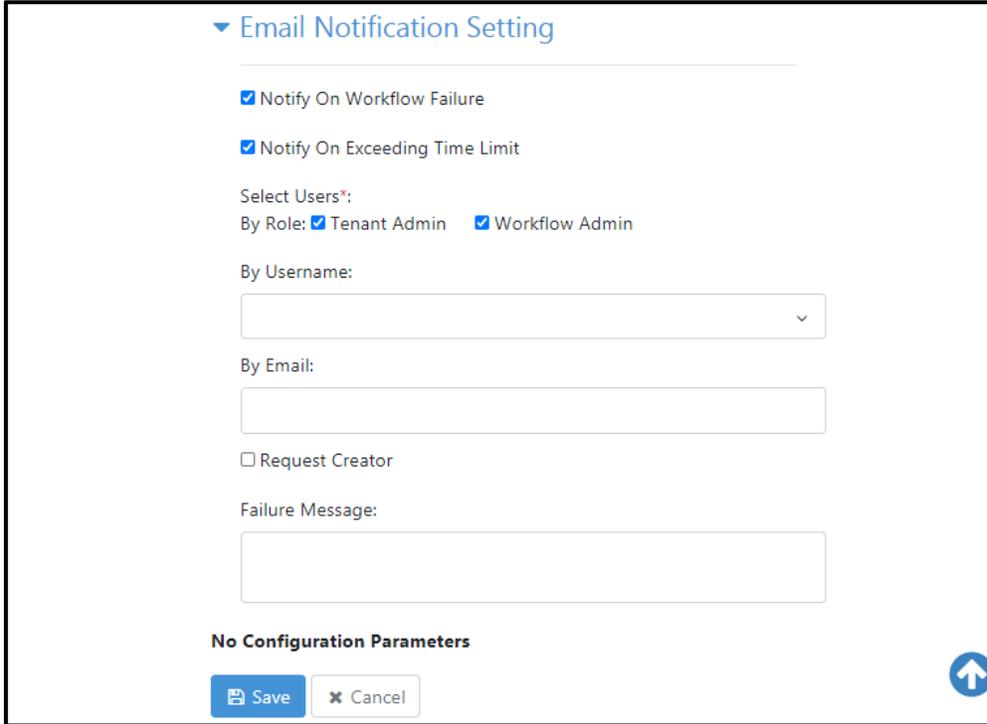
Minutes ▼

▶ **Email Notification Setting**

No Configuration Parameters

Save
Cancel

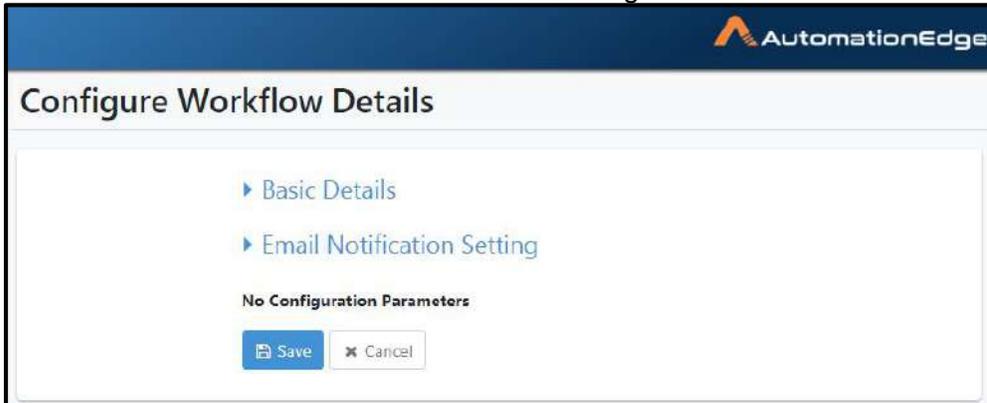
10. Provide details in **Email Notification Setting** section as shown below.



The screenshot shows the 'Email Notification Setting' configuration panel. It includes the following elements:

- Section Header:** Email Notification Setting (with a dropdown arrow)
- Checkboxes:**
 - Notify On Workflow Failure
 - Notify On Exceeding Time Limit
- Select Users*:**
 - By Role: Tenant Admin Workflow Admin
- By Username:** A dropdown menu.
- By Email:** A text input field.
- Request Creator:** Request Creator
- Failure Message:** A text area.
- Footer:** 'No Configuration Parameters', 'Save' button, 'Cancel' button, and an upward arrow icon.

11. Once Basic Details and Email Notification Settings are set click Save.



The screenshot shows the 'Configure Workflow Details' page. It includes the following elements:

- Section Header:** Configure Workflow Details
- Navigation:** Two expandable sections: 'Basic Details' and 'Email Notification Setting'.
- Footer:** 'No Configuration Parameters', 'Save' button, and 'Cancel' button.

12. Workflow updated successfully message appears. Note that initially the workflow is in inactive status. Click the toggle button to activate it.

The screenshot shows the AutomationEdge Workflows interface. At the top, there is a search bar and an 'Import' button. Below the search bar is an 'Advanced Search' section with a refresh icon and a 'Page Size' dropdown set to 10. The main content is a table with the following columns: Name, Category, Metering Units, Created, Last Updated, Active, and Actions. The table contains four rows of workflow data. The first row, 'Get Single Stock Value', has its 'Active' toggle switch turned off. A green success message box is displayed at the bottom right of the table area, stating 'Success Workflow updated successfully'.

Name	Category	Metering Units	Created	Last Updated	Active	Actions
Get Single Stock Value	Default	0	9-Nov-2020 17:05:07	9-Nov-2020 17:05:27	<input type="checkbox"/>	
HelloWorld	Default	0	22-Jul-2020 10:26:30	9-Nov-2020 16:19:25	<input checked="" type="checkbox"/>	
HelloWebGeneric	Default	0	21-Jul-2020 18:00:05	9-Nov-2020 16:19:25	<input checked="" type="checkbox"/>	
HelloWeb	Default	0	21-Jul-2020 14:43:48	9-Nov-2020 16:19:26	<input checked="" type="checkbox"/>	

Page 1 of 1 (Total 4 Record(s)) Page Size 10

13. You can now see the newly imported and updated workflow in the Workflow List in Active status.

The screenshot shows the AutomationEdge Workflows interface after the workflow has been activated. The table now shows the 'Get Single Stock Value' workflow with its 'Active' toggle switch turned on. The 'Last Updated' timestamp for this workflow is now 17:09:36. The success message is no longer present.

Name	Category	Metering Units	Created	Last Updated	Active	Actions
Get Single Stock Value	Default	0	9-Nov-2020 17:05:07	9-Nov-2020 17:09:36	<input checked="" type="checkbox"/>	
HelloWorld	Default	0	22-Jul-2020 10:26:30	9-Nov-2020 16:19:25	<input checked="" type="checkbox"/>	
HelloWebGeneric	Default	0	21-Jul-2020 18:00:05	9-Nov-2020 16:19:25	<input checked="" type="checkbox"/>	
HelloWeb	Default	0	21-Jul-2020 14:43:48	9-Nov-2020 16:19:26	<input checked="" type="checkbox"/>	

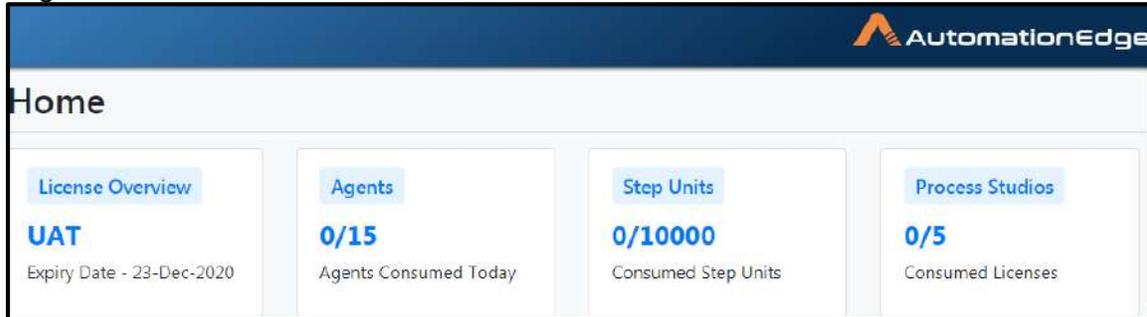
Page 1 of 1 (Total 4 Record(s)) Page Size 10

7.5 Workflows: Export from UAT Instance

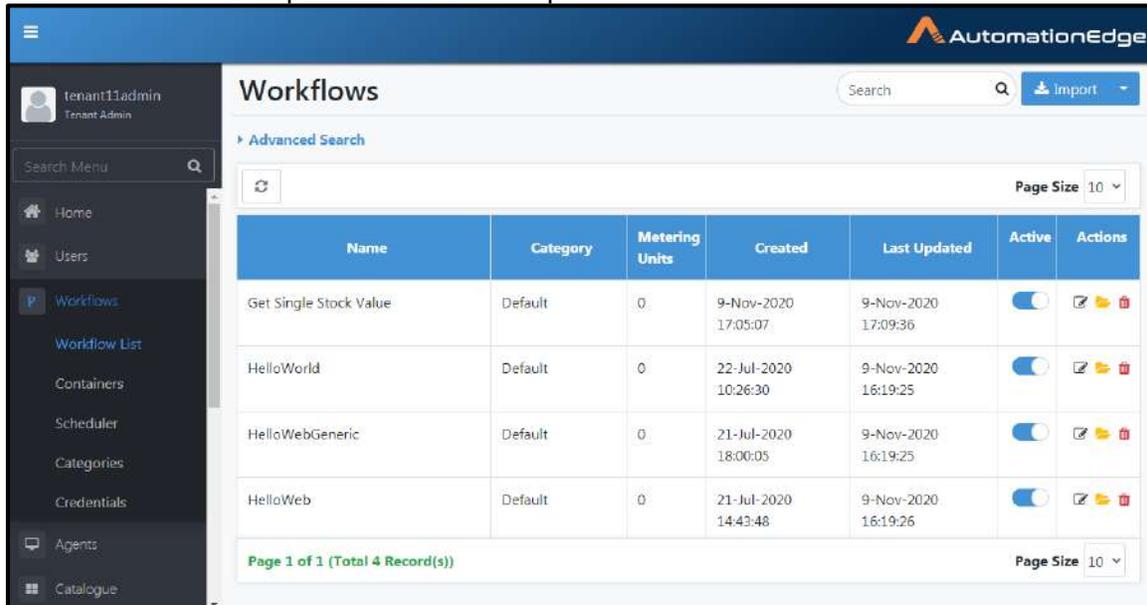
Once workflow is found to be working satisfactorily on UAT instance, it's usually uploaded on Production instance. A workflow can be exported from UAT instance and imported to Production (Enterprise/Subscription) instances.

The exported workflow is self-contained. i.e. it has all the .psp and .psw it needs, and all the supporting files that are required for execution. The exported zip file from AutomationEdge also contains a signature and a manifest file.

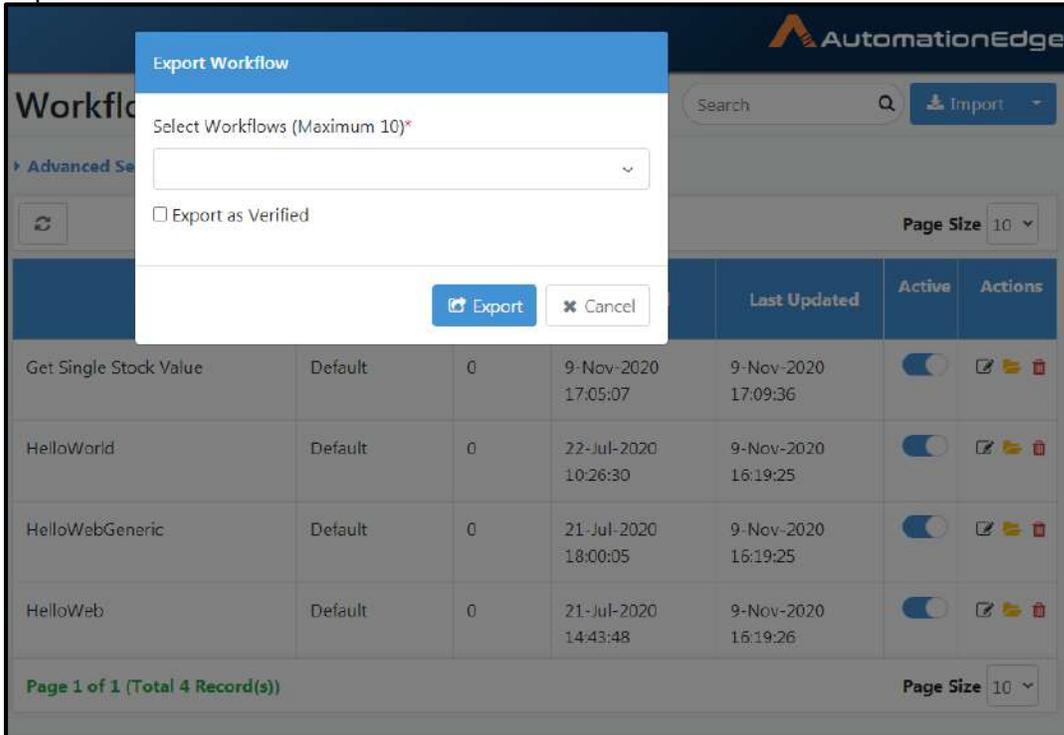
1. Login to a UAT instance



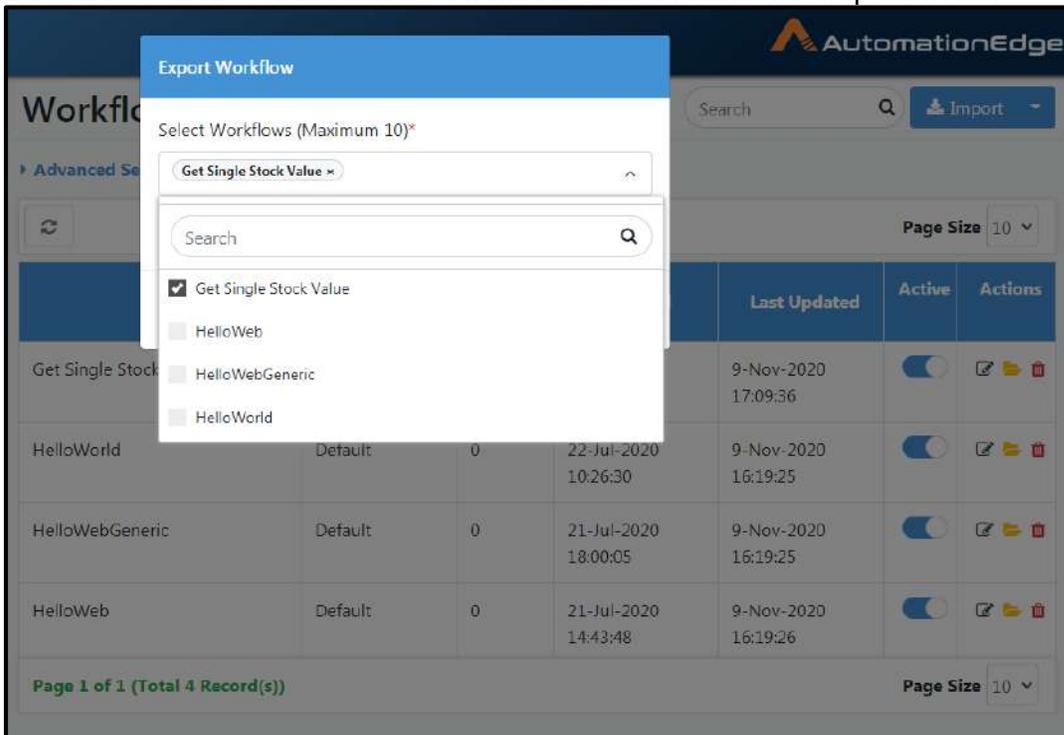
2. Click Workflows menu.
3. Workflows List menu is selected by default.
4. Click arrow next to Import button. Click Export.



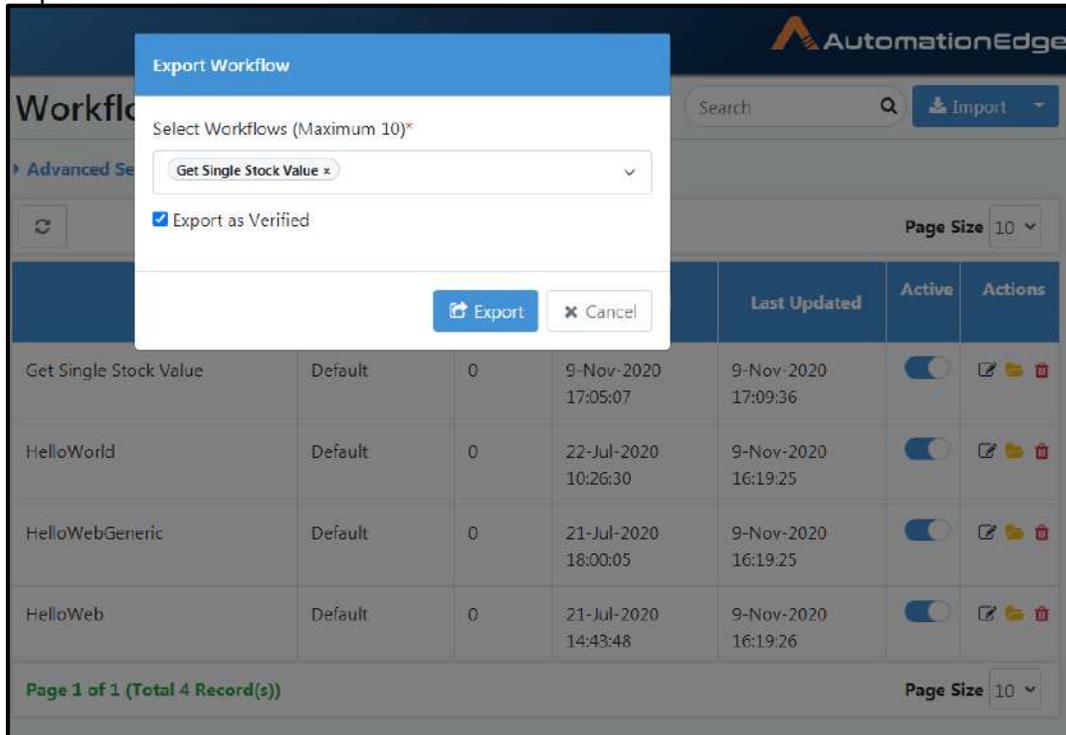
5. Select workflows from the dropdown menu
6. You have the option to enable Export as Verified checkbox. If Exported as verified it will be imported as a verified workflow in Production instance.



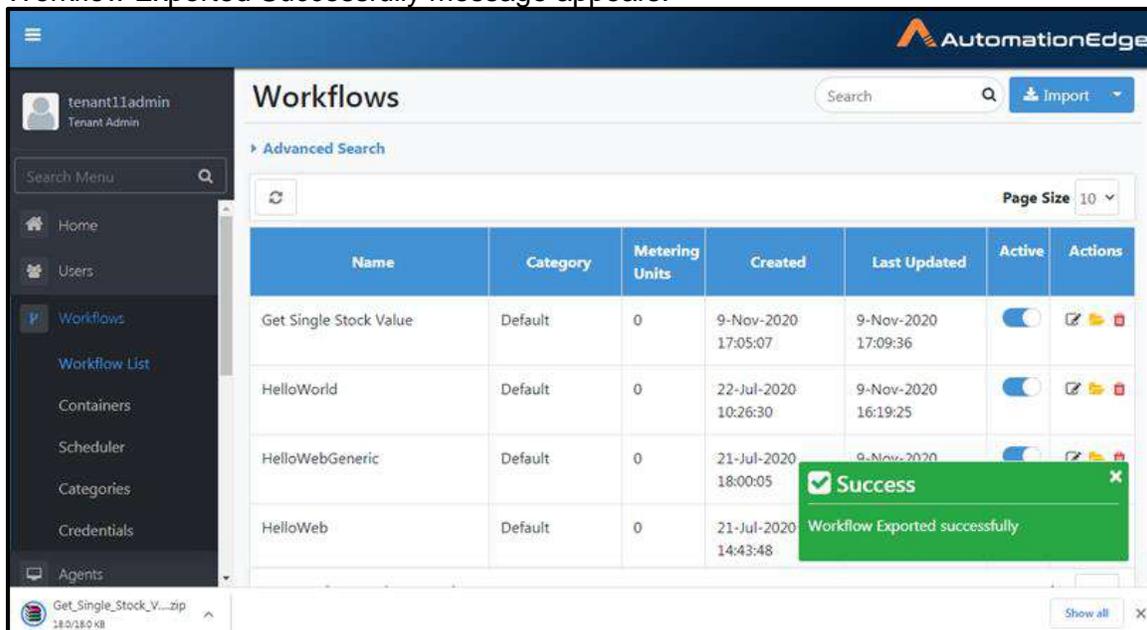
7. Enable checkbox next to HelloWeb from the Select Workflows drop down list.



8. HelloWeb is selected and in this case we have enabled the Export as Verified checkbox. Click Export.



9. Workflow Exported Successfully message appears.



10. This completes the process of exporting workflow from UAT instance.

7.6 Workflows: Import to Production Instance

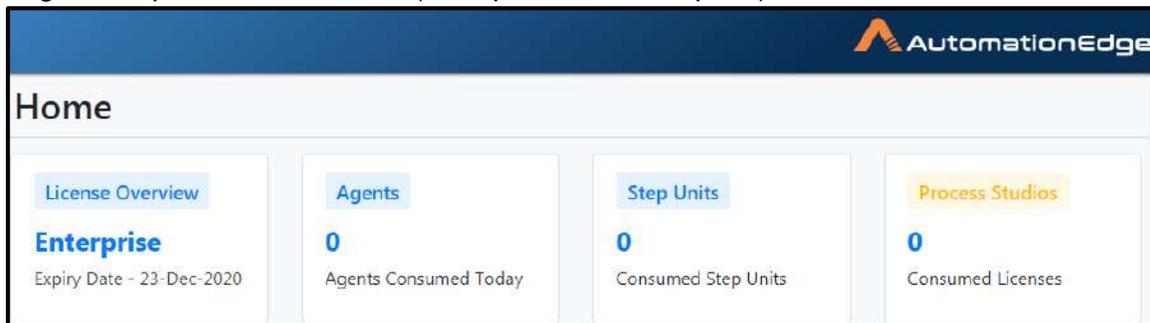
Import button on Workflow menu supports import of Process Studio Processes and Workflows. It can be used to import the zip file exported from Process Studio. Adapter workflows are for backward compatibility only. There is no provision to import Adapter workflows.

Once workflow is found to be working satisfactorily on UAT instance, it is usually exported and then uploaded on Production (Enterprise/Subscription) instance. Usually workflows exported as verified from UAT instance are imported on Production Instance. These workflows are marked as verified Workflows on Production instance.

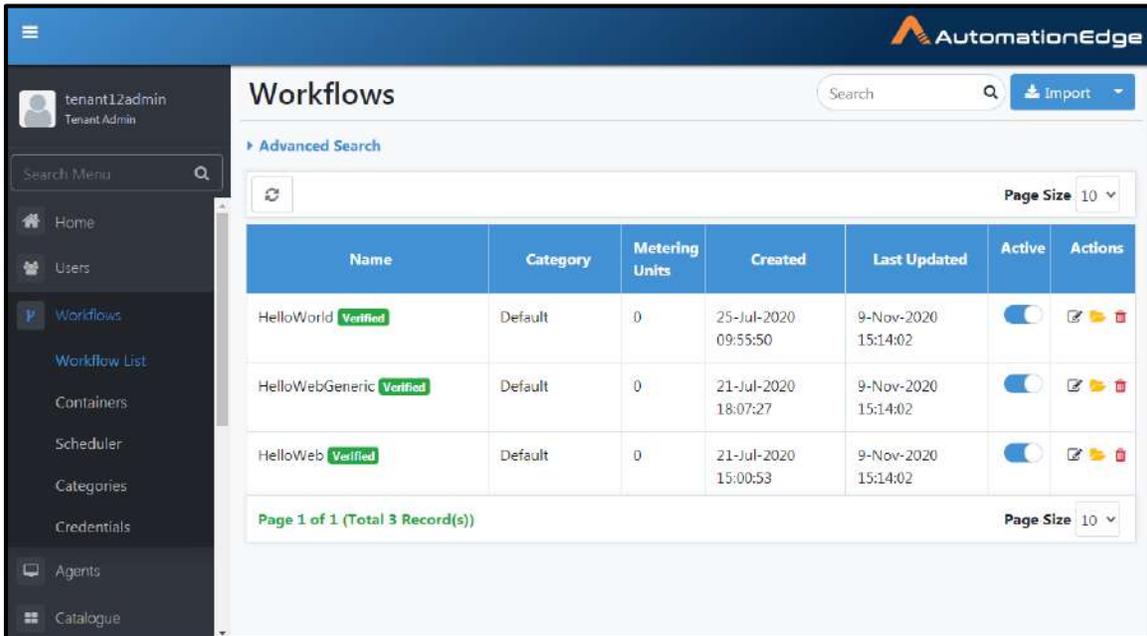
While importing, AE server checks the exported zip file from any AutomationEdge instance is not tampered with, and rejects if not compatible.

Import the workflow as exported from UAT instance. The process to import the workflow is exactly same as Import to Development or UAT instances. Detailed steps to import a workflow have been shown during the process of import to Development instance. Following are the steps to import the workflow Production in short.

1. Logon to a production instance (Enterprise or Subscription)

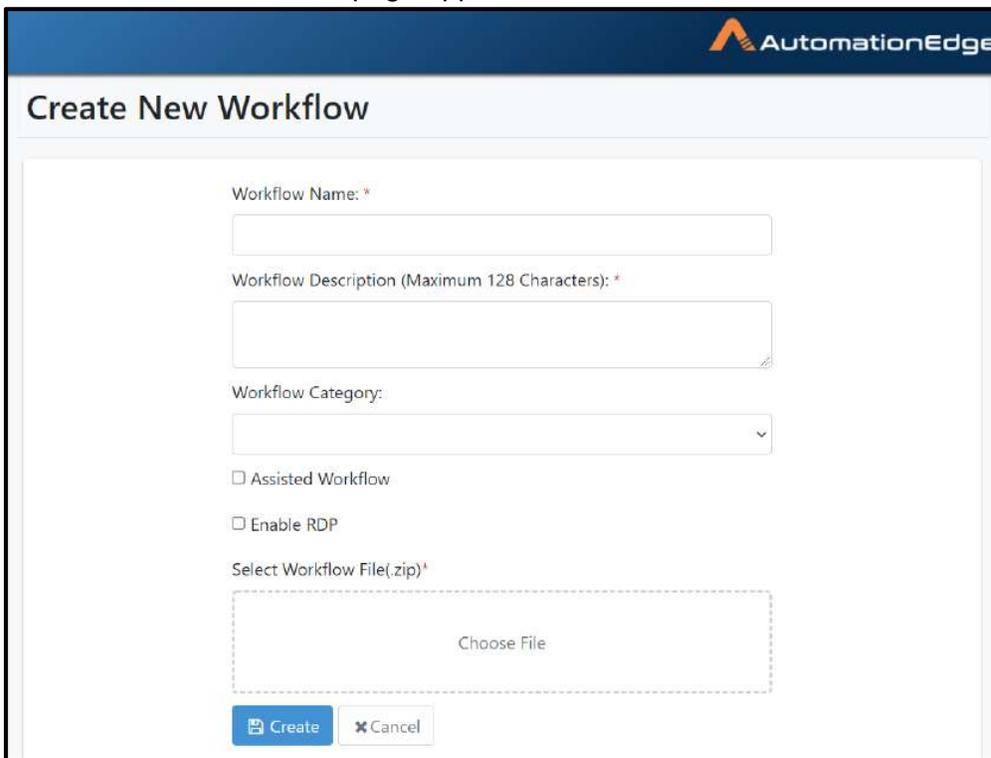


2. Click Workflows menu. Workflows List menu is selected by default.
3. Click arrow next to Import button. Click Import.



Name	Category	Metering Units	Created	Last Updated	Active	Actions
HelloWorld Verified	Default	0	25-Jul-2020 09:55:50	9-Nov-2020 15:14:02	<input checked="" type="checkbox"/>	  
HelloWebGeneric Verified	Default	0	21-Jul-2020 18:07:27	9-Nov-2020 15:14:02	<input checked="" type="checkbox"/>	  
HelloWeb Verified	Default	0	21-Jul-2020 15:00:53	9-Nov-2020 15:14:02	<input checked="" type="checkbox"/>	  

4. The Create New Workflow page appears.



Workflow Name: *

Workflow Description (Maximum 128 Characters): *

Workflow Category:

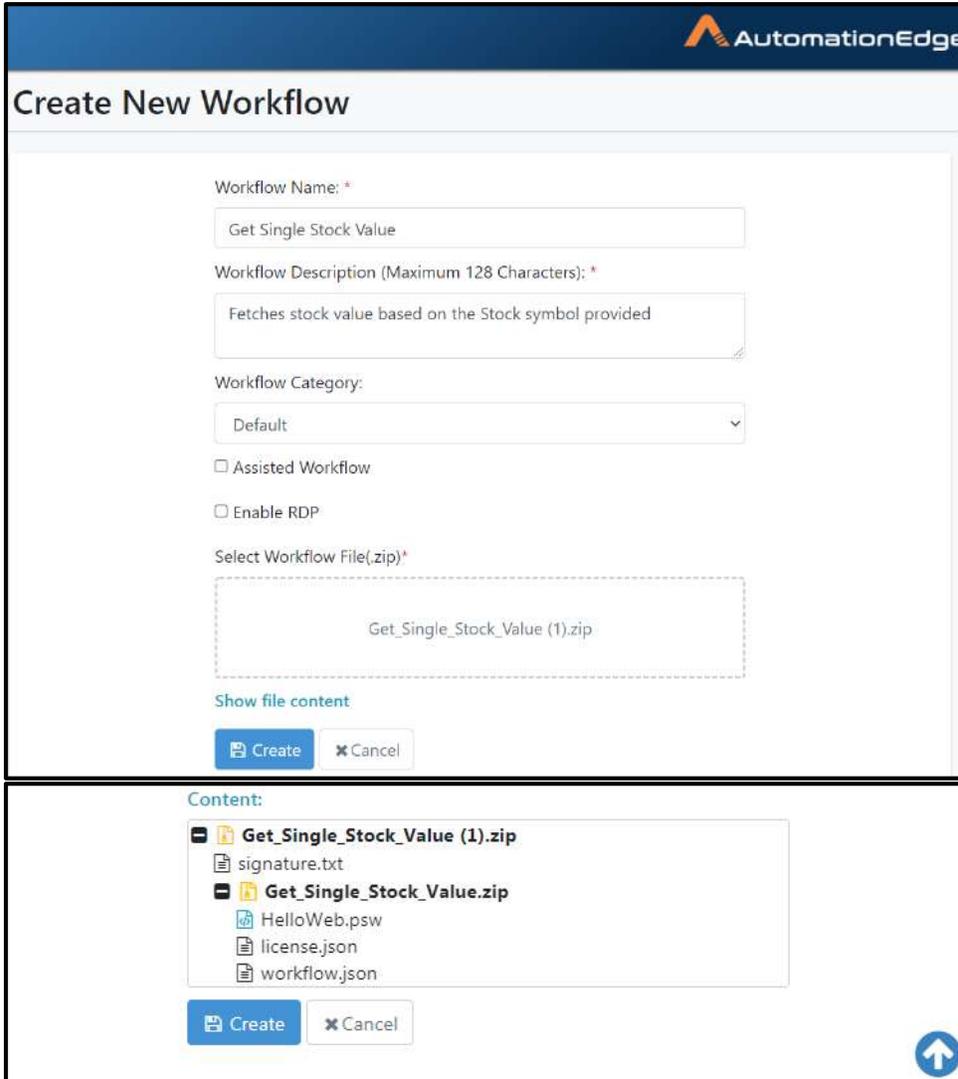
Assisted Workflow

Enable RDP

Select Workflow File(.zip)*

Choose File

5. Provide the details for the new workflow. Browse and upload the zip file exported from UAT server.
6. You may click Show file content link to see the contents of the zip file.
7. Click Create.



Create New Workflow

Workflow Name: *

Get Single Stock Value

Workflow Description (Maximum 128 Characters): *

Fetches stock value based on the Stock symbol provided

Workflow Category:

Default

Assisted Workflow

Enable RDP

Select Workflow File(.zip)*

Get_Single_Stock_Value (1).zip

[Show file content](#)

Content:

- Get_Single_Stock_Value (1).zip
 - signature.txt
 - Get_Single_Stock_Value.zip
 - HelloWeb.psw
 - license.json
 - workflow.json



- The Configure Workflow Details page appears. Basic details are populated by default in the zip file exported from UAT server.

Configure Workflow Details

▼ Basic Details

Workflow Name: **Get Single Stock Value**

Workflow Description (Maximum 128 Characters): *

Fetches stock value based on Stock symbols provided

Workflow Category:

Default

Workflow Icon: 

Assisted Workflow : **false**

Enable Sequential Execution

Enable RDP

Enable Input Attributes

Workflow Priority:

Default

Expected Completion Time(Seconds): *

5

Maximum Completion Time(Seconds): *

20

Cleanup Requests older than(Hours):

36

Manual Execution Time:

5 Minutes

▶ Email Notification Setting

No Configuration Parameters



9. Provide Email Notification Setting details as shown below.

▼ Email Notification Setting

Notify On Workflow Failure

Notify On Exceeding Time Limit

Select Users*:

By Role: Tenant Admin Workflow Admin

By Username:

By Email:

Request Creator

Failure Message:

No Configuration Parameters

10. Once both Basic Details an Email Notification Setting are provided. Click Save.

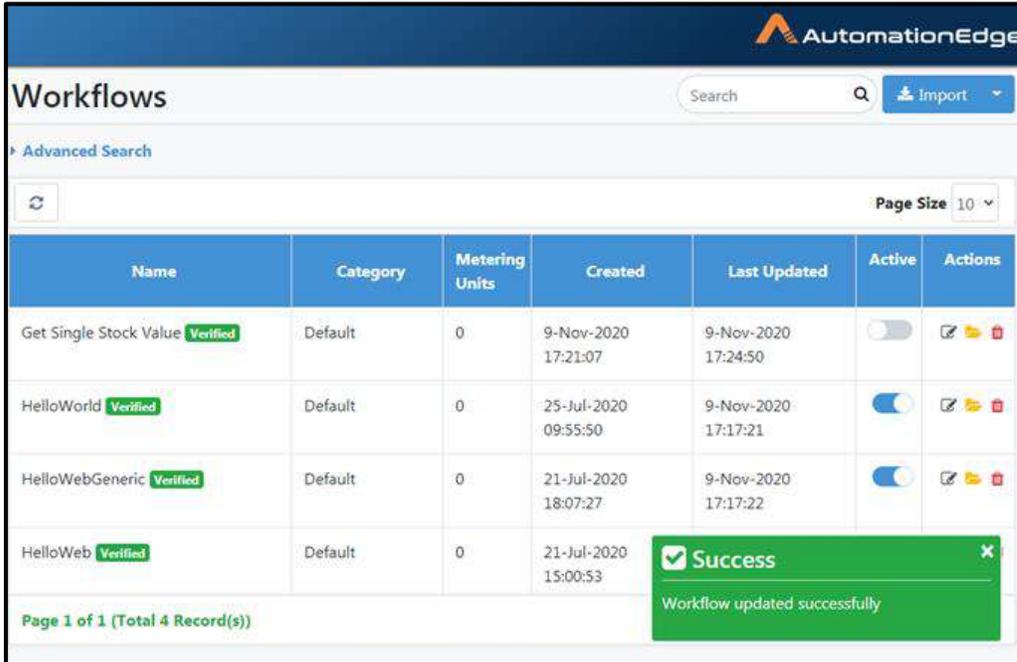
Configure Workflow Details

▶ Basic Details

▶ Email Notification Setting

No Configuration Parameters

11. Workflow added successfully message appears as shown below. Note that the workflow is inactive. Click the toggle button to activate it.



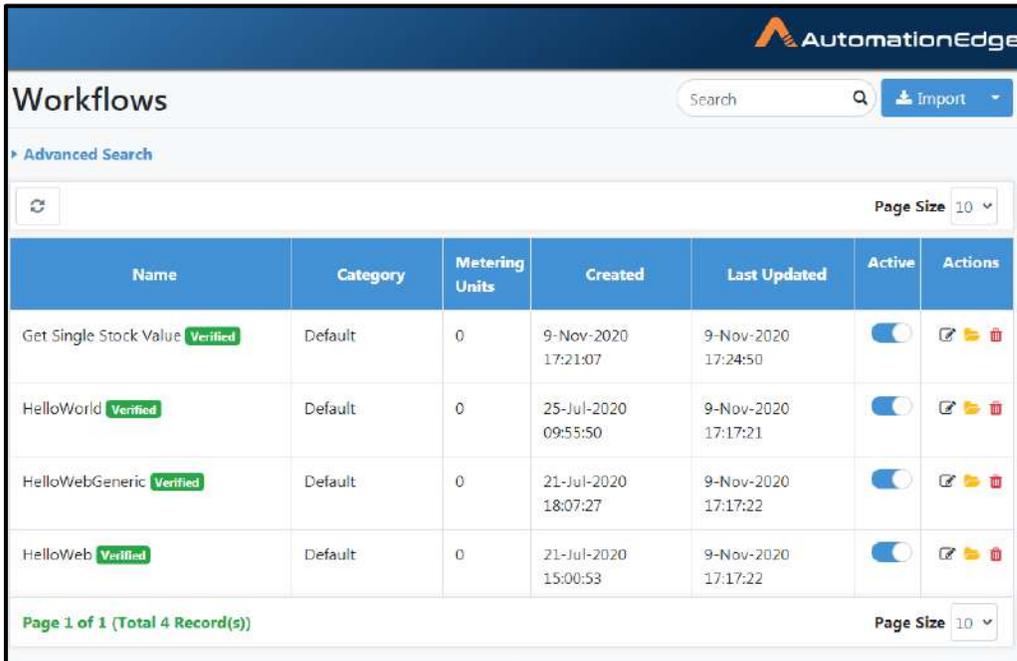
The screenshot shows the AutomationEdge Workflows interface. At the top, there is a search bar and an 'Import' button. Below the search bar is an 'Advanced Search' section with a refresh icon and a 'Page Size' dropdown set to 10. The main content is a table with the following columns: Name, Category, Metering Units, Created, Last Updated, Active, and Actions. The table contains four rows of workflow data. The first row, 'Get Single Stock Value', has its 'Active' toggle switch turned off. A green success message box is overlaid on the table, indicating that the workflow was updated successfully.

Name	Category	Metering Units	Created	Last Updated	Active	Actions
Get Single Stock Value Verified	Default	0	9-Nov-2020 17:21:07	9-Nov-2020 17:24:50	<input type="checkbox"/>	
HelloWorld Verified	Default	0	25-Jul-2020 09:55:50	9-Nov-2020 17:17:21	<input checked="" type="checkbox"/>	
HelloWebGeneric Verified	Default	0	21-Jul-2020 18:07:27	9-Nov-2020 17:17:22	<input checked="" type="checkbox"/>	
HelloWeb Verified	Default	0	21-Jul-2020 15:00:53	9-Nov-2020 17:17:22	<input checked="" type="checkbox"/>	

Page 1 of 1 (Total 4 Record(s))

Success
Workflow updated successfully

12. The newly imported is now active.



The screenshot shows the AutomationEdge Workflows interface, similar to the previous one. The 'Get Single Stock Value' workflow's 'Active' toggle switch is now turned on. The success message is no longer present.

Name	Category	Metering Units	Created	Last Updated	Active	Actions
Get Single Stock Value Verified	Default	0	9-Nov-2020 17:21:07	9-Nov-2020 17:24:50	<input checked="" type="checkbox"/>	
HelloWorld Verified	Default	0	25-Jul-2020 09:55:50	9-Nov-2020 17:17:21	<input checked="" type="checkbox"/>	
HelloWebGeneric Verified	Default	0	21-Jul-2020 18:07:27	9-Nov-2020 17:17:22	<input checked="" type="checkbox"/>	
HelloWeb Verified	Default	0	21-Jul-2020 15:00:53	9-Nov-2020 17:17:22	<input checked="" type="checkbox"/>	

Page 1 of 1 (Total 4 Record(s))

Page Size 10

13. This completes the process of importing the workflow to Production server. The workflow can now be assigned to an Agent and it becomes visible in the catalog. Permissions can be given to users or user groups for the workflow. The workflow can now be executed from the Catalog and monitored from the Requests menu. This is the same as was done for the Development instance in the previous sections.

IV Workspace for Data Processing

8 Project 5: Sales Revenue

8.1 Building your first data processing workflow: HelloWorld

HelloWorld workflow can be summarised in the screenshot below.

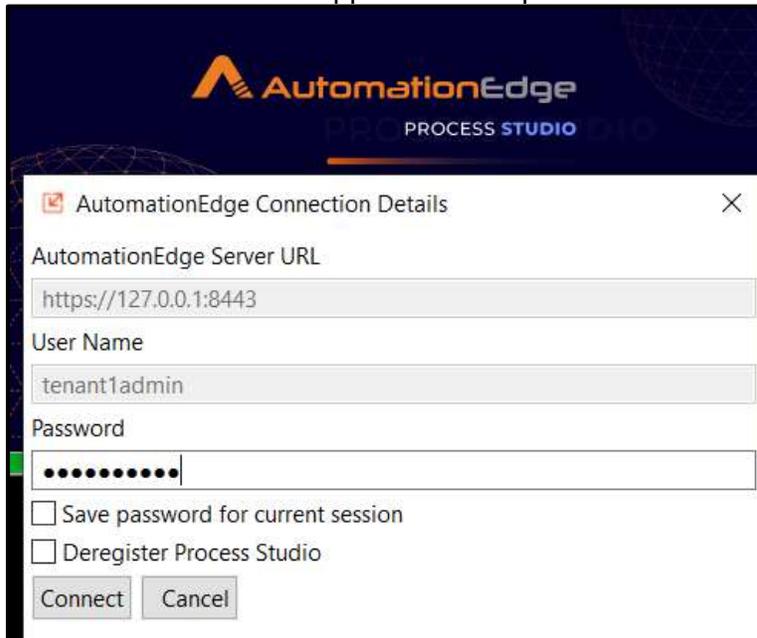


We will be building a workflow which would read sales data from a csv file, do simple processing on it and write the output to an excel file. Create a csv file in an editor with name sales_data.csv.

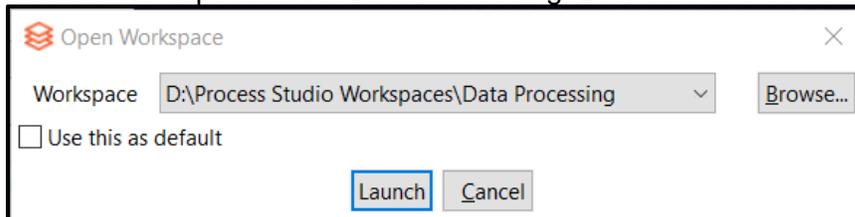
sales_data.csv

```
customer, product, quantity, unitprice
Thunderboltz FC,football,20,1000
Google,basketball,15,900
AutomationEdge,cricket bat,10,500
```

1. Launch Process Studio Application and provide Connection Details to AutomationEdge server.

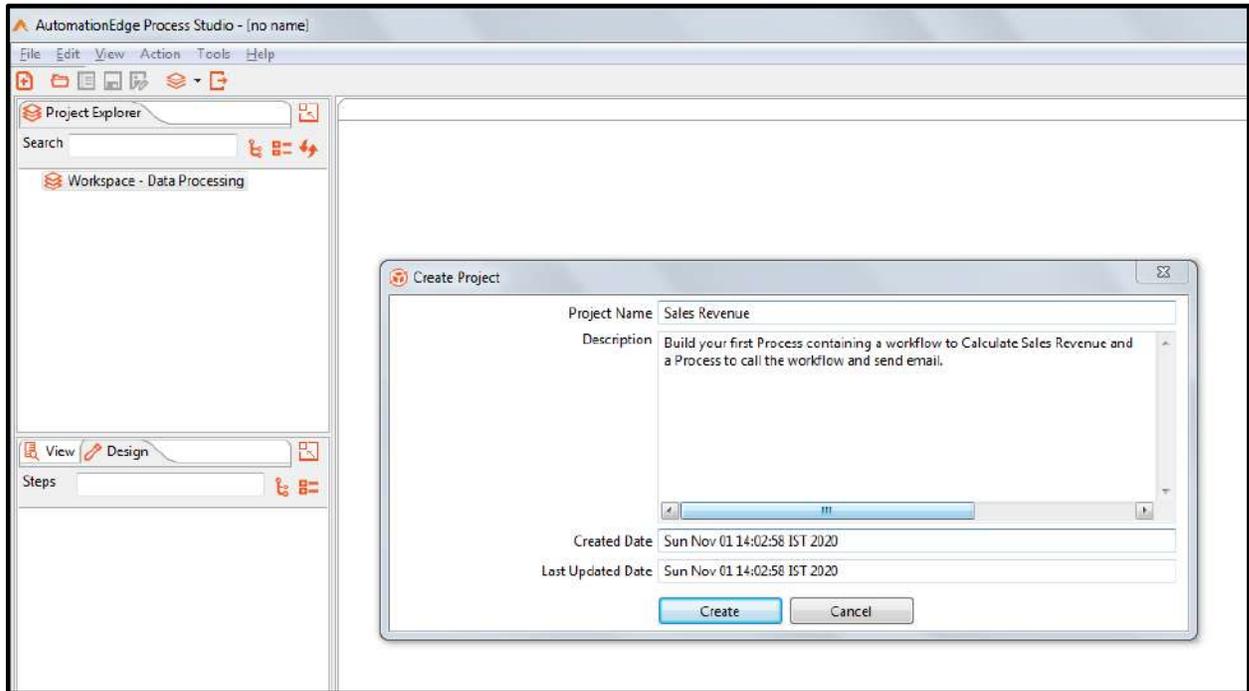


2. The Open Workspace popup appears. We shall create a new Workspace Data Processing. First create a workspace folder Data Processing. Browse the folder for the Workspace.

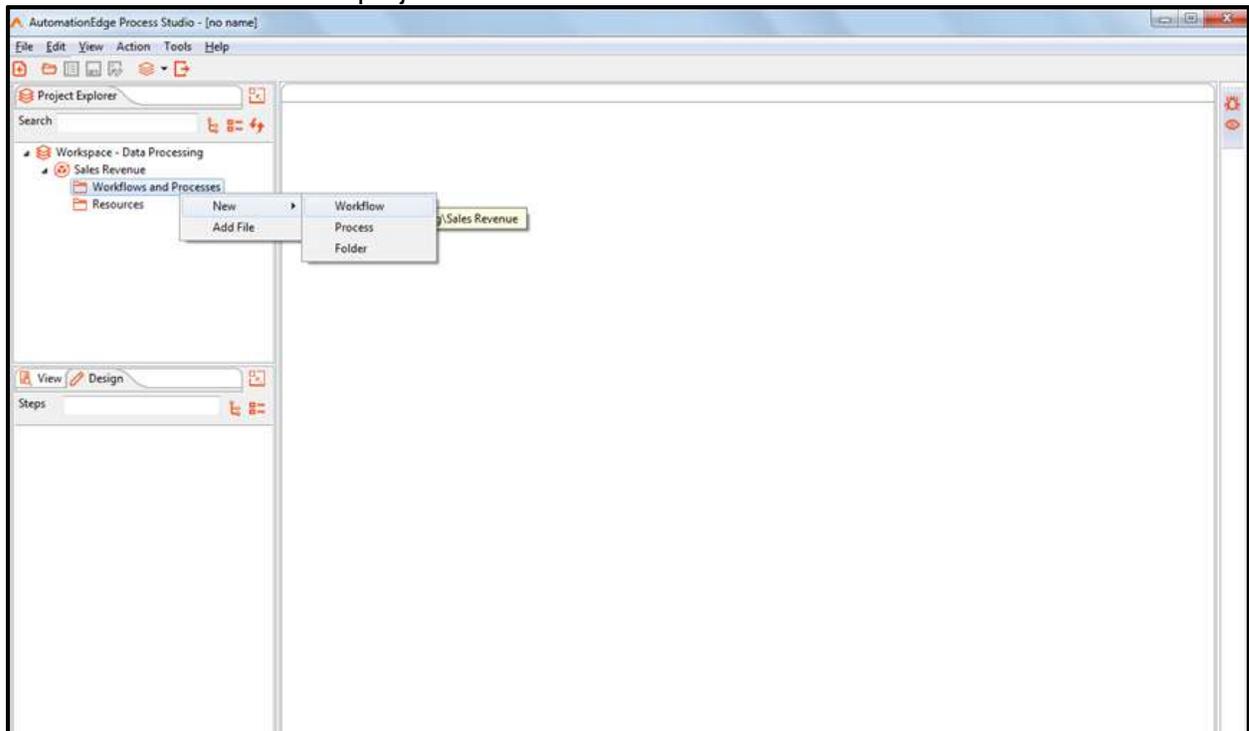


3. The new Workspace Data Processing opens as seen below.

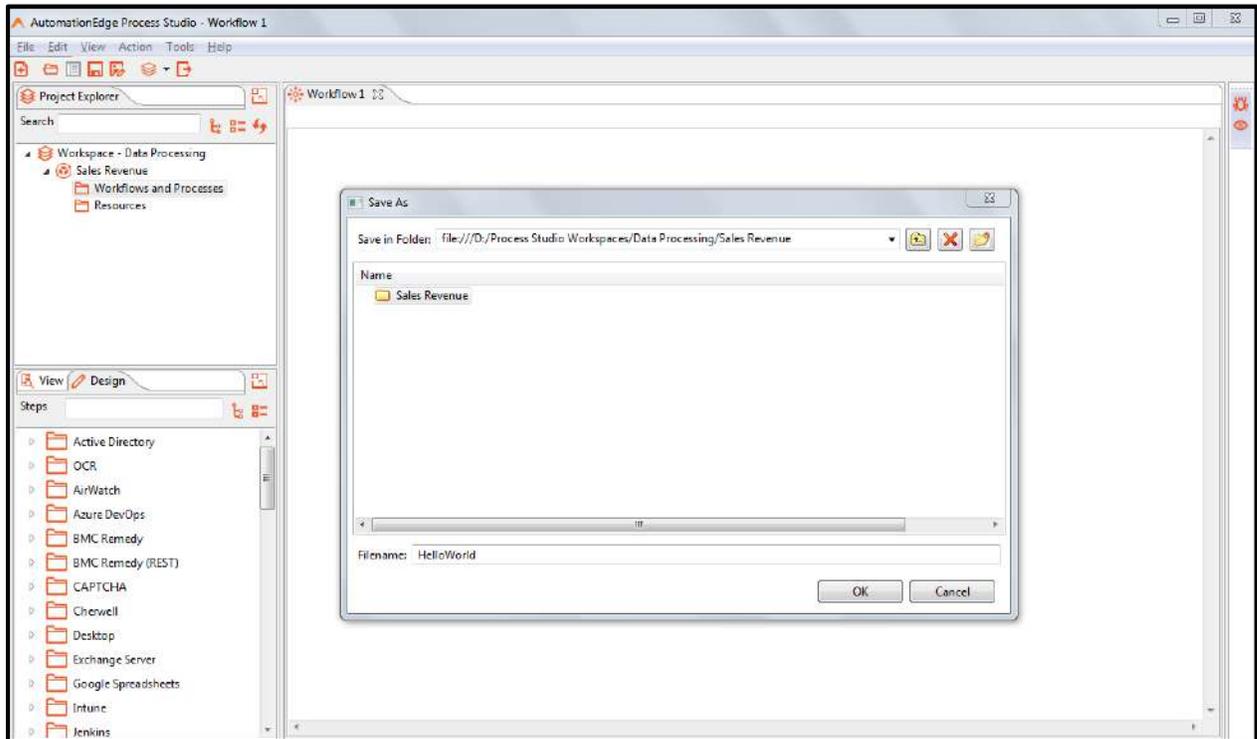
- Right click on Data Processing workspace and select New Project option. Create a new project named Sales Revenue.



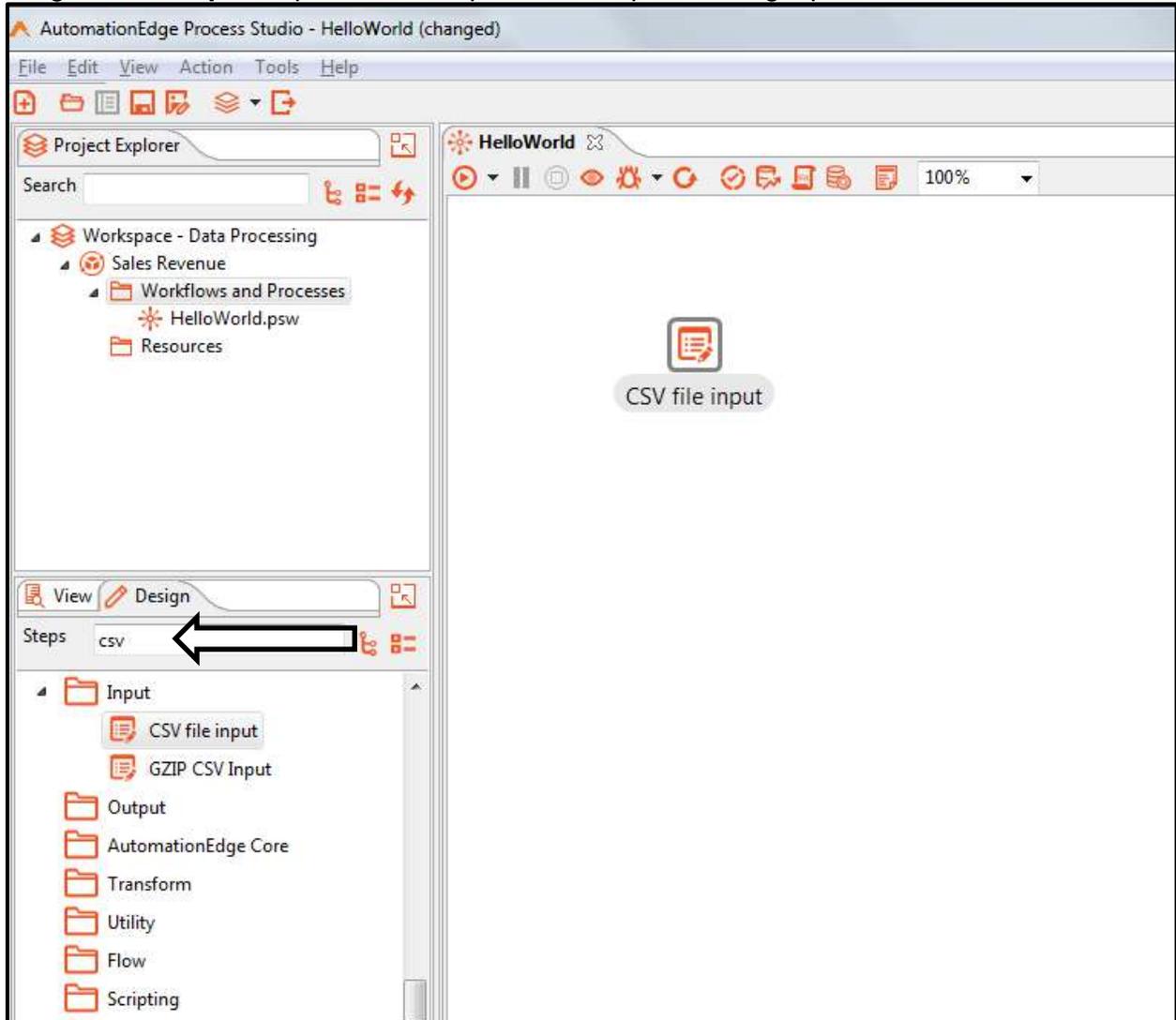
- Next add a workflow to the project.



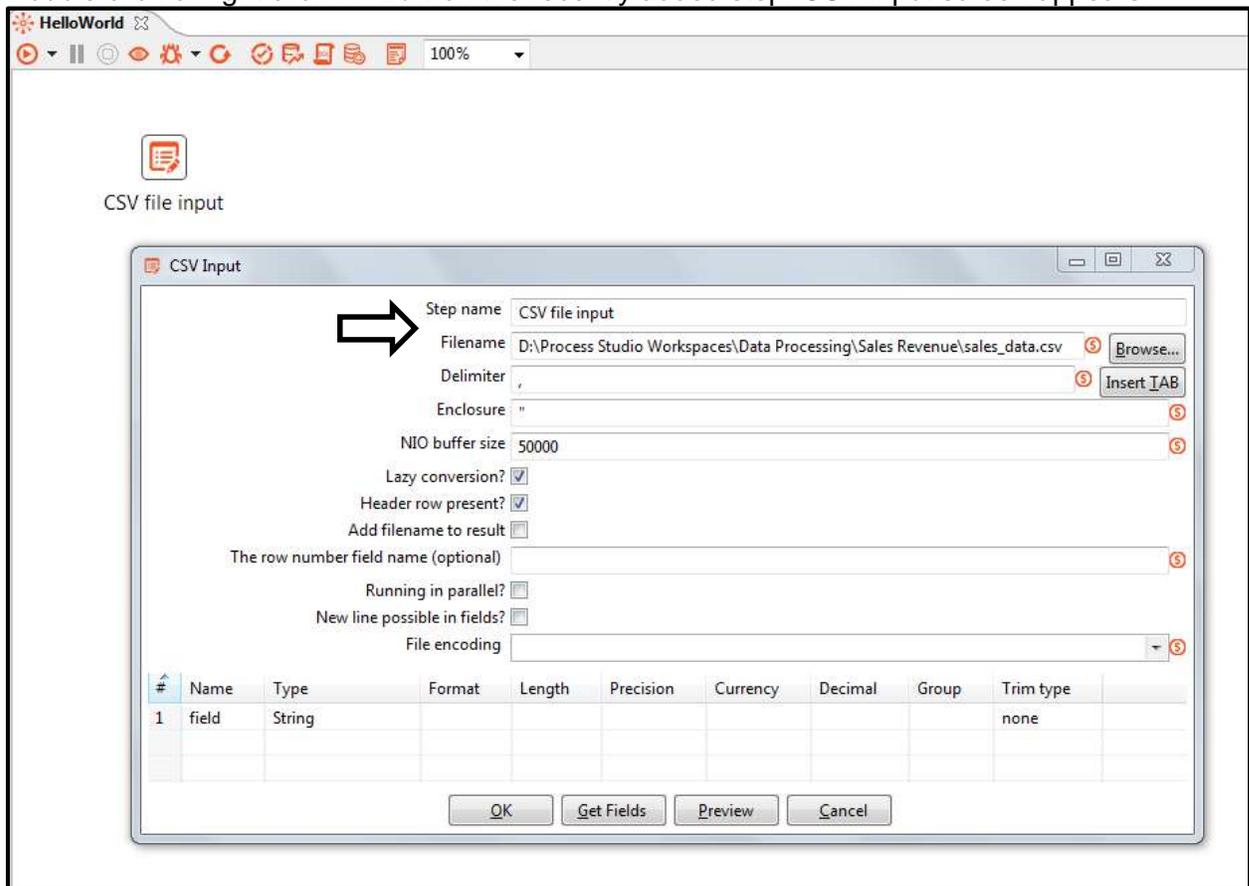
6. Name the workflow **HelloWorld**. Click **OK**.



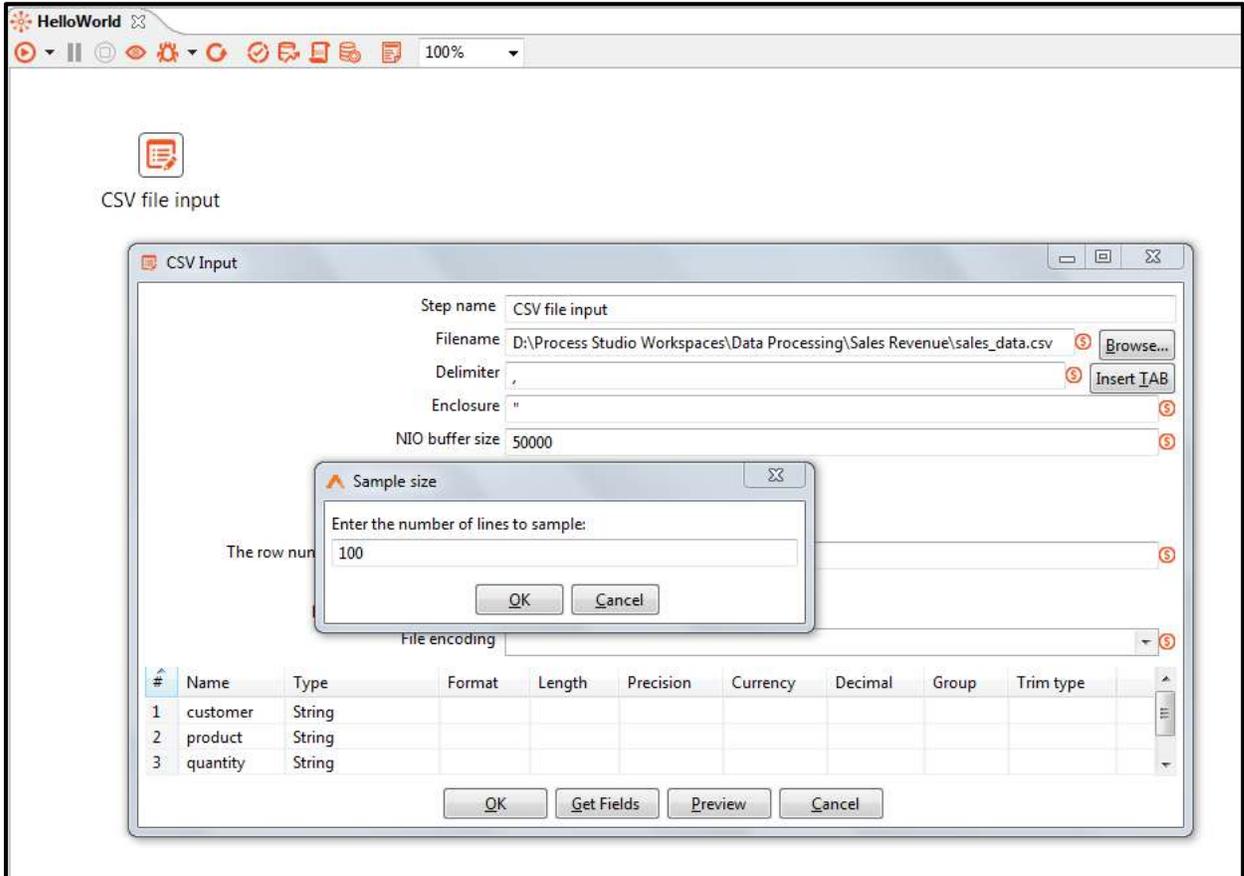
7. Type csv in the text box next to **Steps** in the left pane. All the steps having csv in their name will be displayed.
Drag **CSV file input** step from the left pane and drop it in the right pane.



8. Double click or right click >> **Edit** on the recently added step. CSV input screen appears.

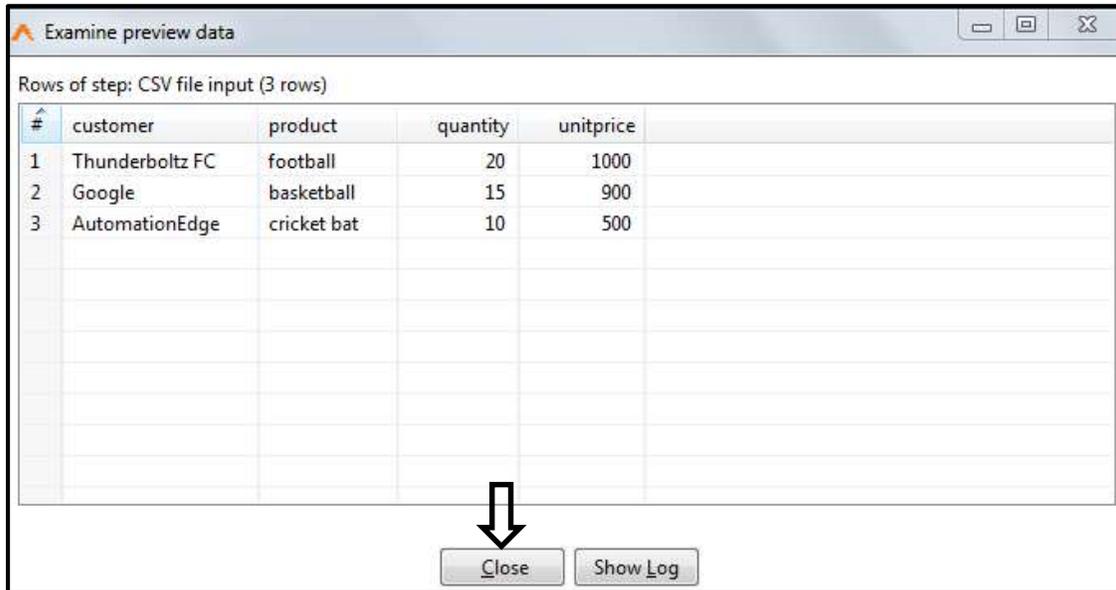


Change the step name to **Read Sales data**, enter complete path of file sales_data.csv in the text box next or **Filename** (or click the browse button, select the file from the location where you have kept sales_data.csv and click open) and click **Get Fields**. A prompt appears to **Enter the number of lines to sample**. The default value is 100. It can be increased or decreased. click **OK**.



Scan results window will be displayed. Details of the fields deduced which it does based on the header row in the csv file (by default **Header row present** checkbox is checked) is displayed. In this example, it deduced four fields. Click **Close** to exit the window.

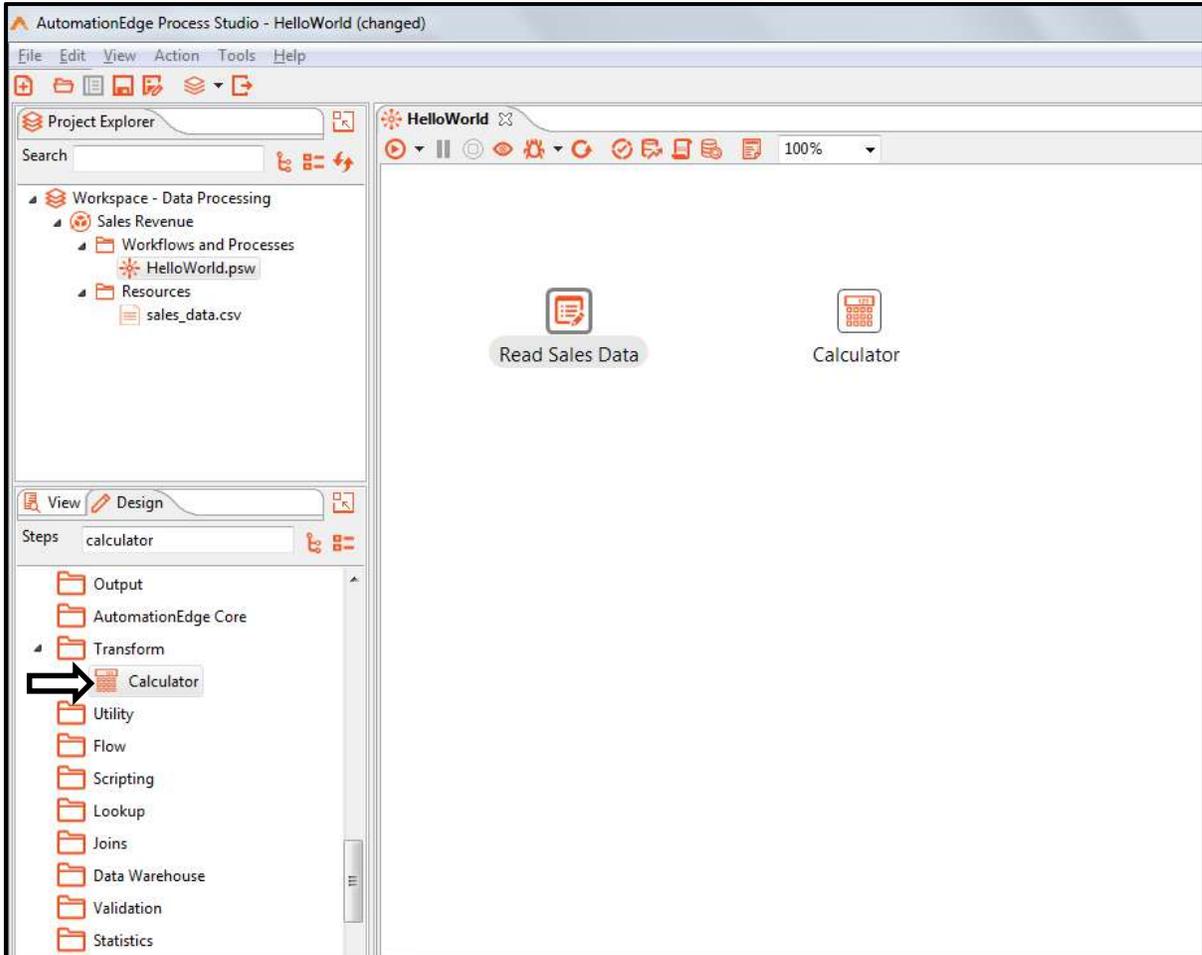
To make sure that the file is getting read correctly, you can click **Preview**. Please do not change any value in the auto generated fields. Click **Close** to exit the window.



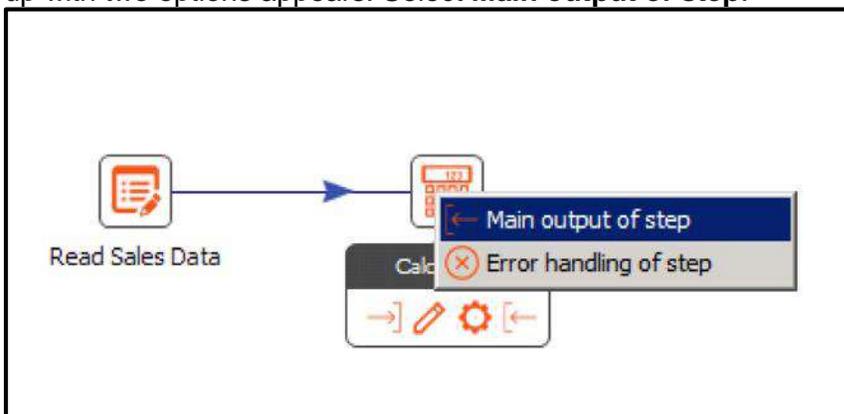
You can save the intermediate workflow by clicking **File >> Save** or using Ctrl+S keyboard shortcut.

 **Note:** A Workflow file is an XML file with .psw extension.

- Clear the word **csv** from the **Steps** search box in explorer tab and click **Transform** in the left pane. All steps listed under category **Transform** will be displayed. You can also do a search for step **Calculator**. Drag and drop **Calculator** step in the right pane.



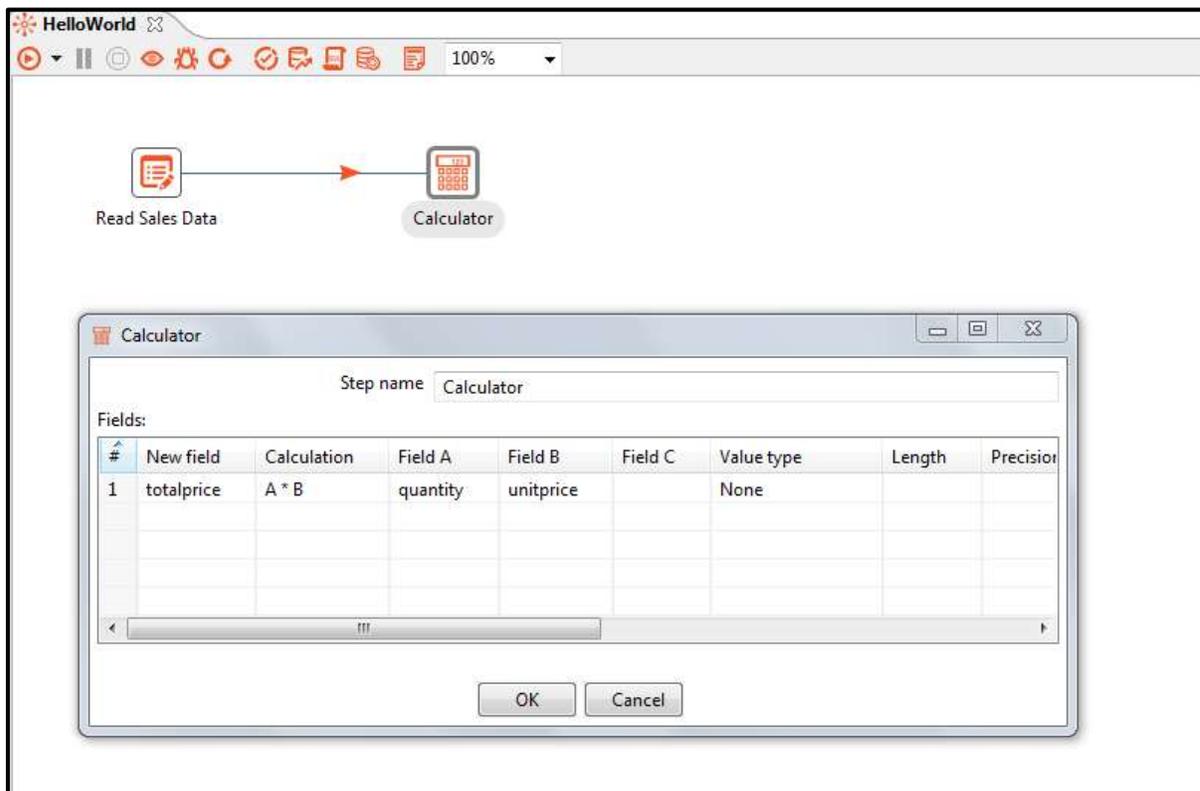
- To connect **Read Sales Data** with **Calculator**, click **Read Sales Data**. Press Shift (on keyboard) with Left Mouse button, drag the mouse over to Calculator and release the mouse button. A pop up with two options appears. Select **Main output of step**.



Read Sales Data and **Calculator** steps are connected now. Connection between two steps is called **hop** which is shown in the user interface as a directed arrow. After connecting steps, workflow would appear as shown in the next screen capture.

11. Double click **Calculator** to change the properties. To set the calculation, click the cell below **Calculation** column and enter the values as shown in the table below:

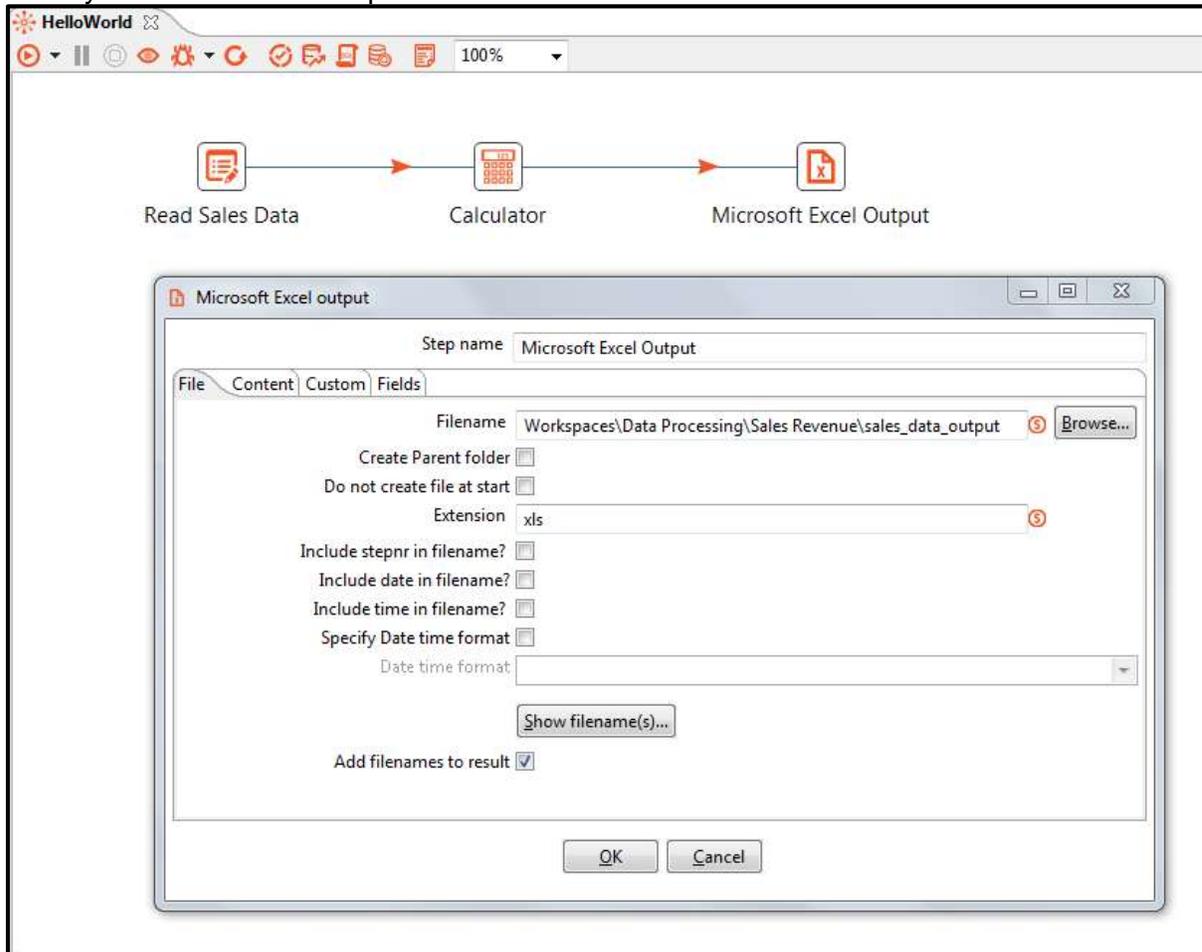
Field	Text / option
New field	totalprice
Calculation	A*B
Field A	quantity
Field B	unitprice



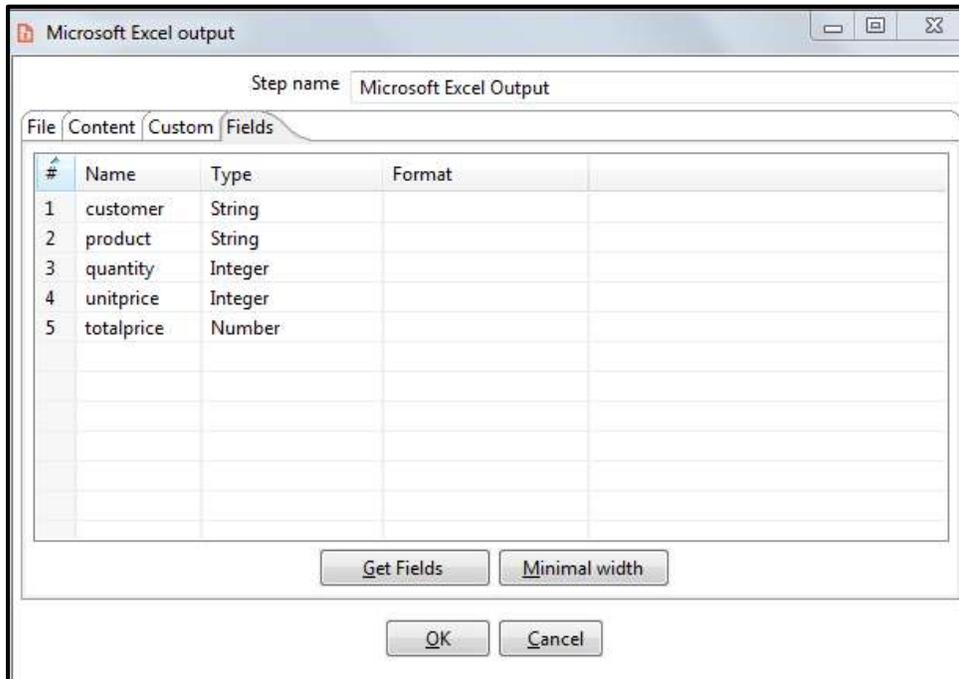
In the process of configuring **Calculator** step, we have generated a new field called **totalprice** which will be computed as product of quantity and unitprice.

- Now let us add one more step. Click **Output** in the left pane to unfold all listed steps. Drag and drop **Microsoft Excel Output** step in the right pane. Connect steps **Calculator** and **Microsoft Excel Output** as by following the same process described earlier.

Double click Microsoft Excel Output in the right pane to edit the configuration. In the **File** tab, provide exact path of the file for **Filename** where you want it to be stored. Do not change any other fields (You can use **Browse...** button to choose Filename, but make sure in this case, file extension is either present in **Filename** field or in **Extension** field). Optionally enable check boxes related to date and time to suffix filename with date and time. This can help uniquely identify each instance of output file.

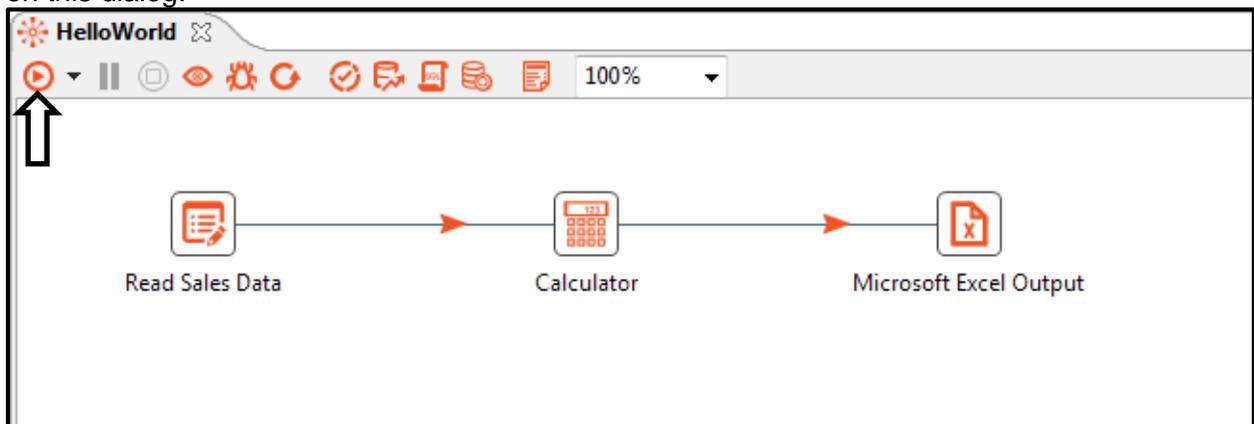


In the **Fields** tab, click **Get Fields** button. Note that, in addition to initial four fields which were obtained from **Read Sales Data** step, additional field **totalprice** which was created as part of **Calculator** step also appears in the list. Click **OK** to save the workflow. The workflow is ready for testing now.

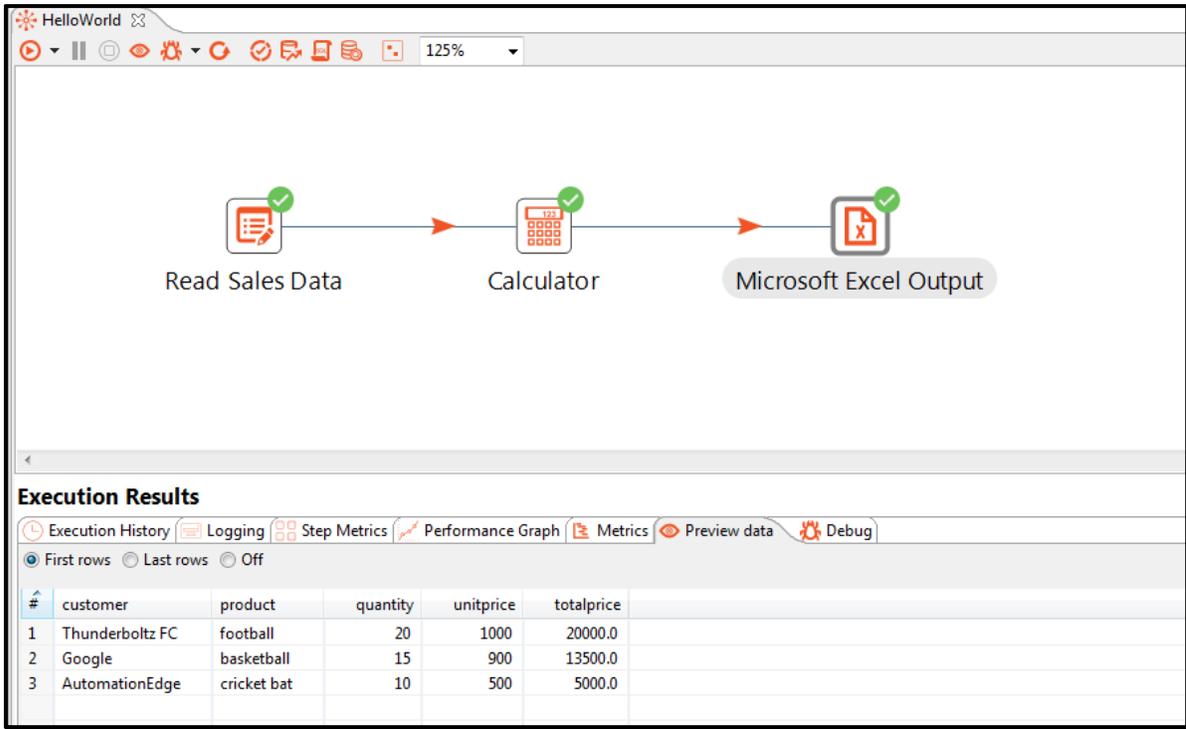


Note: A Workflow always deals with one or more rows each having one or more columns. Every column is termed as a Field. A field has a name and datatype.

13. To run the workflow, click **Run** which is placed exactly below the workflow name in the right pane. You also can execute workflow by clicking **Action >> Run** from the main menu or by simply clicking **F9**. When you run a workflow by clicking **Action >> Run Options** “Run options” dialog appears. For now, we will not configure anything in this dialog and just click **Run** button on this dialog.



After the workflow gets executed, a green mark appears on the top right corner of every successful step. Step metrics of that workflow are displayed below the workflow in the right pane. You can check data in the generated sales_data_output.xls file.



The screenshot displays the Process Studio interface for a workflow named "HelloWorld". The workflow consists of three steps: "Read Sales Data", "Calculator", and "Microsoft Excel Output". Each step has a green checkmark in its top right corner, indicating successful execution. Below the workflow is the "Execution Results" pane, which includes tabs for "Execution History", "Logging", "Step Metrics", "Performance Graph", "Metrics", "Preview data", and "Debug". The "Step Metrics" tab is selected, showing a table of data for the first three rows.

#	customer	product	quantity	unitprice	totalprice
1	Thunderboltz FC	football	20	1000	20000.0
2	Google	basketball	15	900	13500.0
3	AutomationEdge	cricket bat	10	500	5000.0

**Note:**

Process Studio doesn't need to have Microsoft Excel installed to read from or write to an Excel file.

8.1.1 How a workflow runs

When workflow execution starts, all steps run in parallel. Each step has an input **buffer** or a rowset with a default value of 10k rows. So, when a step has read or processed its first 10k rows, it will then fill the buffer. The next step will then start processing those rows while the first step is still reading. You can think of each step as being a worker thread with queues for input and output rows. Each thread processes its rows as they become available. Assume the file we read in **HelloWorld** has 20K rows. In this case, after first 10K rows are read by first step, **Calculator** step will start processing those while first step continues reading next rows. This design increases overall throughput. However, if we were to add a **Sort** step between Read Sales Data and Calculator step as shown in below image, **Sort** step would make sure that **Calculator** step doesn't get any rows to process unless all the source rows have been sorted. This is the case, when a step must see all rows in order to work properly.

8.1.2 Types of steps and start step

There are three basic types of steps:

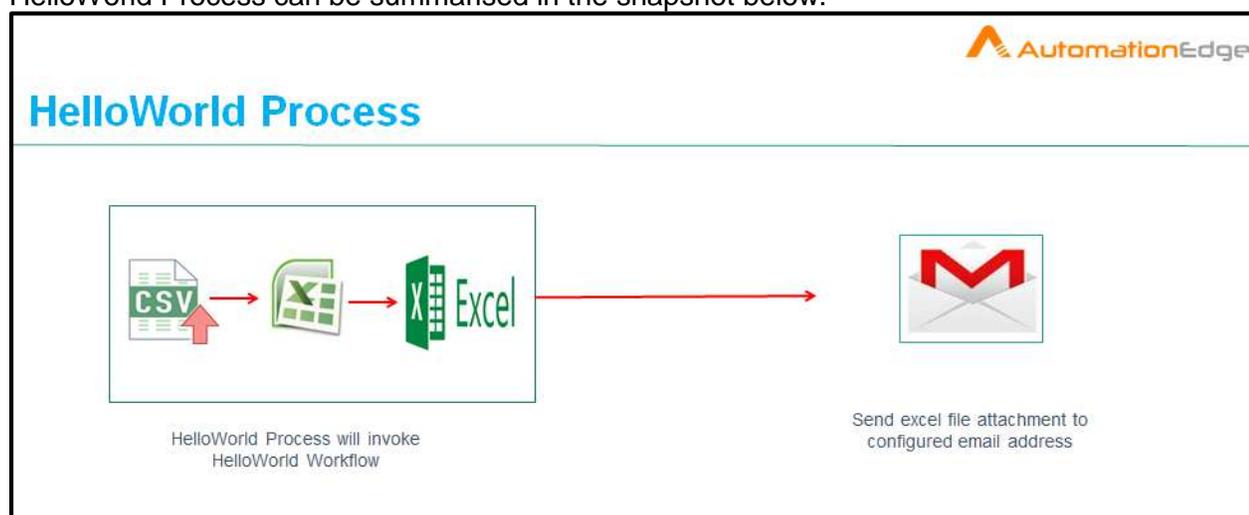
1. **Input step** - This is a step which ingests data into a workflow. It generates rows which the workflow processes (e.g. CSV file input, Table input and so on).
2. **Transformation step** - This is a step which processes data (for example: Calculator, Filter rows, Modified Java Script Value and so on).
3. **Output step** - This is a step which outputs transformed data (for example: Text file output, Table output and so on).

There is no explicit start step in a workflow. As we saw above, all the steps in a workflow execute in parallel. They wait till they receive rows from the previous step. In that sense, a step having no predecessor can be considered as a start step. It also means that for a workflow to do any meaningful work, implicit start step has to be of type **Input Step** which is going to ingest data into the workflow. A workflow can have more than one **Input step**.

8.2 Building your first process

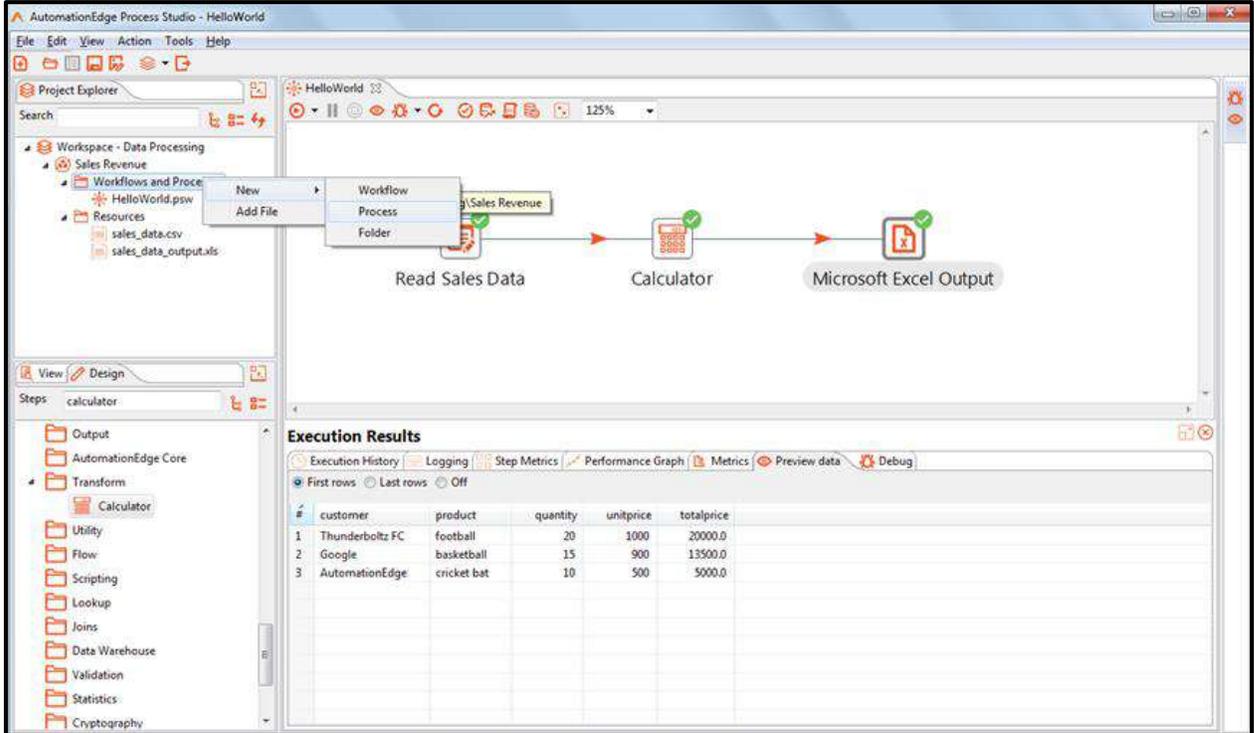
We will now build a process using **HelloWorld** workflow which was built earlier.

HelloWorld Process can be summarised in the snapshot below.



This process will send sales_data_output.xls file which **HelloWorld** workflow generates as an email attachment.

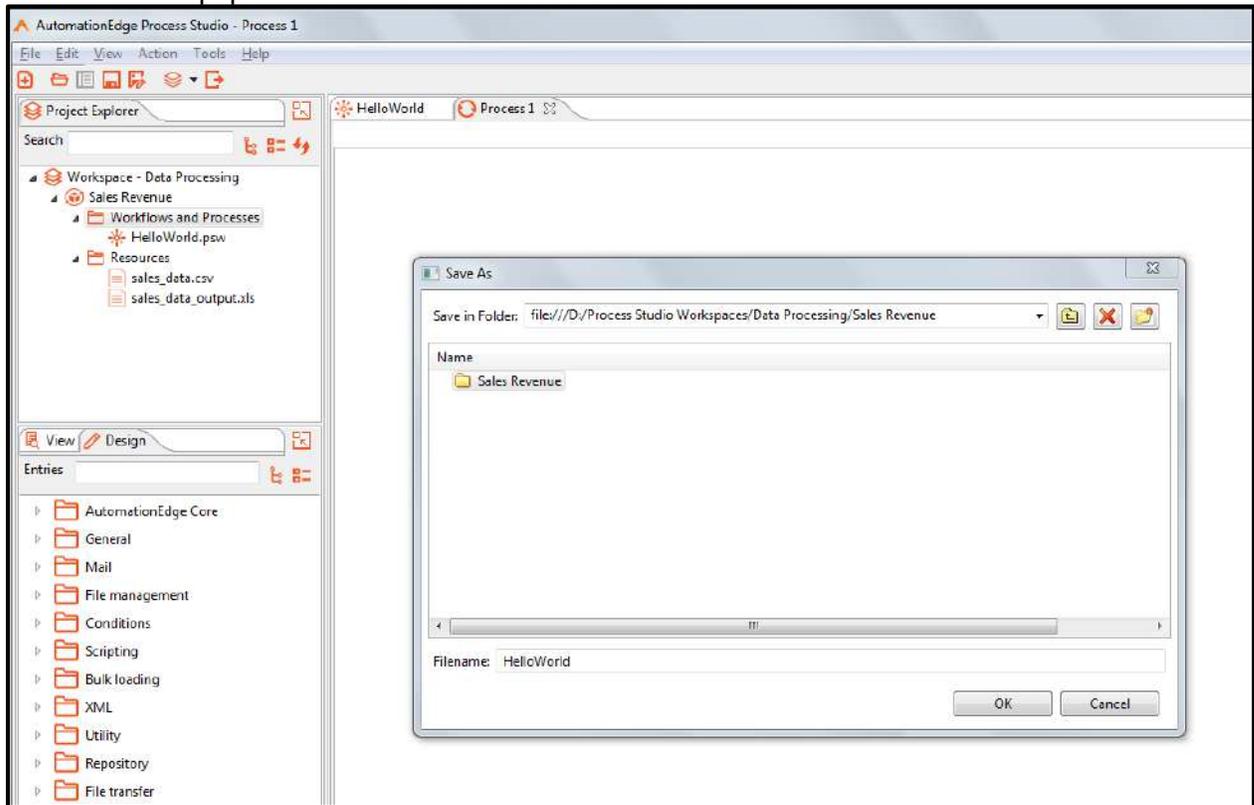
1. right click on Workflows and Processes under Sales Revenue project. Click Process to create a new process.



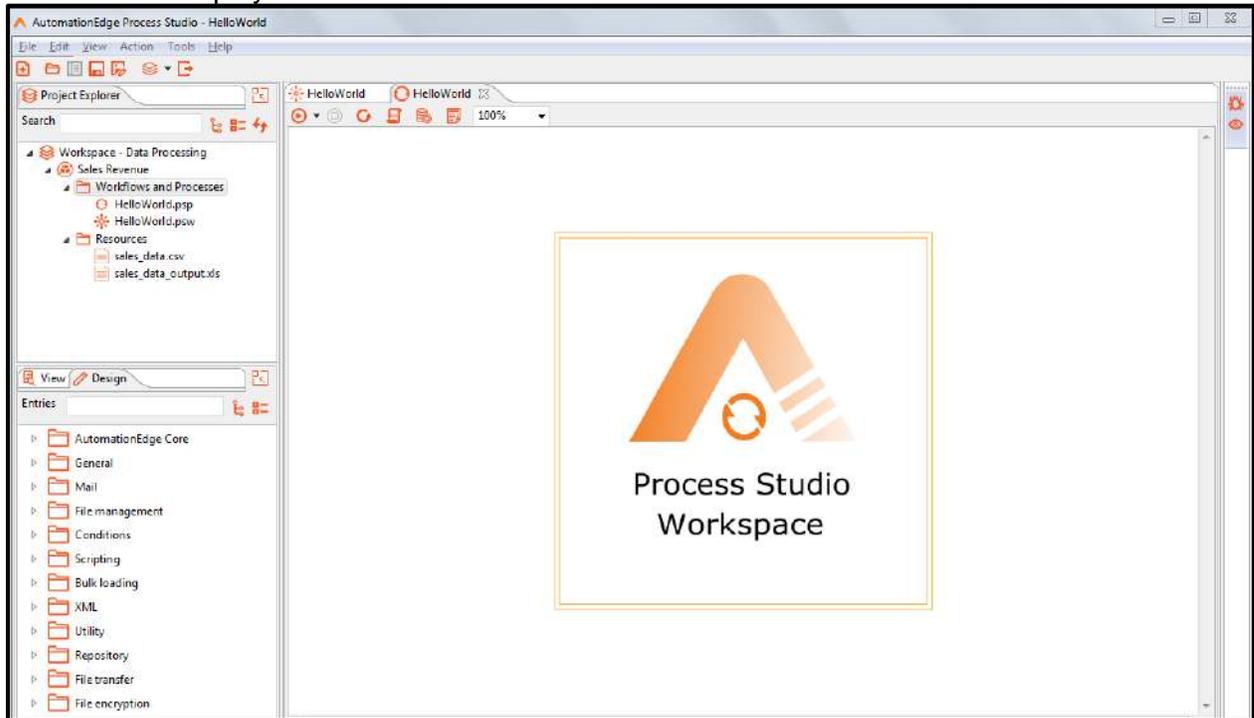
The screenshot displays the AutomationEdge Process Studio interface. The main workspace shows a workflow named 'HelloWorld' with three steps: 'Read Sales Data', 'Calculator', and 'Microsoft Excel Output'. A context menu is open over the 'Read Sales Data' step, showing options: 'New', 'Add File', 'Workflow', 'Process', and 'Folder'. The 'Process' option is selected. The 'Execution Results' panel at the bottom shows a table with the following data:

#	customer	product	quantity	unitprice	totalprice
1	Thunderboltz FC	football	20	1000	20000.0
2	Google	basketball	15	900	13500.0
3	AutomationEdge	cricket bat	10	500	5000.0

2. Name the process HelloWorld. You can name the file as **HelloWorld** and save it at an appropriate location. The default location is the project directory. Process is stored in an XML file with extension psp which stands for Process Studio Process.



3. Type start in the text box next to **Entries** in the left pane. All the steps having **START** in their name will be displayed.



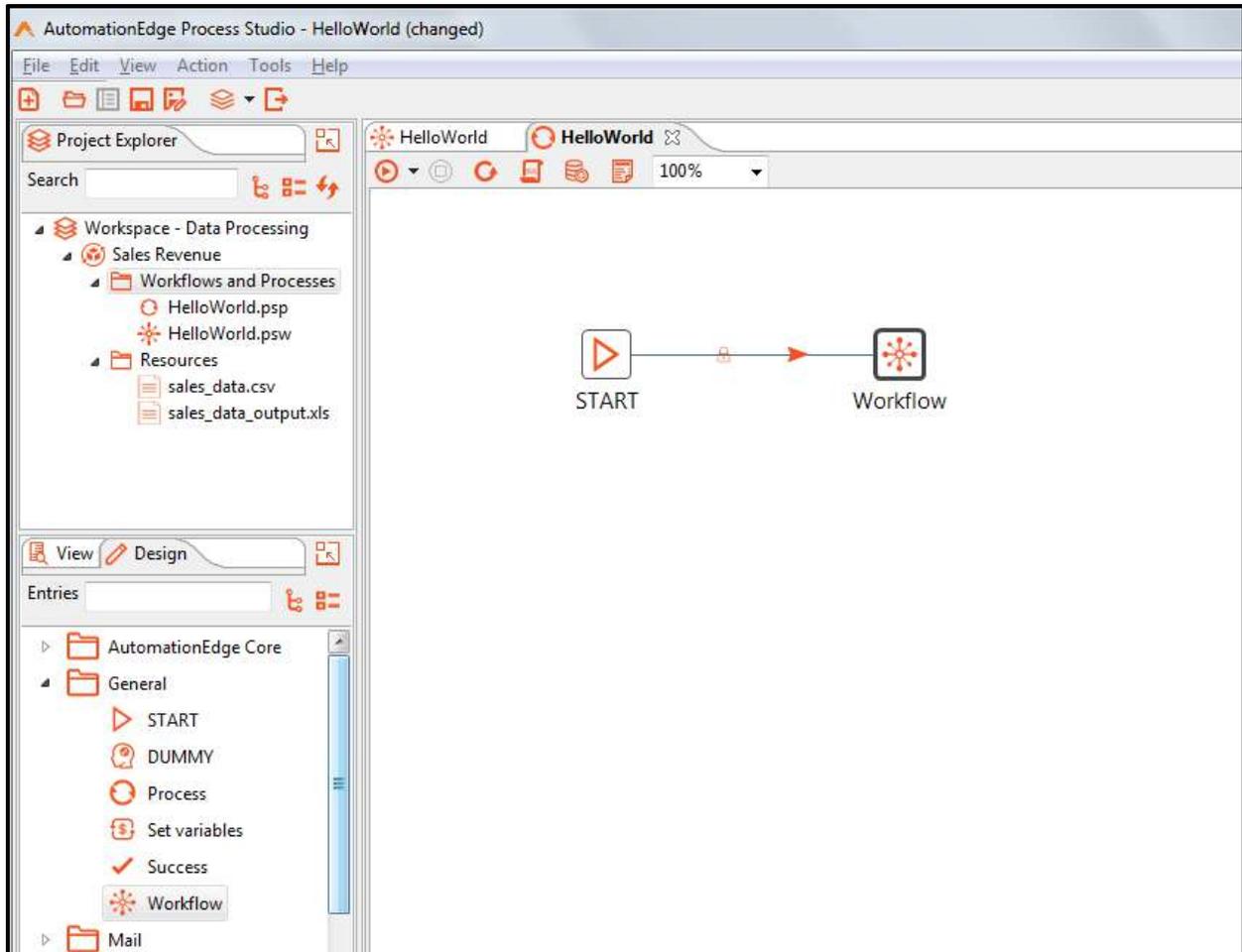
Drag **START** entry from the left and drop it in the right pane. Unlike a workflow, a process can have only one **START** entry and that too at the beginning. It marks the starting point of a process.

4. Now you can save the intermediate process by clicking **File** >> **Save** or using Ctrl+S keyboard shortcut.

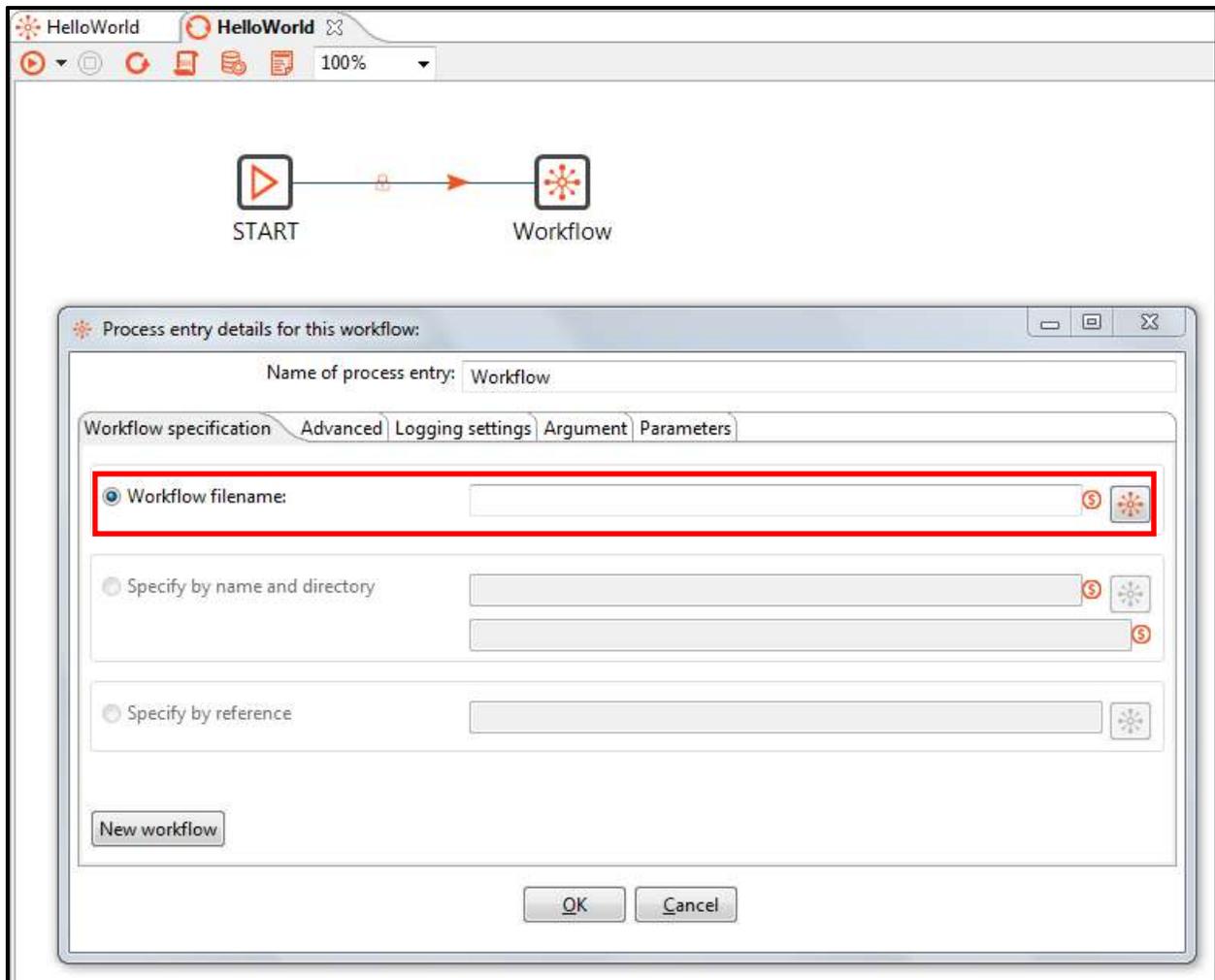
 **Note:** A Process file is an XML file with .psp extension.

 **Note:** When building a process, what you drag and drop on working area is called as an Entry, on the other hand, while building a workflow what you drag and drop on working area is called as a Step. Workflow gets built using steps and process gets built using entries. Connection between two entries in a process and between two steps in a workflow is called as hop in both the cases.

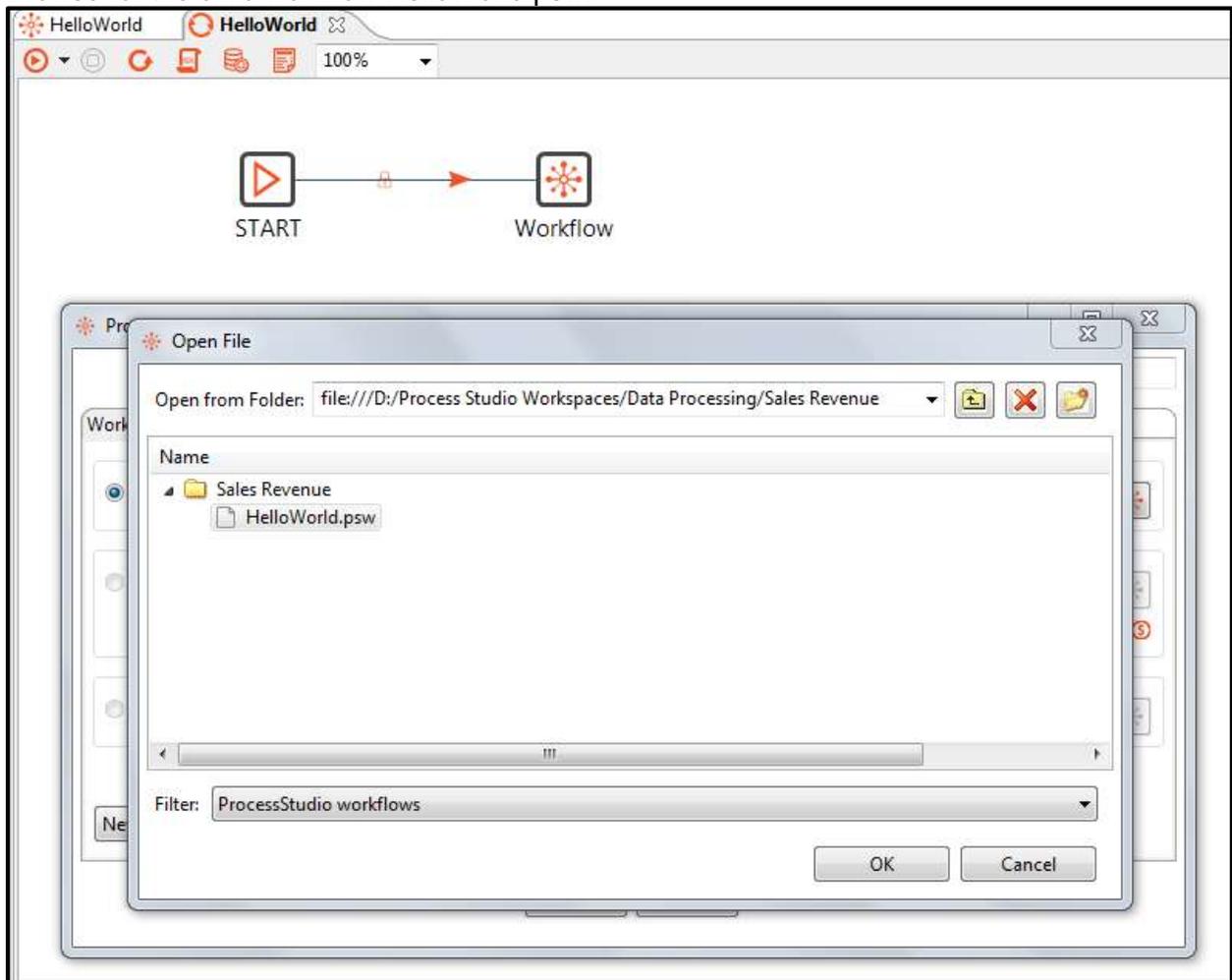
5. Type workflow in the text box next to **Entries** in the left pane. All the steps having **Workflow** in their name will be displayed. Drag **Workflow** entry and drop it in right pane. Connect **START** entry with **Workflow** entry.



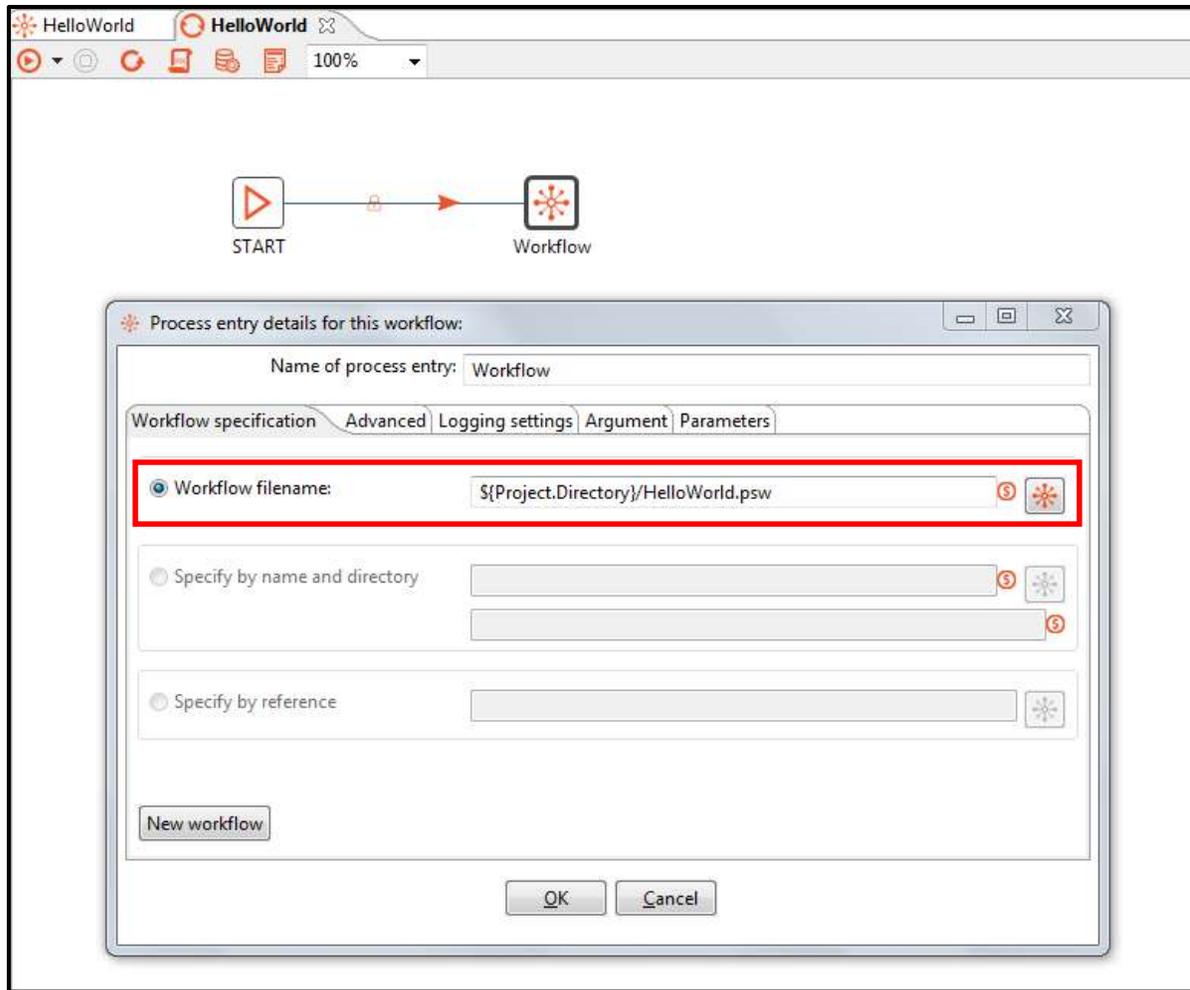
6. Double click **Workflow** entry to configure its properties. **Workflow** entry lets you invoke a workflow. Provide exact path of **HelloWorld.psw** workflow we created earlier in **Workflow filename:** textbox.



7. Browse for the child workflow HelloWorld.psw.

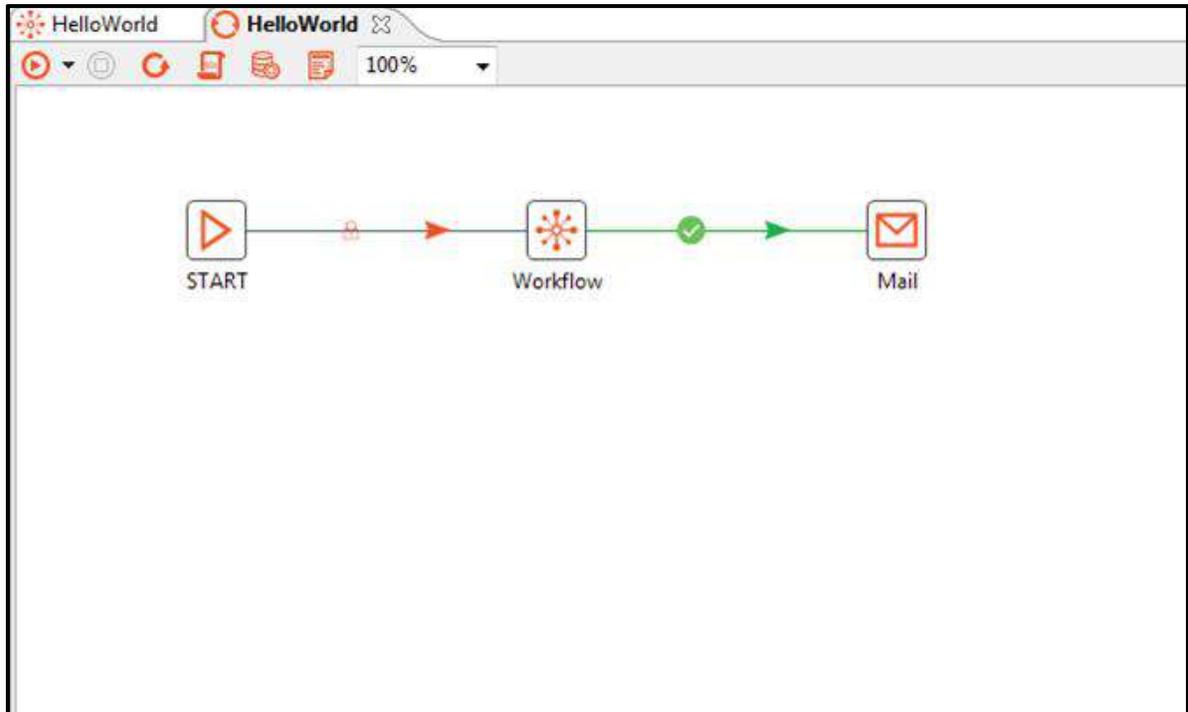


8. HelloWorld.psp and HelloWorld.psw both happen to be in the same top most Sales Revenue Project Directory. Hence, you can provide the path $\${Project.Directory}$.
9. Alternately, if the workflow file you are choosing here happens to be in the same directory as the process file, Process Studio will put the path as $\${Internal.Entry.Current.Directory}/HelloWorld.psw$, where $\${Internal.Entry.Current.Directory}$ is an internal Process Studio variable which indicates directory where file you are working on is saved. As both HelloWorld.psp and HelloWorld.psw files are in the same directory, it makes sense to use this variable as opposed to hardcoding path of HelloWorld.psw file.

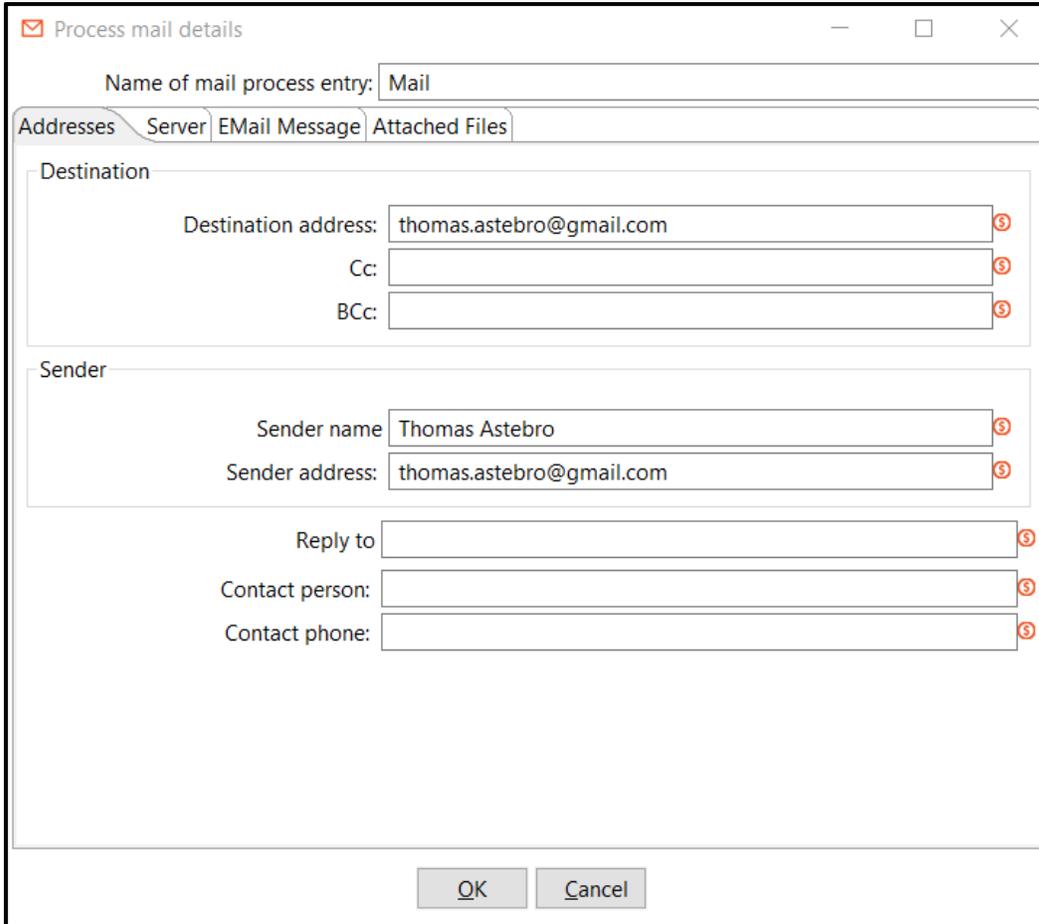
**Note:**

Any string in the form `$${XXX}` has a special meaning for Process Studio. XXX indicates either an internal variable or a parameter passed to a workflow or a process.

10. Type mail in the text box next to **Entries** in the left pane. All the steps having **Mail** in their name will be displayed. Drag **Mail** entry and drop it in right pane. Connect **Workflow** entry with **Mail** entry.



11. Double click **Mail** node to edit its configuration. In **Addresses** tab, provide **Destination address:**, **Sender name** and **Sender address:** as seen below. (Please provide **Destination address:** such that, you can check email received by this address).



Process mail details

Name of mail process entry: Mail

Addresses Server EMail Message Attached Files

Destination

Destination address: thomas.astebro@gmail.com

Cc:

Bcc:

Sender

Sender name: Thomas Astebro

Sender address: thomas.astebro@gmail.com

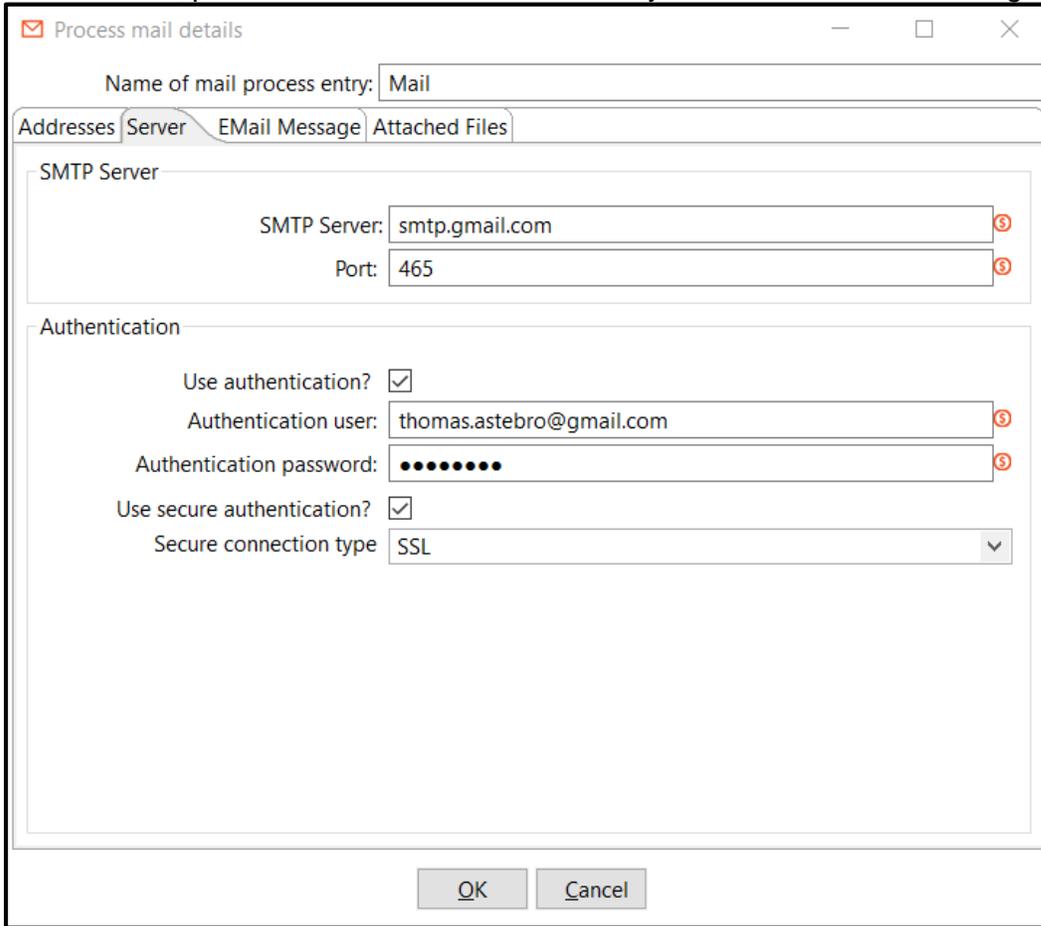
Reply to:

Contact person:

Contact phone:

OK Cancel

In **Server** tab, provide details of the SMTP server you intend to use for sending out email.



The screenshot shows a dialog box titled "Process mail details" with a close button (X) in the top right corner. The "Name of mail process entry" field contains "Mail". Below this is a tabbed interface with three tabs: "Addresses", "Server" (which is selected), and "EMail Message" and "Attached Files".

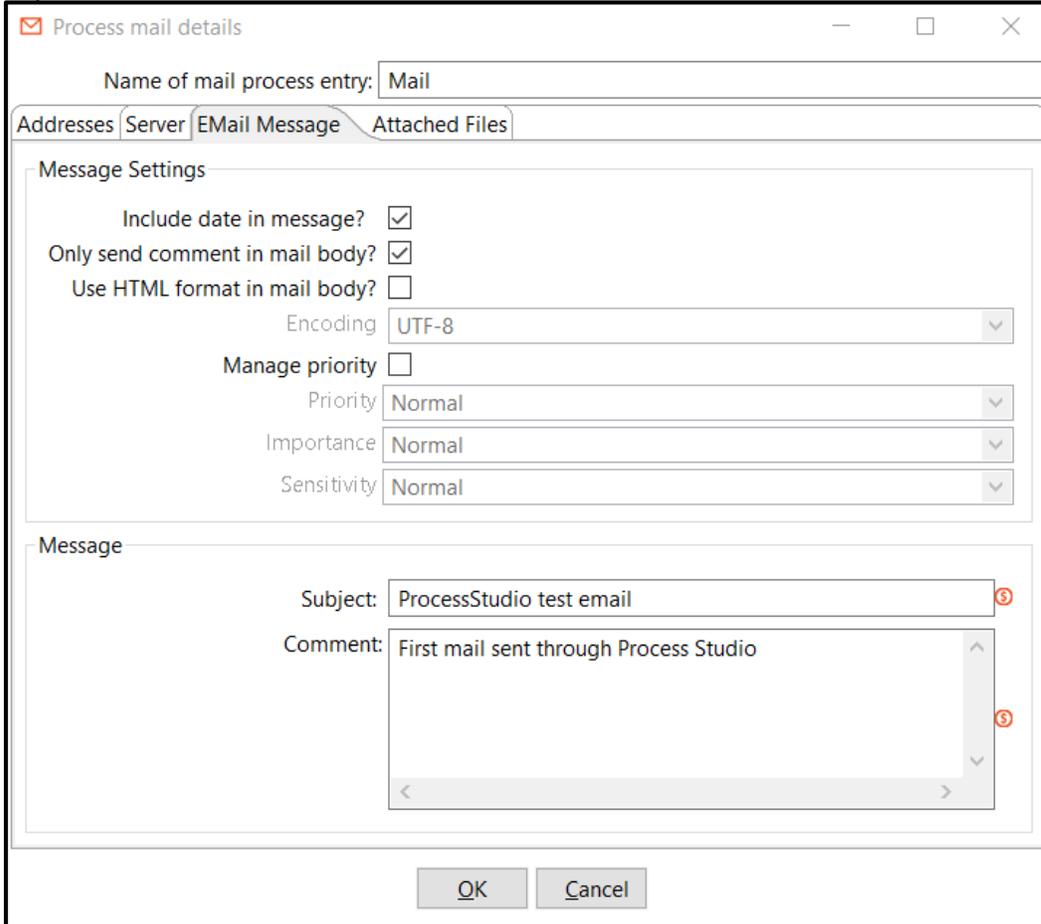
Under the "Server" tab, there is a section for "SMTP Server" with two input fields: "SMTP Server" containing "smtp.gmail.com" and "Port" containing "465".

Below the SMTP Server section is an "Authentication" section with the following fields:

- "Use authentication?" with a checked checkbox.
- "Authentication user:" containing "thomas.astebro@gmail.com".
- "Authentication password:" containing a series of dots.
- "Use secure authentication?" with a checked checkbox.
- "Secure connection type" with a dropdown menu set to "SSL".

At the bottom of the dialog box are "OK" and "Cancel" buttons.

In **Email Message** tab, check **Include date in the message?** and **Only send comment in mail body?**. Provide appropriate Subject and Comment (basically email body) as per the screen capture below.



Process mail details

Name of mail process entry: Mail

Addresses Server **Email Message** Attached Files

Message Settings

Include date in message?

Only send comment in mail body?

Use HTML format in mail body?

Encoding UTF-8

Manage priority

Priority Normal

Importance Normal

Sensitivity Normal

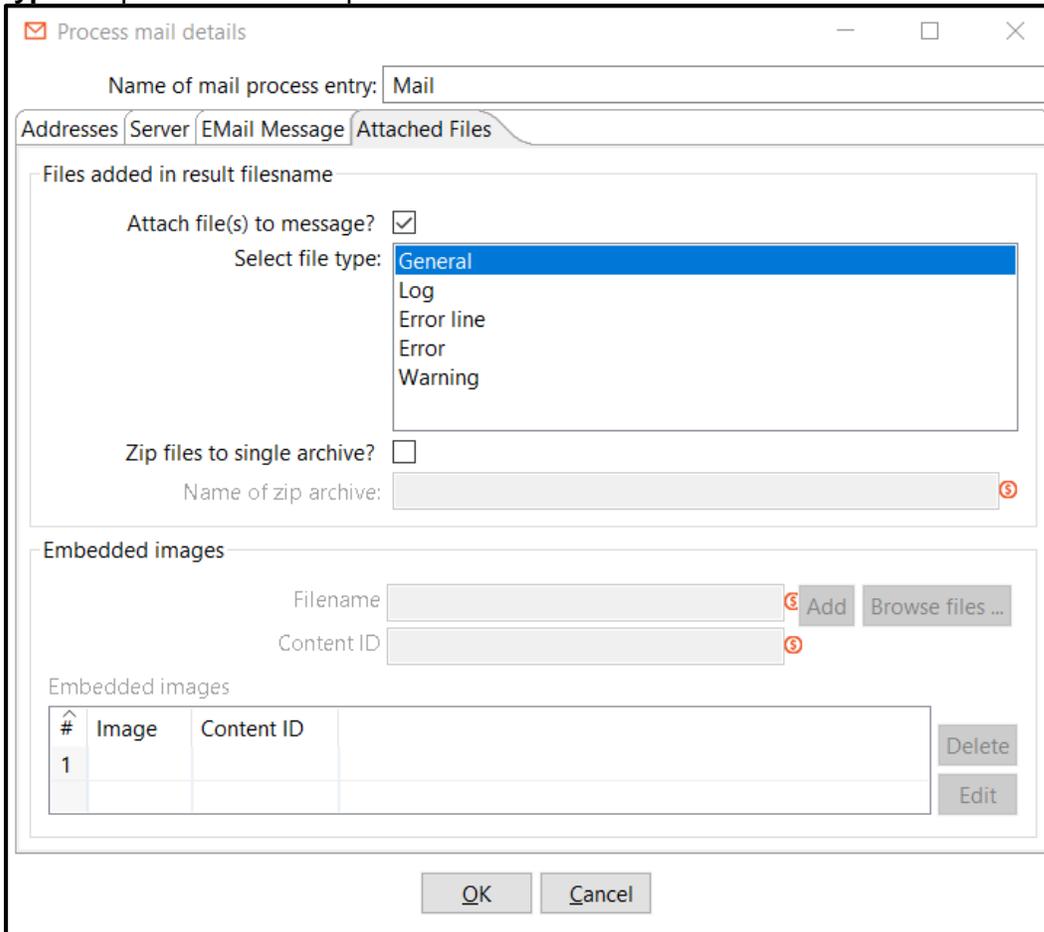
Message

Subject: ProcessStudio test email

Comment: First mail sent through Process Studio

OK Cancel

In **Attached Files** tab, check **Attach file(s) to message?** and select **General** under **Select file type:** as per the screen capture below.



Process mail details

Name of mail process entry: Mail

Addresses Server EMail Message Attached Files

Files added in result filename

Attach file(s) to message?

Select file type: General
Log
Error line
Error
Warning

Zip files to single archive?

Name of zip archive:

Embedded images

Filename Add Browse files ...

Content ID

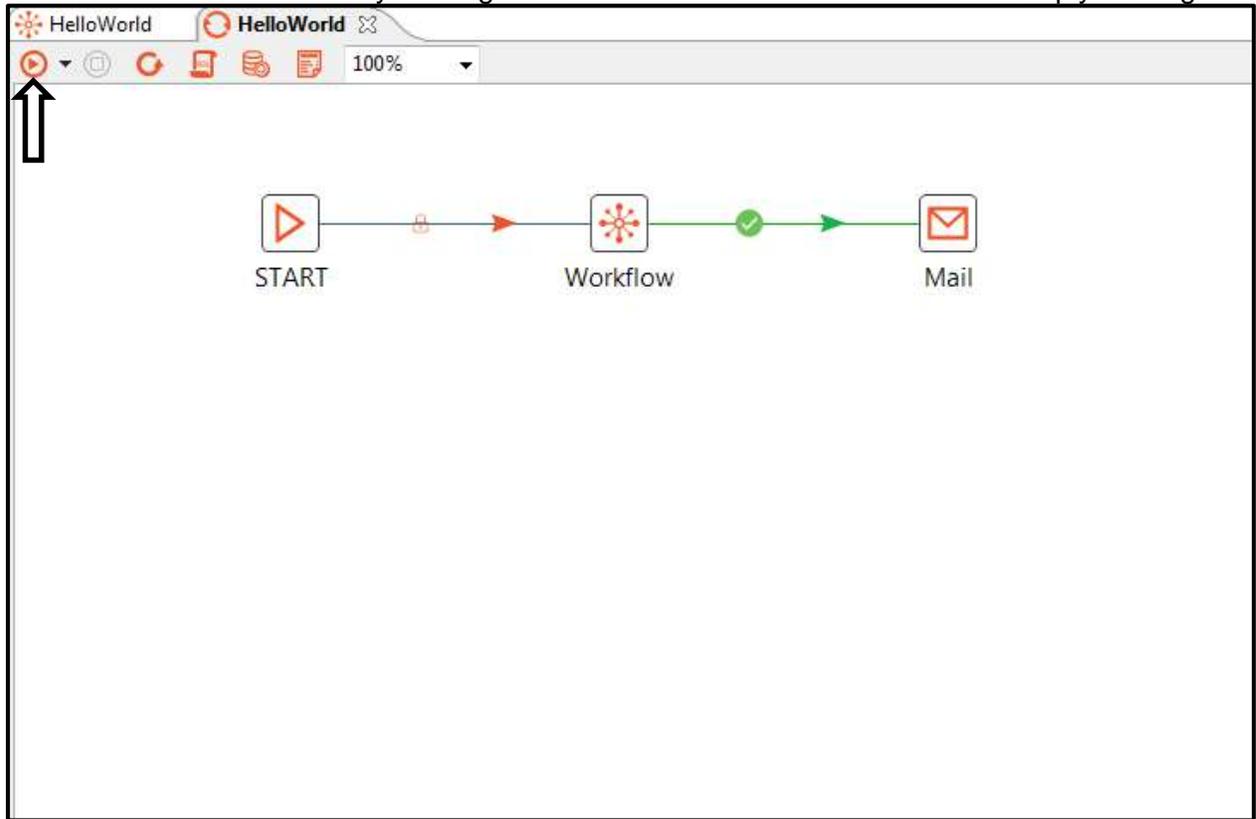
Embedded images

#	Image	Content ID
1		

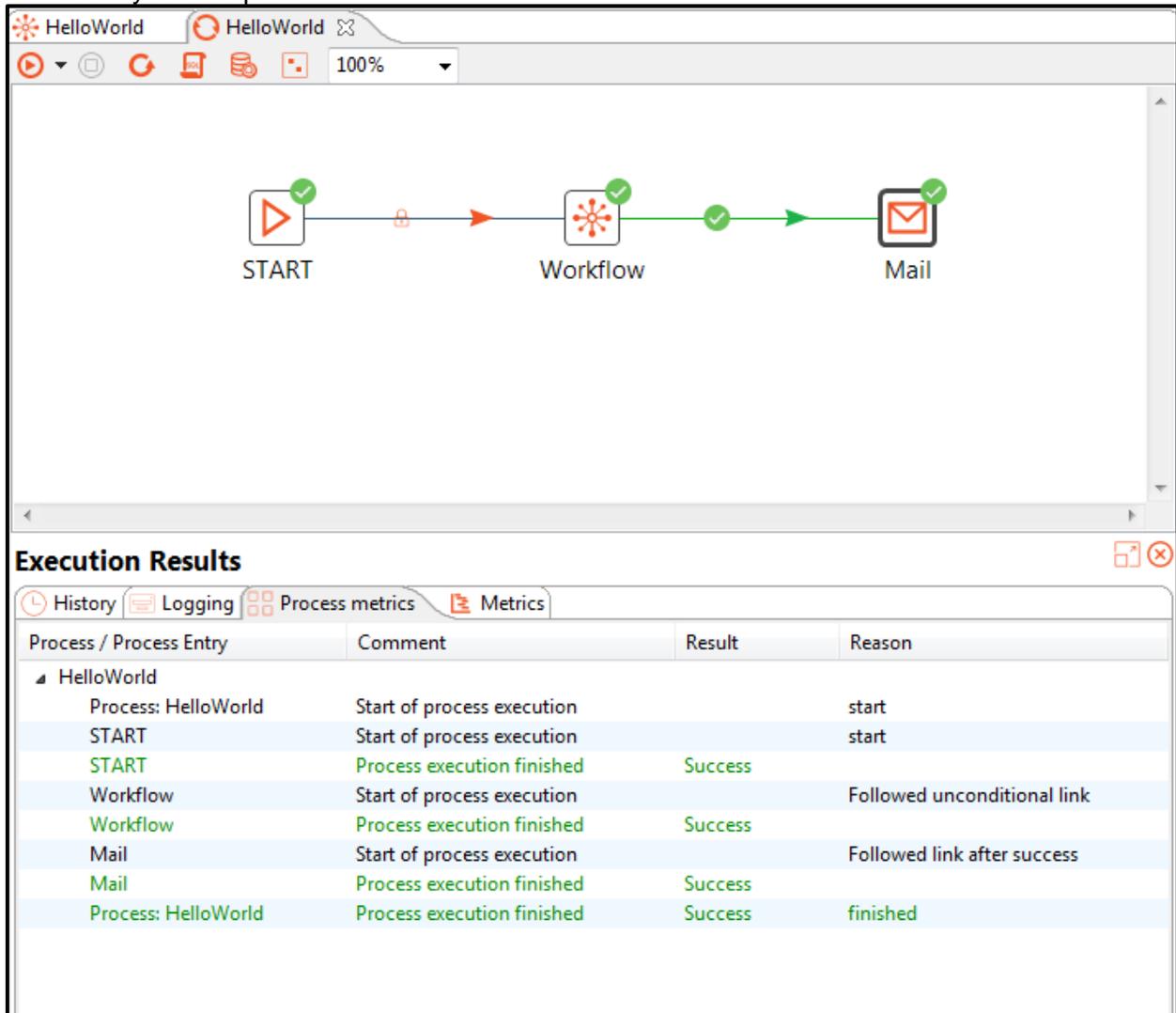
Delete Edit

OK Cancel

14. To run the process, click **Run** which is placed exactly below the workflow name in the right pan. You also execute workflow by clicking **Action >> Run** from the main menu or simply clicking F9.



12. This will result in process getting executed. If process runs successfully, you will see a green check on every entry and below you will see some step metrics. You can also check email received by the recipient



Execution Results

Process / Process Entry	Comment	Result	Reason
Process: HelloWorld	Start of process execution		start
START	Start of process execution		start
START	Process execution finished	Success	
Workflow	Start of process execution		Followed unconditional link
Workflow	Process execution finished	Success	
Mail	Start of process execution		Followed link after success
Mail	Process execution finished	Success	
Process: HelloWorld	Process execution finished	Success	finished

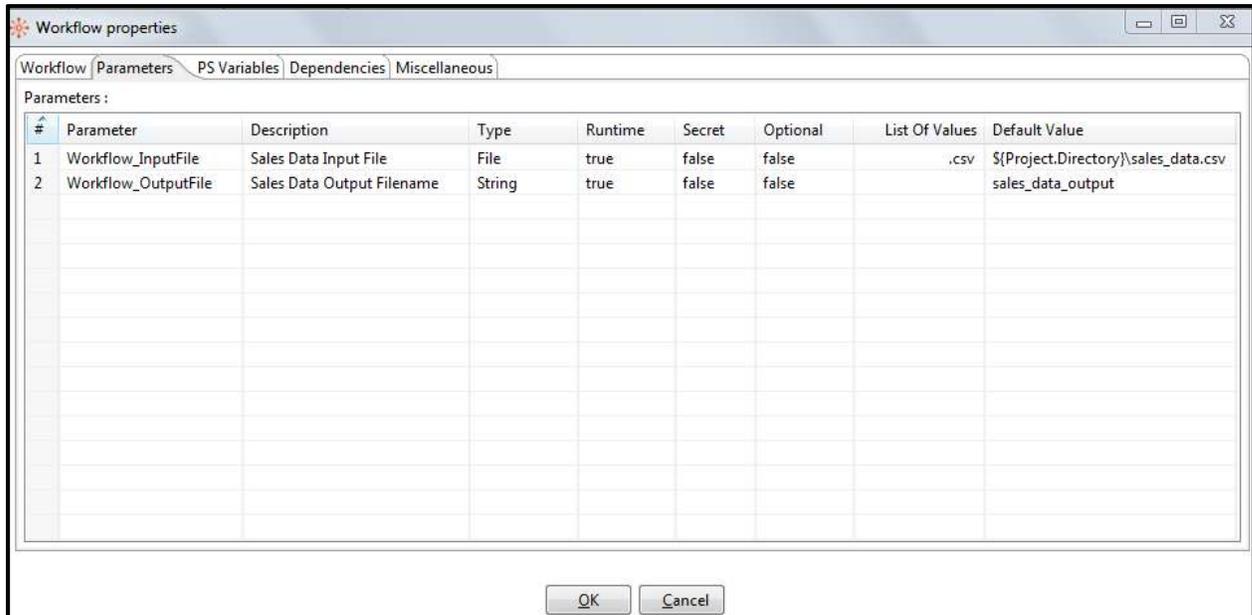
8.2.1 How a process runs

A process is not about processing rows as is the case with a workflow. Process executes in a sequential manner, that is, an entry does not start its execution unless its predecessor is complete. A process is more about executing workflows, sending mails on failure, transferring files via FTP etc. A process in turn can call a process or a workflow.

8.2.2 Modifying HelloWorld workflow to support parameters

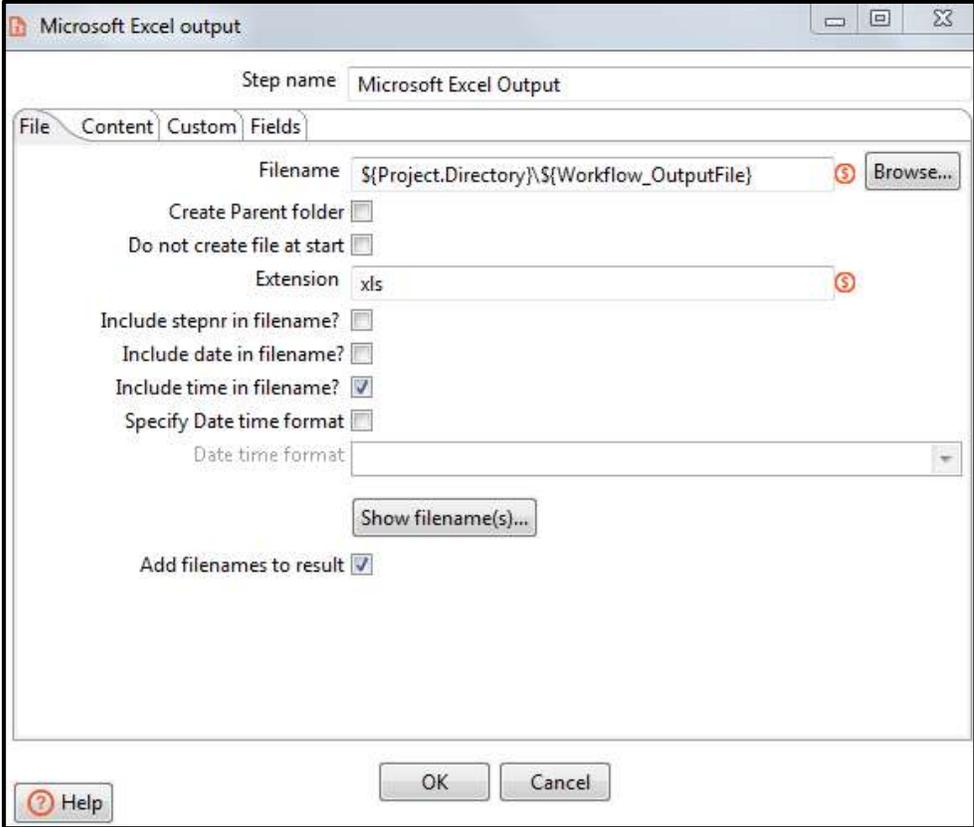
Here we will revisit **HelloWorld** workflow and **HelloWorld** process we built earlier and parameterize them.

- i. We will revisit **HelloWorld** workflow we built earlier. Open HelloWorld.psw file in **Process Studio** and **double click** on the working area to open **Workflow properties** dialog box and go to **Parameters** tab.



Add two parameters to the workflow as shown, one for input file and the second for output filename. Type of these parameters can as well be String, there is not much difference between the two. When we eventually publish this workflow to **AutomationEdge** server, having parameter type as **File**, gives users ability to select a file on their machine using **File Selection Operating System** dialog as opposed to having to type in path of the file, if parameter type were String. At runtime we may prefer to choose the input file hence we choose Type as file. However, excel output is a new file that is generated. We may not want to choose a file but would like to give a preferred name to the file. Hence, we chose Type as String. All, output files in Process Studio are by default stored on AutomationEdge Server in addition to absolute file path provided in Process Studio (this filepath may not be accessible to all Server or Agent machines). Output files are available for download upon request execution completion. Give default value of Workflow_InputFile parameter to the absolute path of sales_data.csv file and default value of Workflow_OutputFile parameter as the filename of Excel output file. **Secret**, **Optional** and **Runtime** fields have meaning only when you publish the workflow to **AutomationEdge** Server. For now, keep default values for **Secret and Optional** as shown in the screen capture. Keep **Runtime** true as we want both these fields to be provided during execution at runtime. If parameter is of type File, we need to specify the extension allowed in List of Values. In this case we have provided .csv

- iii. We have xls in the extension field. Hence, we need not provide .xls extension in the output filename parameter. However, we need to provide the absolute file path before the output filename parameter. We have provided `${Project.Directory}` as the file path for the filename as shown below. Since the output file is in the same directory as the workflow we can also provide `${Internal.Entry.Current.Directory}` as the file path for the filename.



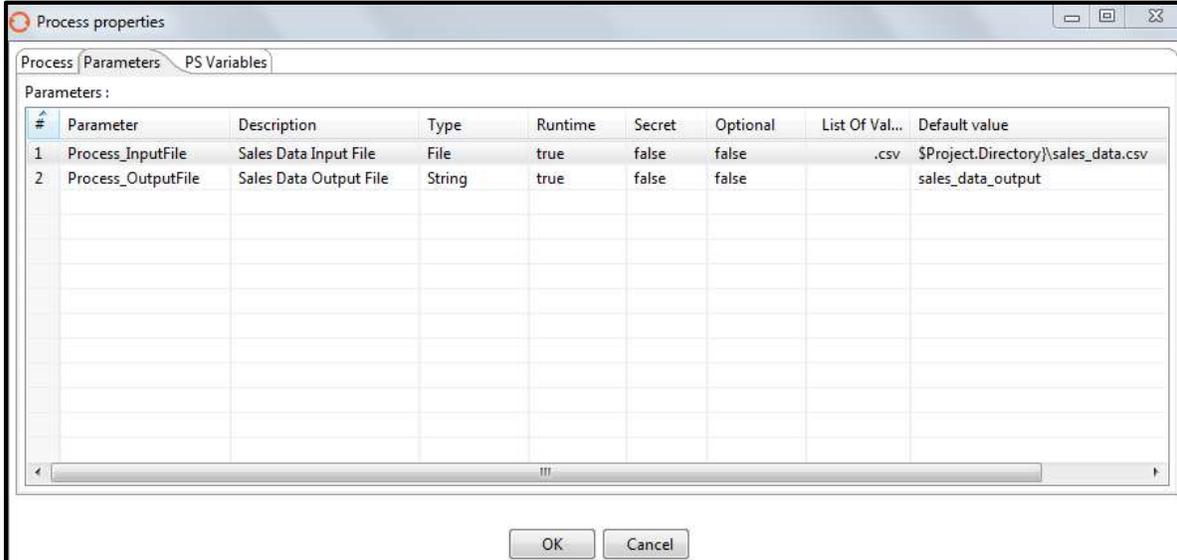
The screenshot displays the 'Microsoft Excel output' dialog box. The 'Step name' is 'Microsoft Excel Output'. The 'Fields' tab is selected, showing the following configuration:

- Filename: `${Project.Directory}\${Workflow_OutputFile}` (with a 'Browse...' button)
- Create Parent folder:
- Do not create file at start:
- Extension: `xls`
- Include stepnr in filename?:
- Include date in filename?:
- Include time in filename?:
- Specify Date time format:
- Date time format: (empty dropdown)
- Show filename(s)...: (button)
- Add filenames to result:

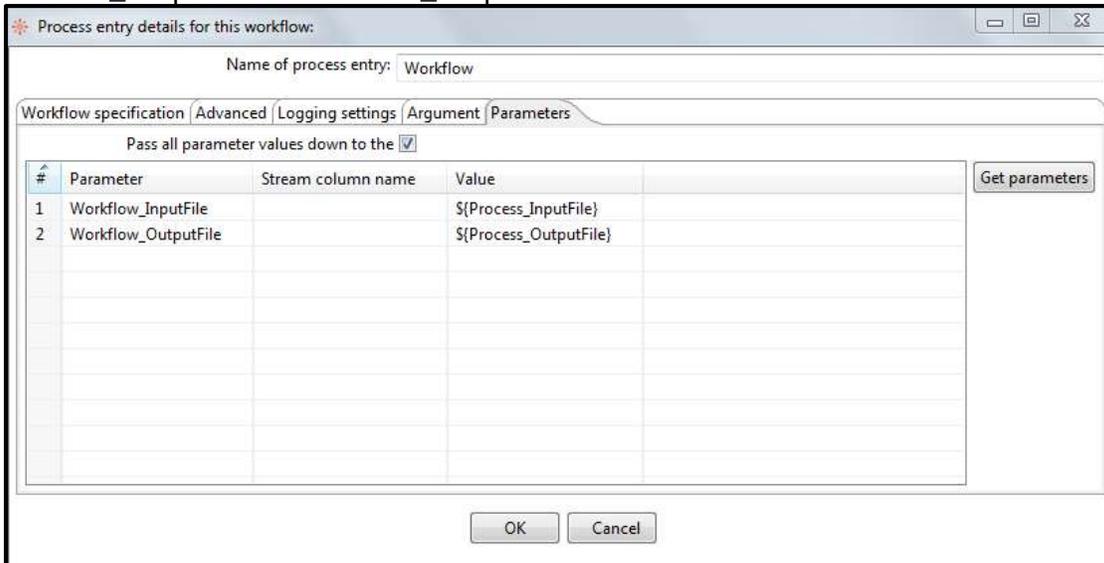
Buttons at the bottom include 'Help', 'OK', and 'Cancel'.

8.2.3 Modifying HelloWorld process to support parameters

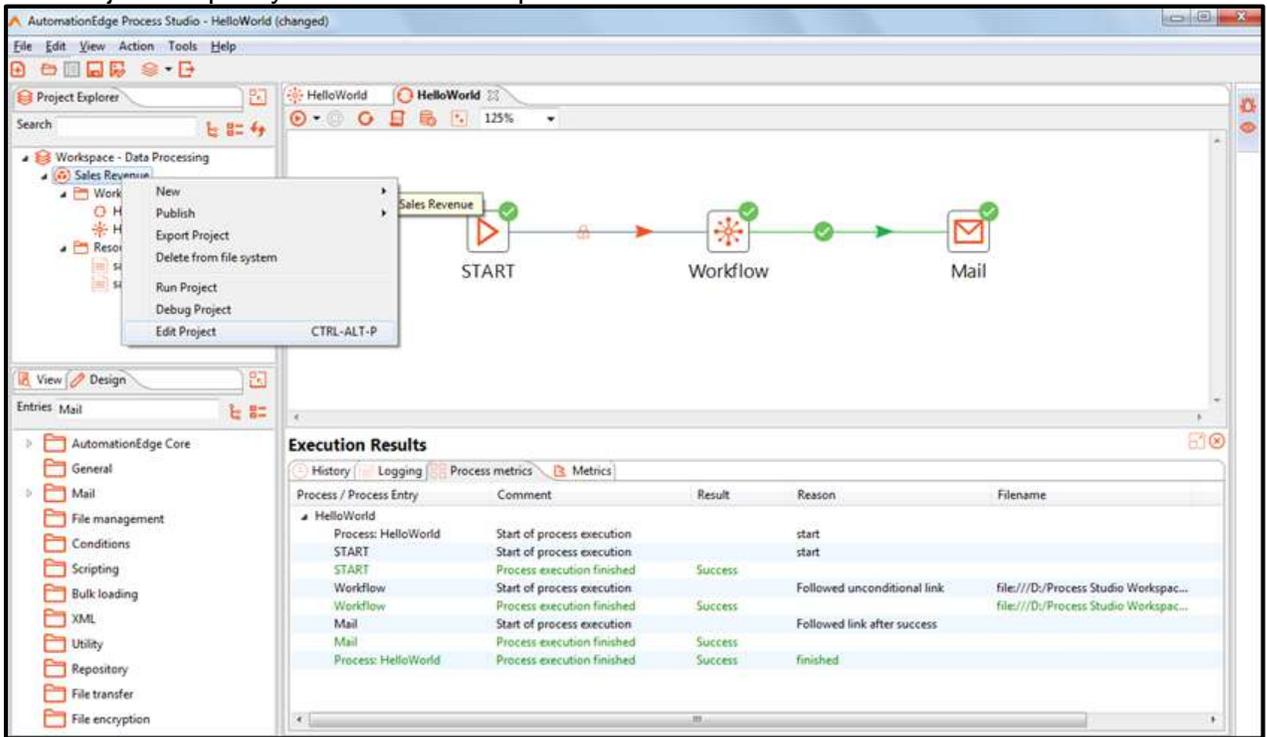
1. Open HelloWorld.psp file in **Process Studio** and double click on the working area to open **Process properties** dialog box to create two parameters. Parameter details are exactly the same as we had done during workflow parameter creation. Just name the parameters as **Process_InputFile** and **Process_OutputFile**.



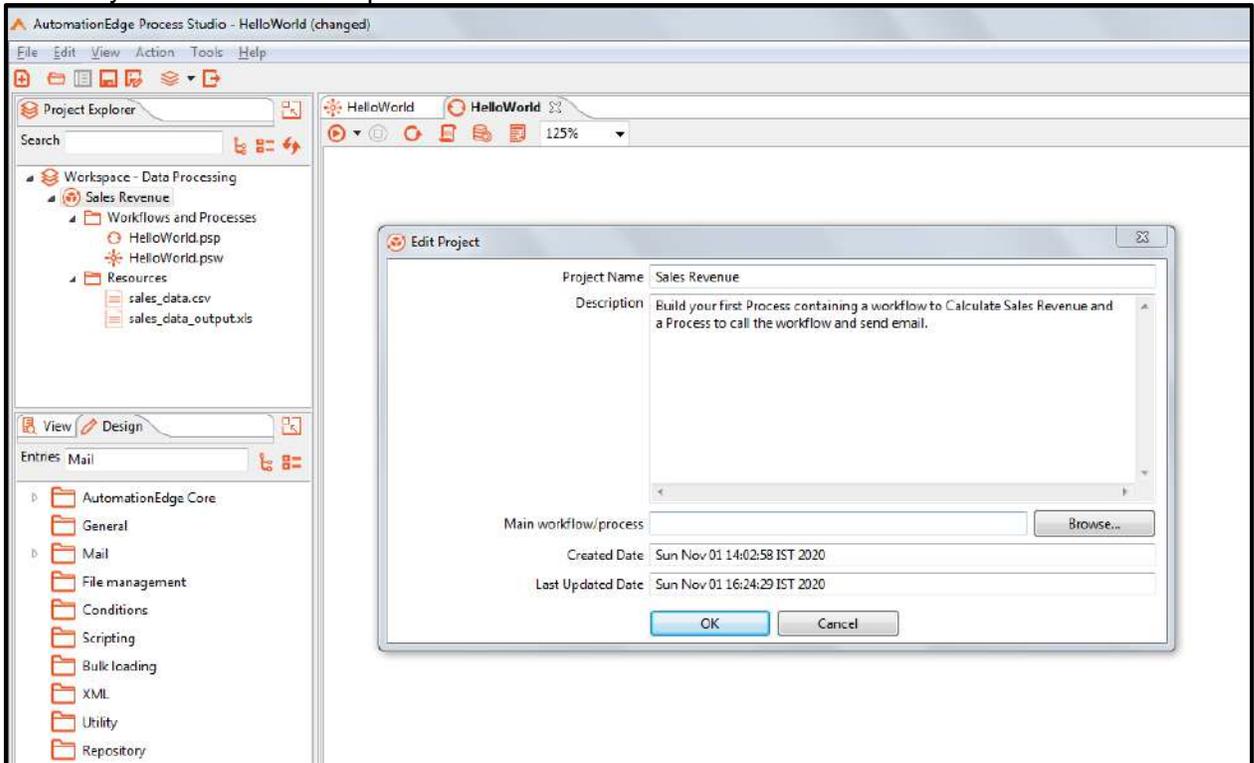
2. Though we have defined these parameters for the process, they are actually getting used in the workflow which this process calls using **Workflow** entry. Double click **Workflow** entry to edit its properties. Go to the **Parameters** tab. This tab lets you map one or more process parameters to one or more workflow parameters. Click **Get Parameters** and you will start seeing workflow parameters in the first column. Provide appropriate **Value** for both these parameters as shown in below image. With this, we have mapped **Process_InputFile** to **Workflow_InputFile** and **Process_OutputFile** to **Workflow_OutputFile**.



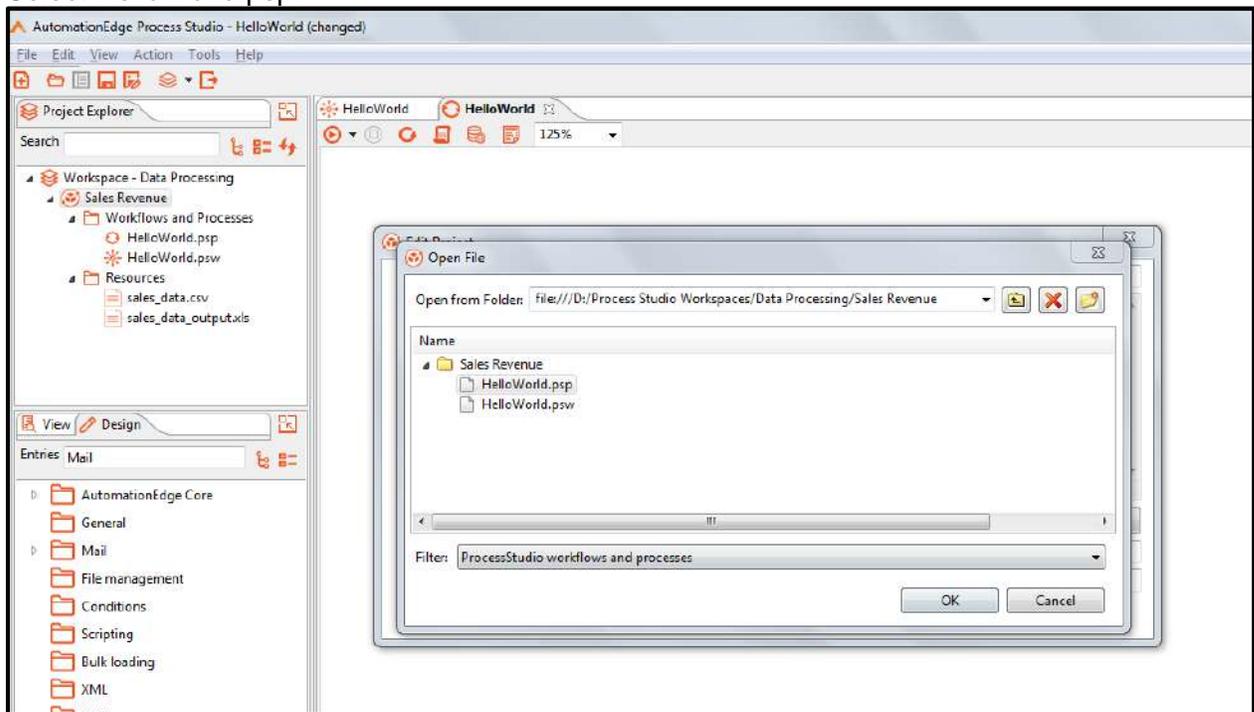
3. Edit Project to specify a Main workflow/ process.



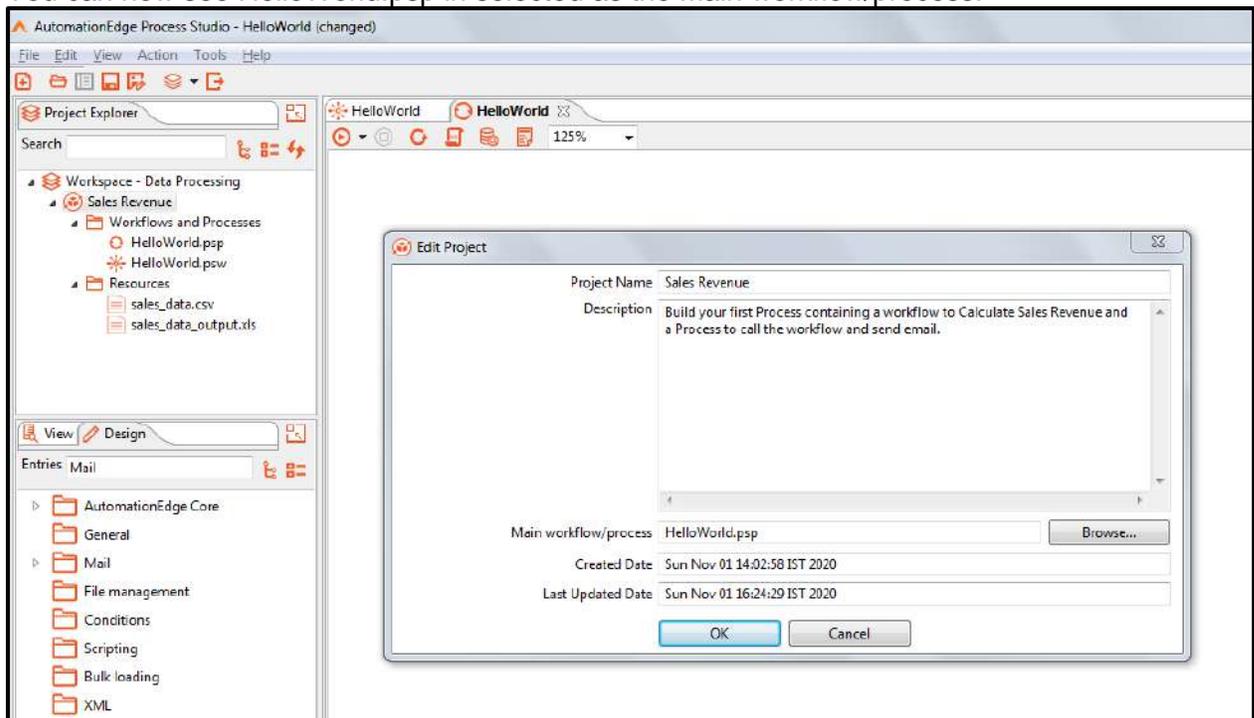
4. Browse your main workflow/process



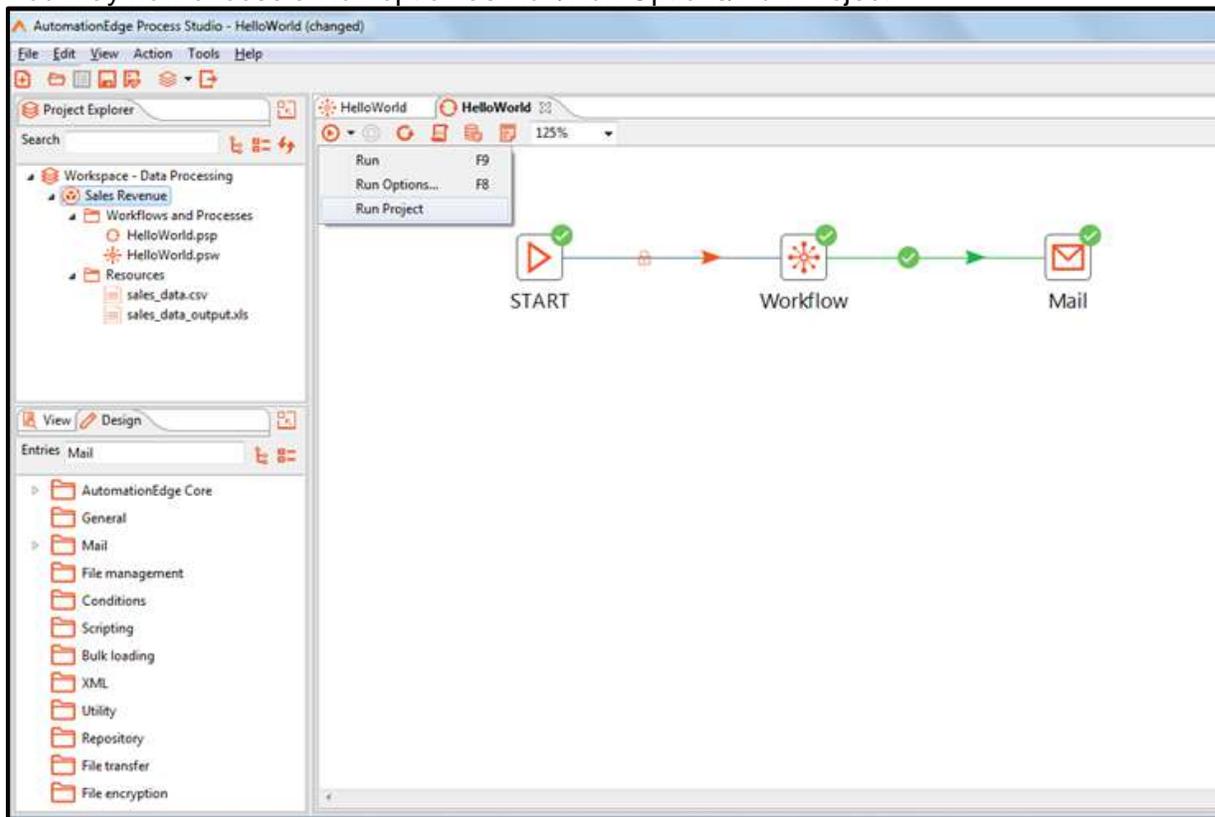
5. Select HelloWorld.psp.



6. You can now see HelloWorld.psp in selected as the Main workflow/process.



7. You may now choose a Run option as Run/Run Options/Run Project.



8. The following is the Project execution result.

The screenshot displays a workflow execution in Process Studio. The workflow consists of three steps: START, Workflow, and Mail. All steps are marked with a green checkmark, indicating successful completion. The Execution Results table below provides a detailed log of the process execution.

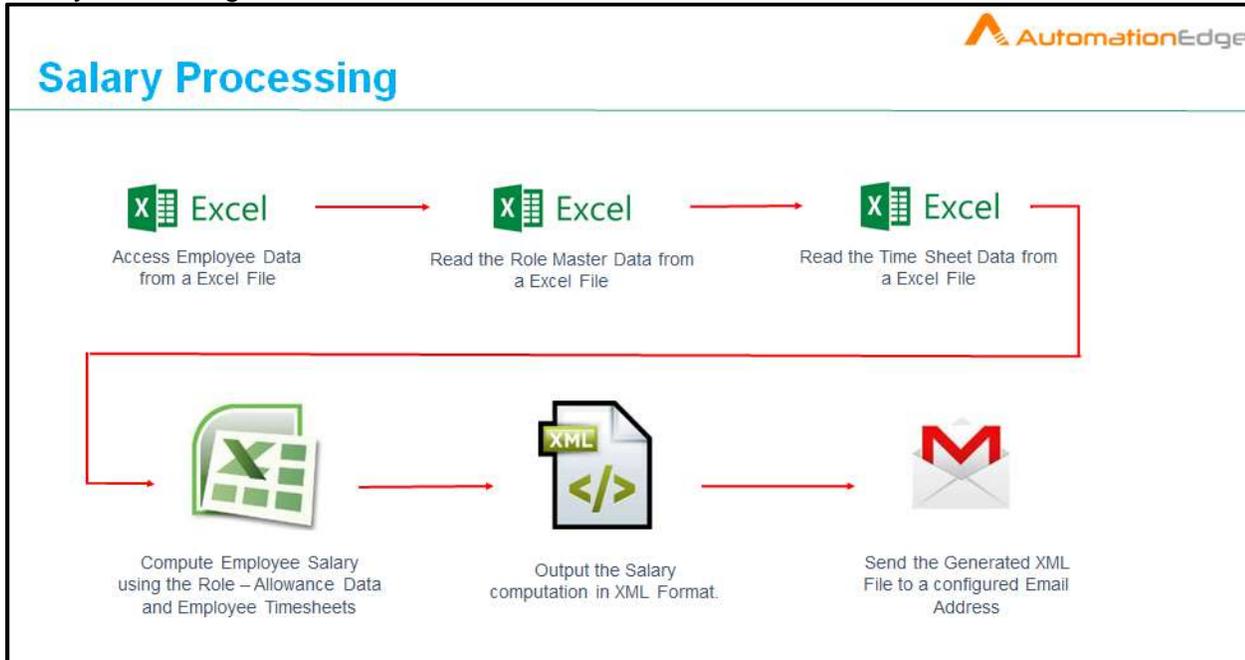
Process / Process Entry	Comment	Result	Reason	Filename	Nr	Log date
HelloWorld						
Process: HelloWorld	Start of process execution		start			2021/07/03 15:...
START	Start of process execution		start			2021/07/03 15:...
START	Process execution finished	Success			0	2021/07/03 15:...
Workflow	Start of process execution		Followed uncondition...	file:///D:/Workspace/D...		2021/07/03 15:...
Workflow	Process execution finished	Success		file:///D:/Workspace/D...	1	2021/07/03 15:...
Mail	Start of process execution		Followed link after suc...			2021/07/03 15:...
Mail	Process execution finished	Success			1	2021/07/03 15:...
Process: HelloWorld	Process execution finished	Success	finished		1	2021/07/03 15:...

9. The first Data Processing Project: Sales Revenue is completed.

9 Project 6: Payroll

9.1 Build your Second project – Payroll for SalaryProcessing

Salary Processing can be summarized in the screen shot below.

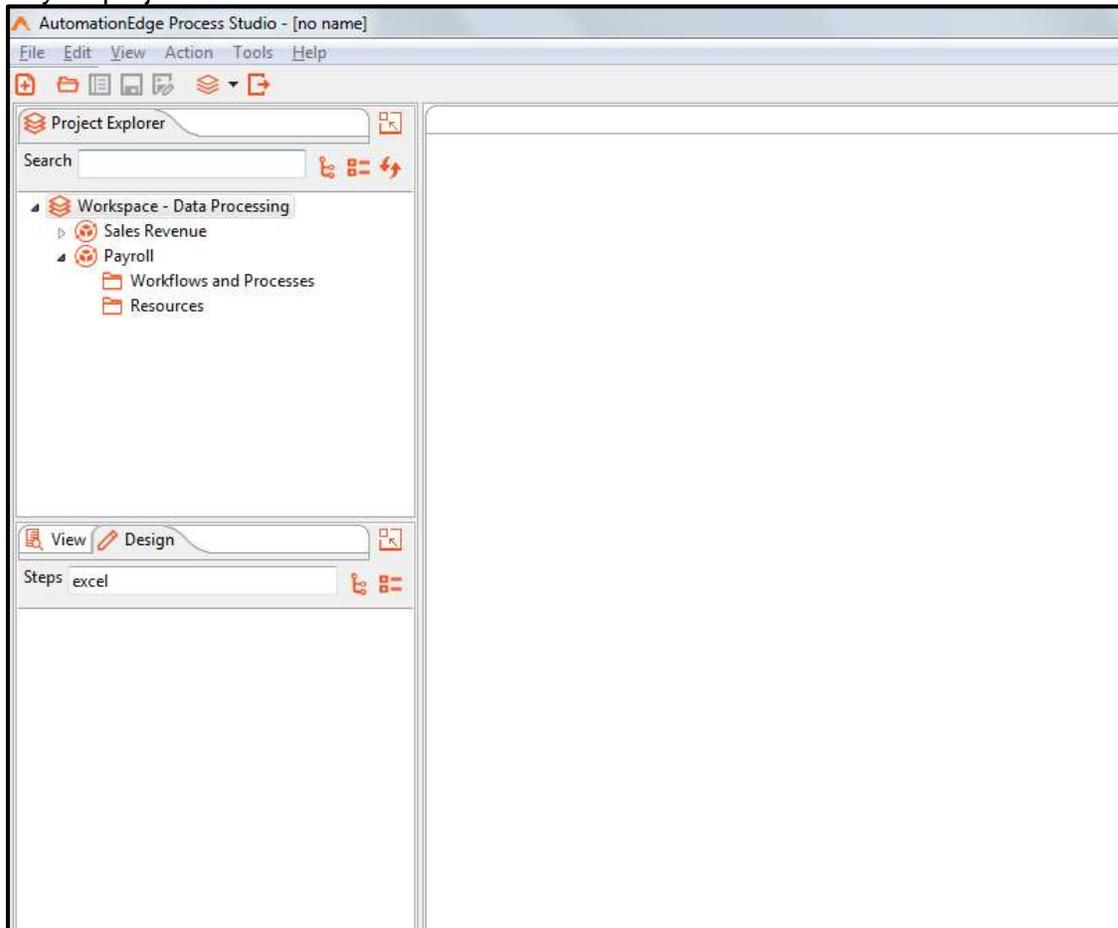


We will implement Salary Processing (or Payroll) with two child Workflows and a parent Process,

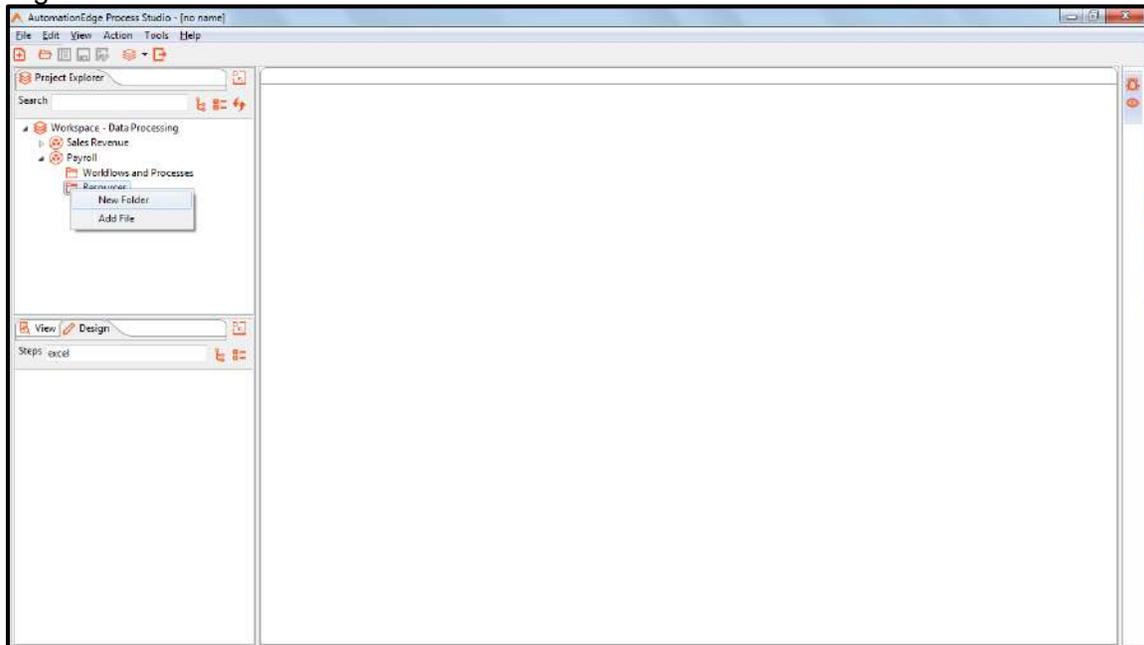
- i. Compute Salary Workflow
- ii. Send Email Workflow
- iii. Employee Main Process

Following are the steps to create Payroll (for salary processing) project with the workflows.

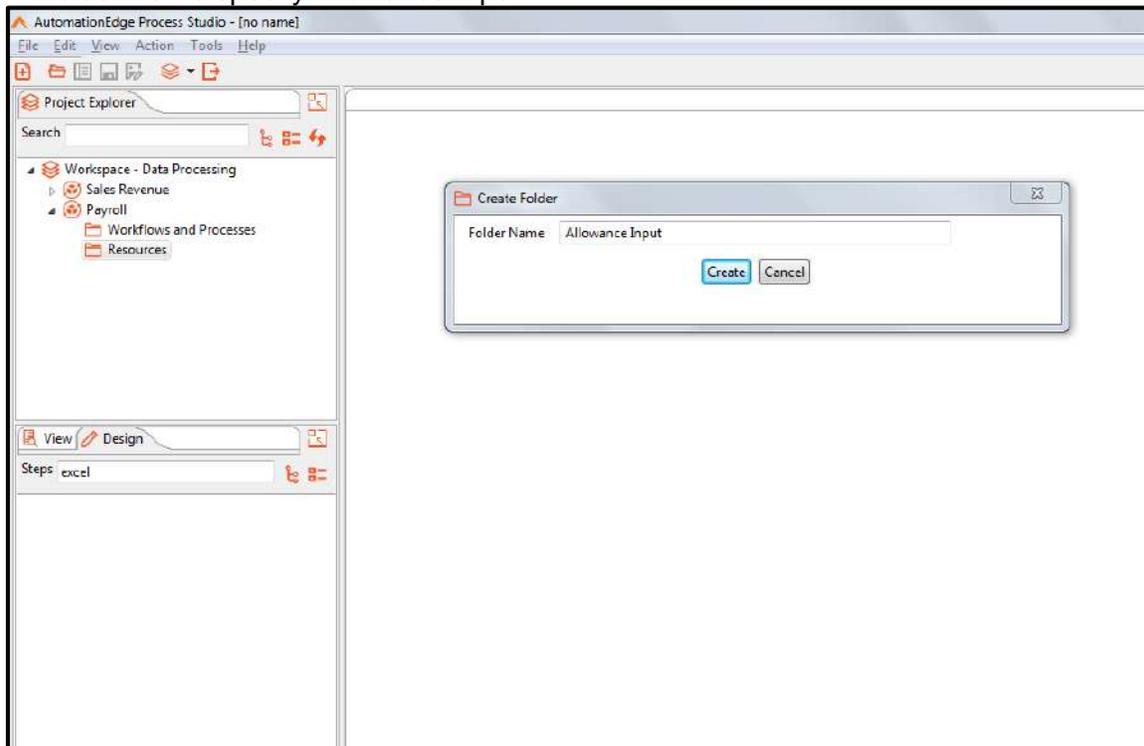
1. Open the Data Processing workspace.
2. Right click on the workspace and select the option New Project. Name the new project Payroll. Payroll project is created as seen below.



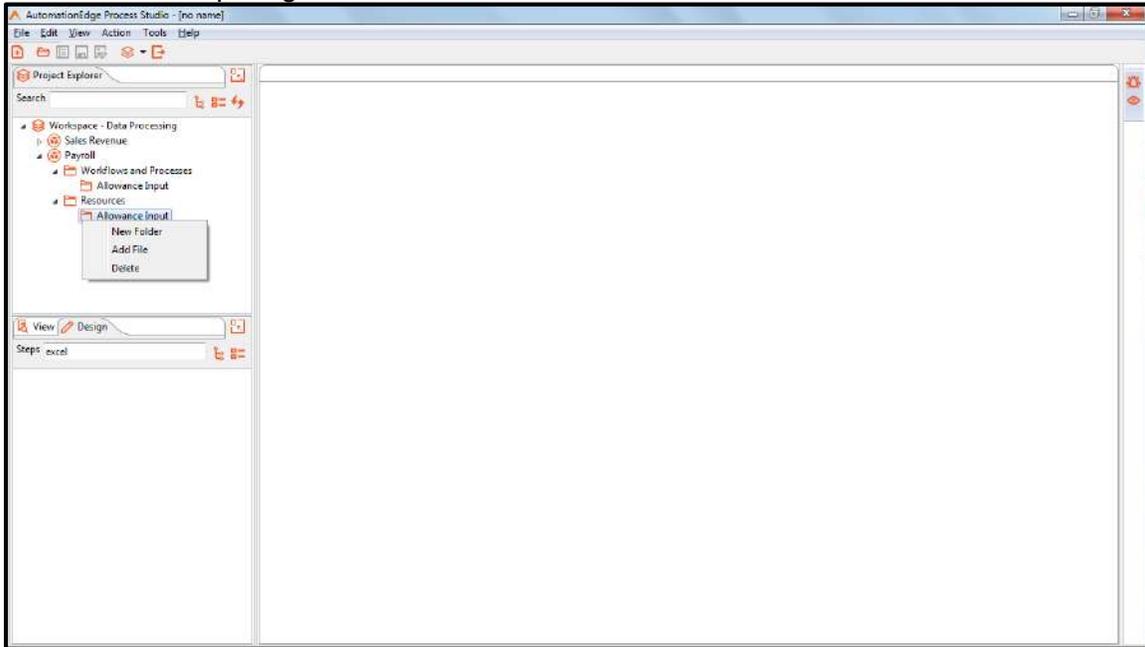
3. We wish to create three folders – Allowance input, Employee list and TimeSheet Input to store Payroll related input files.
4. Right click on Resources and select New Folder to create a folder.



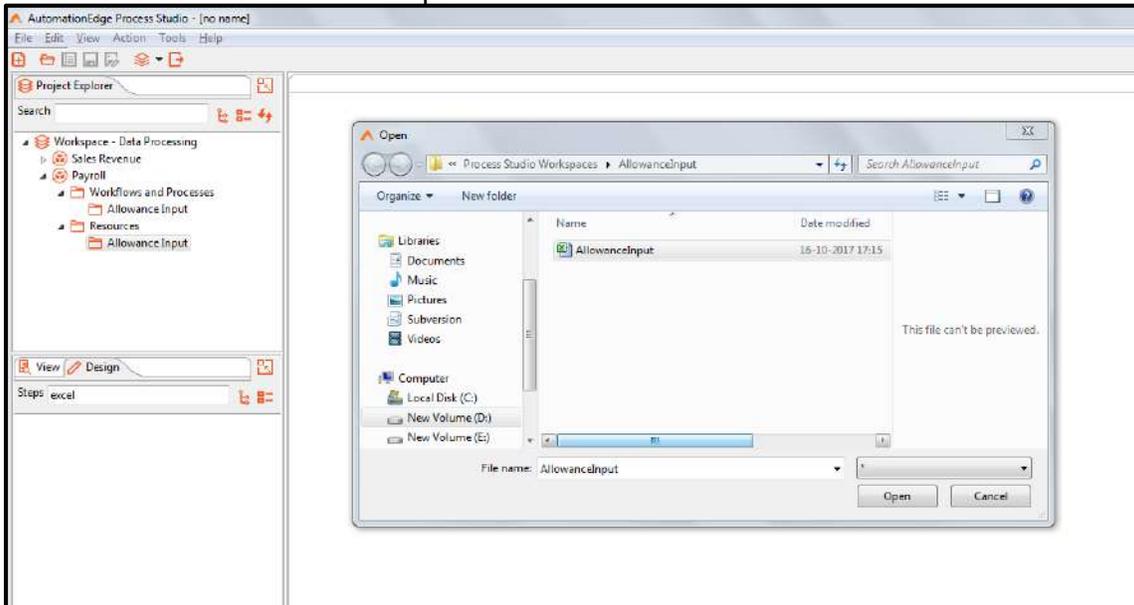
5. In Folder Name specify Allowance Input.



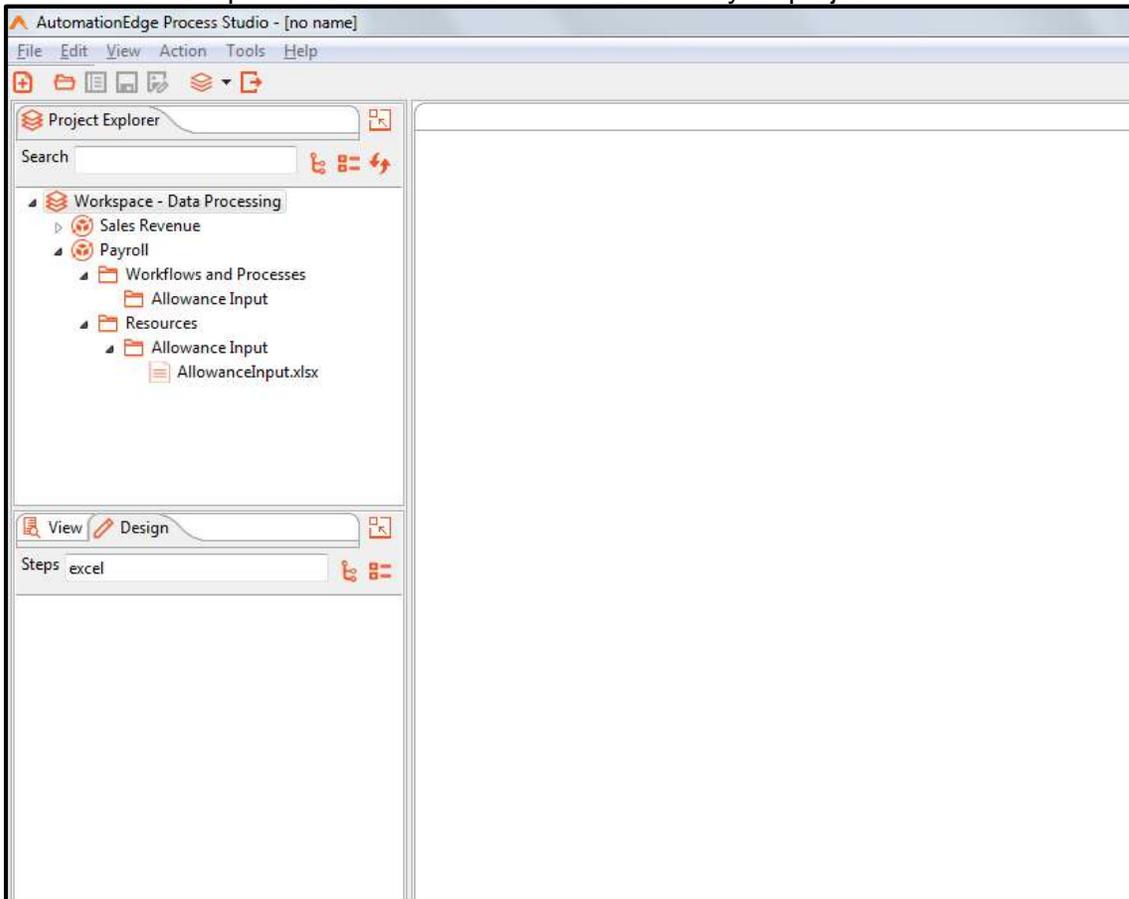
6. On Allowance Input right click to Add File.



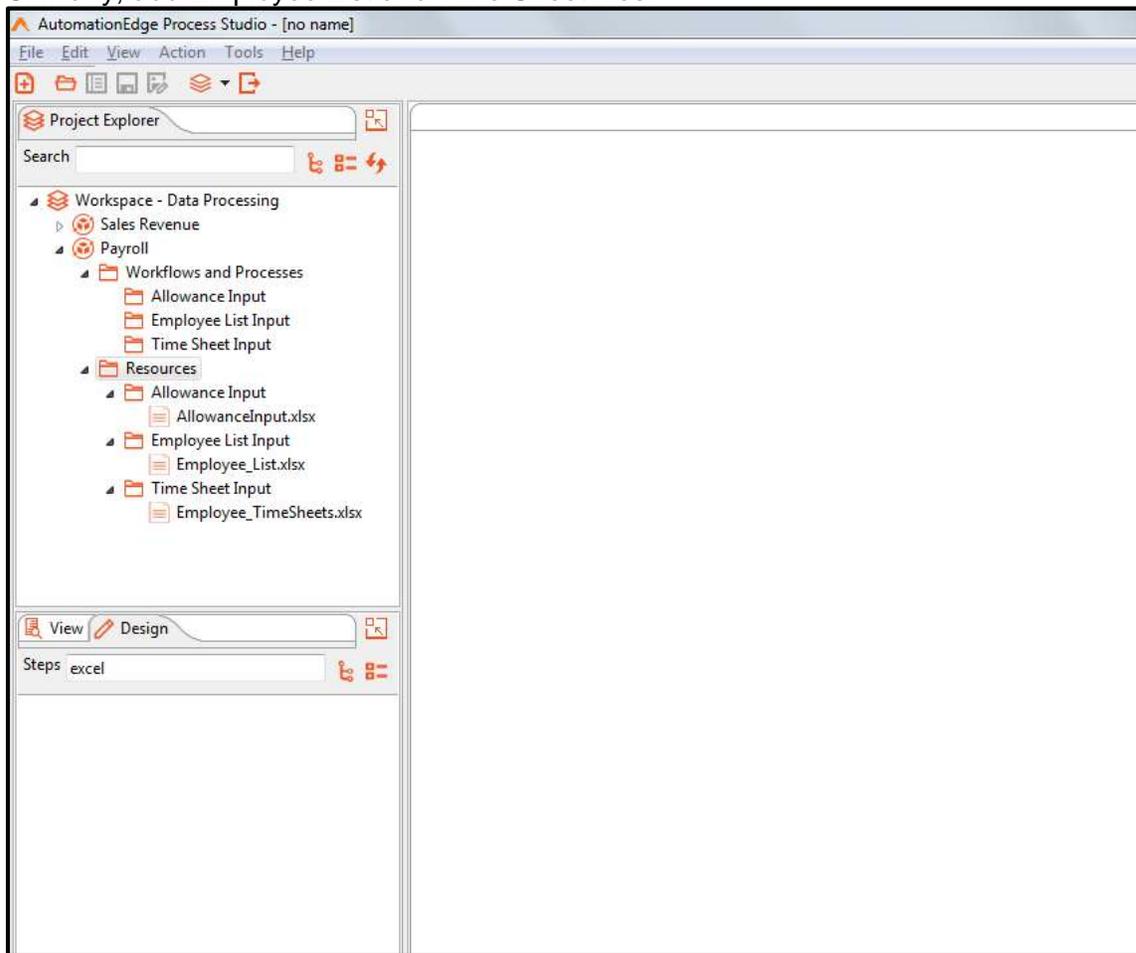
7. Locate and add file Allowance input.



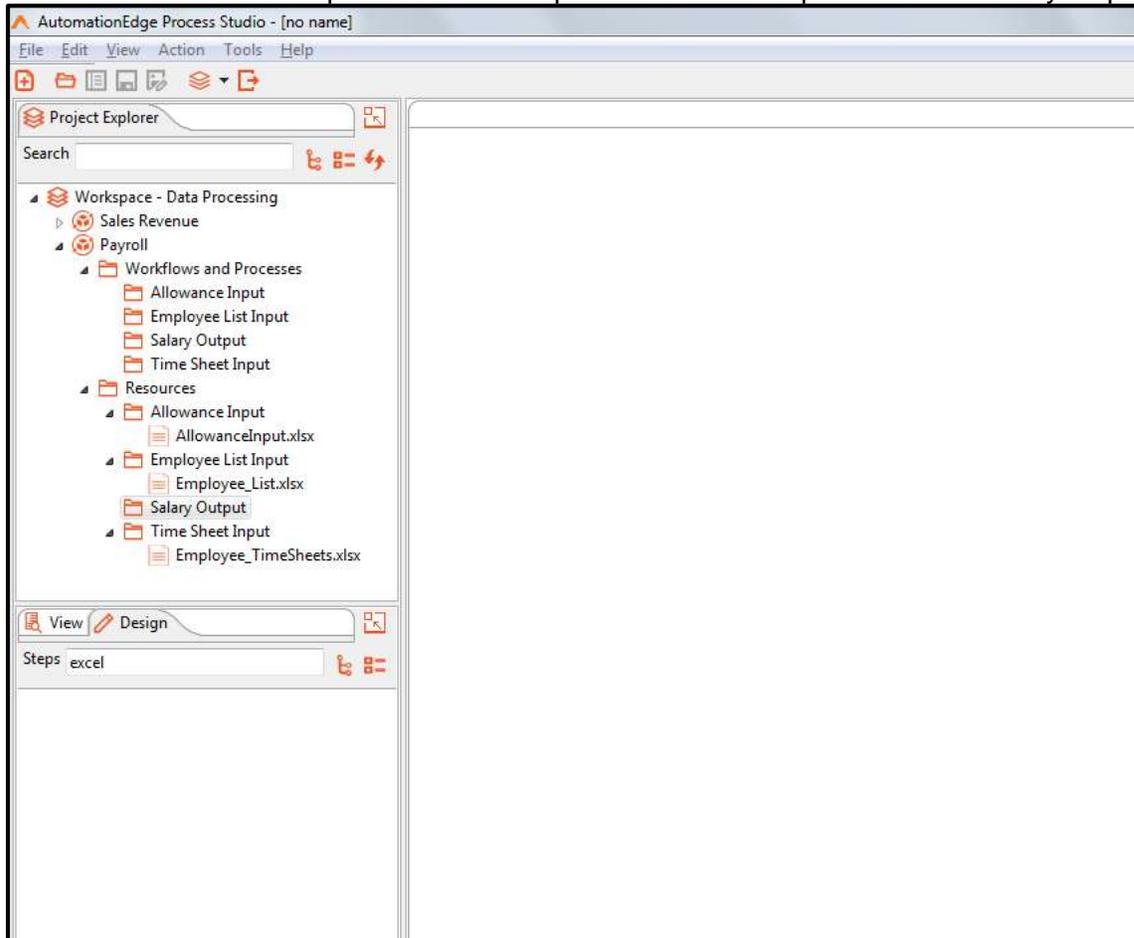
8. The Allowance Input folder and file are now visible in Payroll project.



9. Similarly, add Employee List and Time Sheet files.



10. Create one more folder for the output Payroll calculation file.
11. We can now see three input folders with input files and an output folder in the Payroll project.

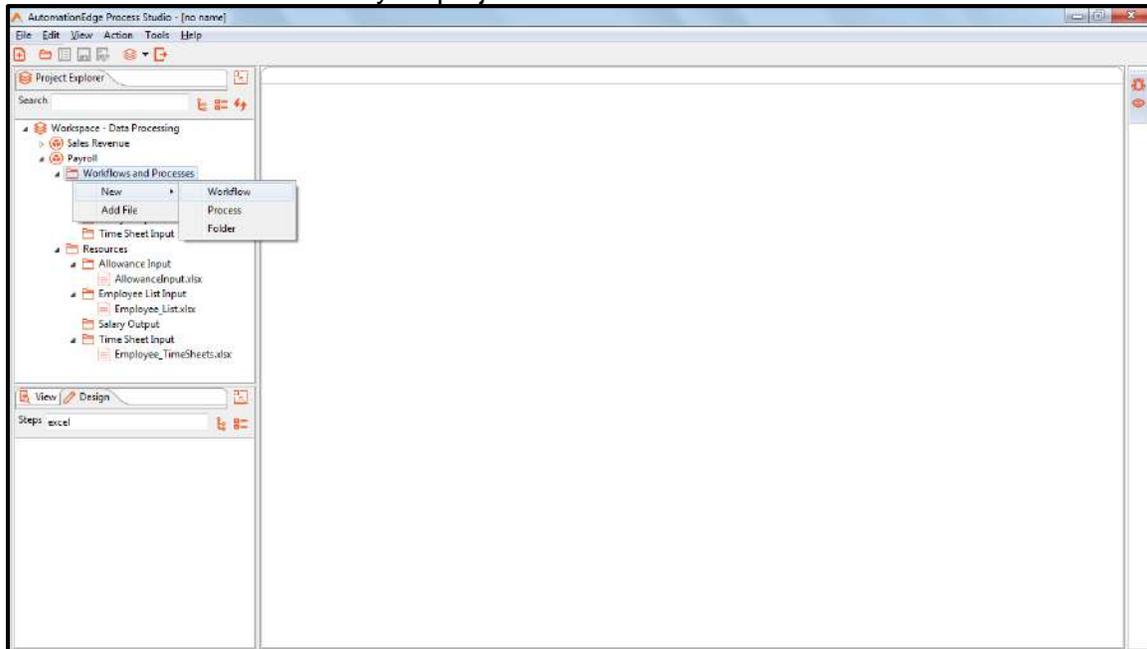


12. In the next sections we will design the three workflows

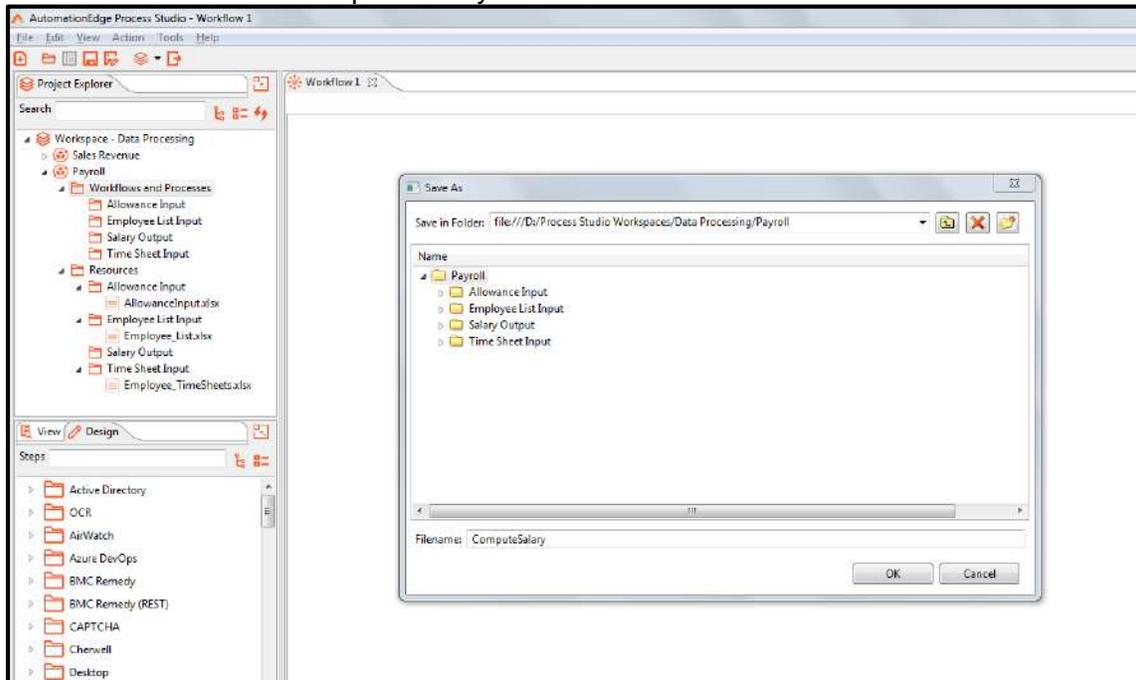
9.2 Compute Salary Workflow

First let us design the main computation workflow. Following are the steps.

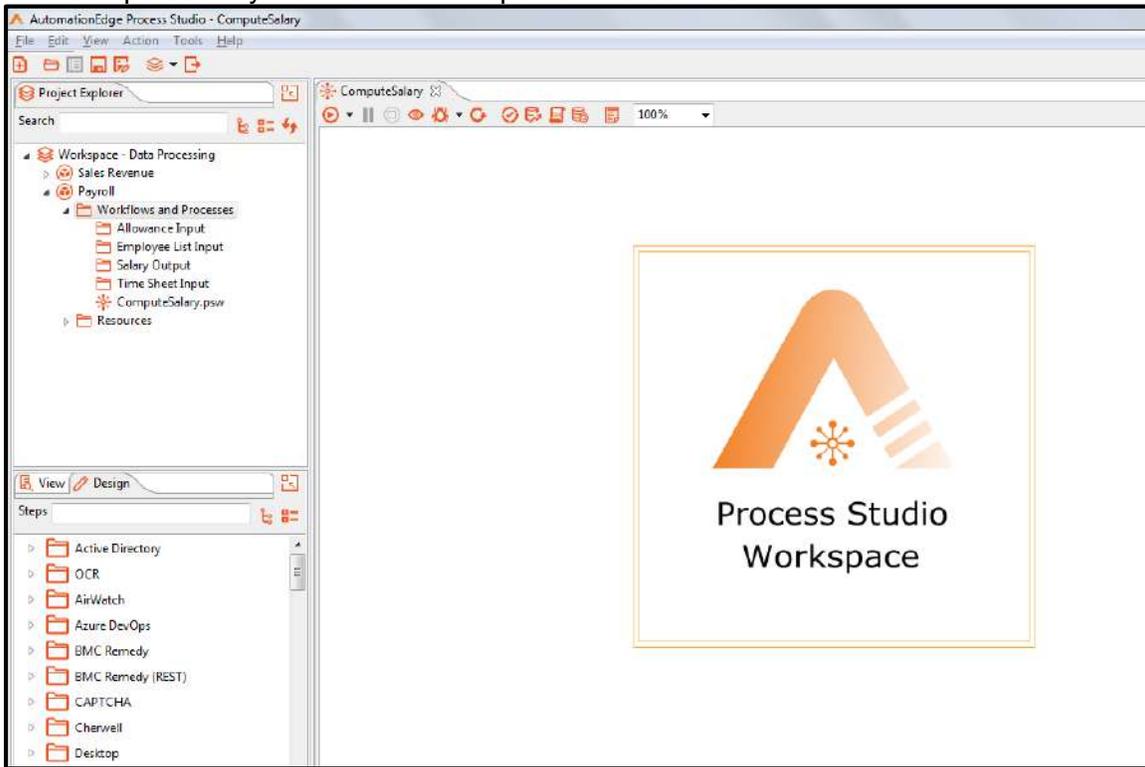
1. Create a new workflow in Payroll project.



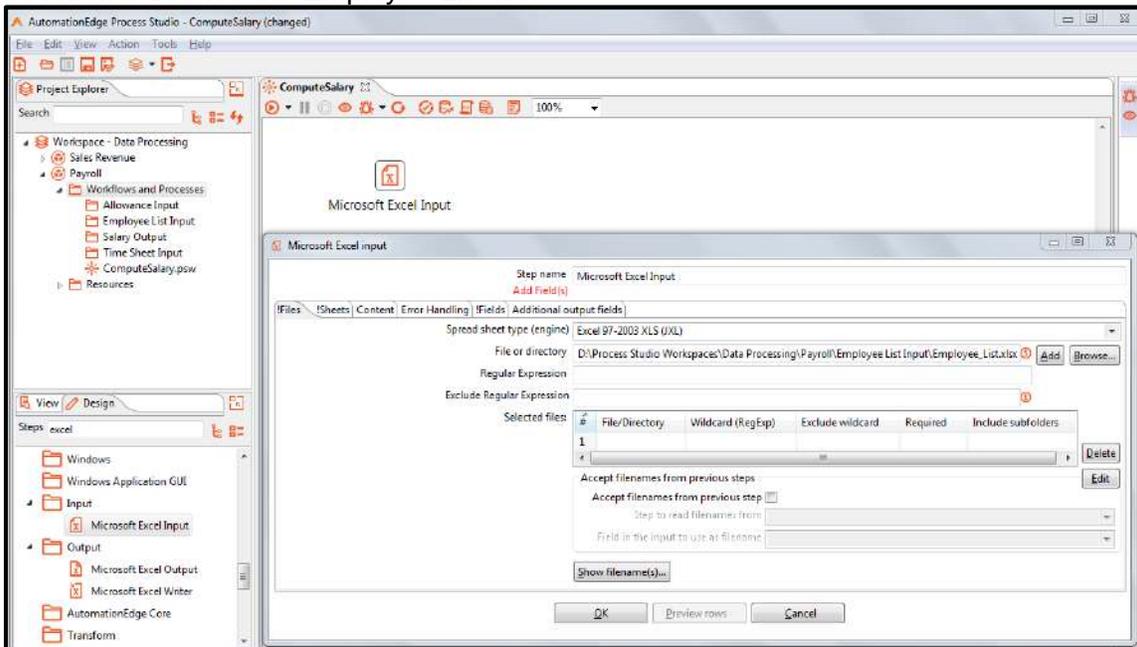
2. Save the workflow as ComputeSalary Workflow.



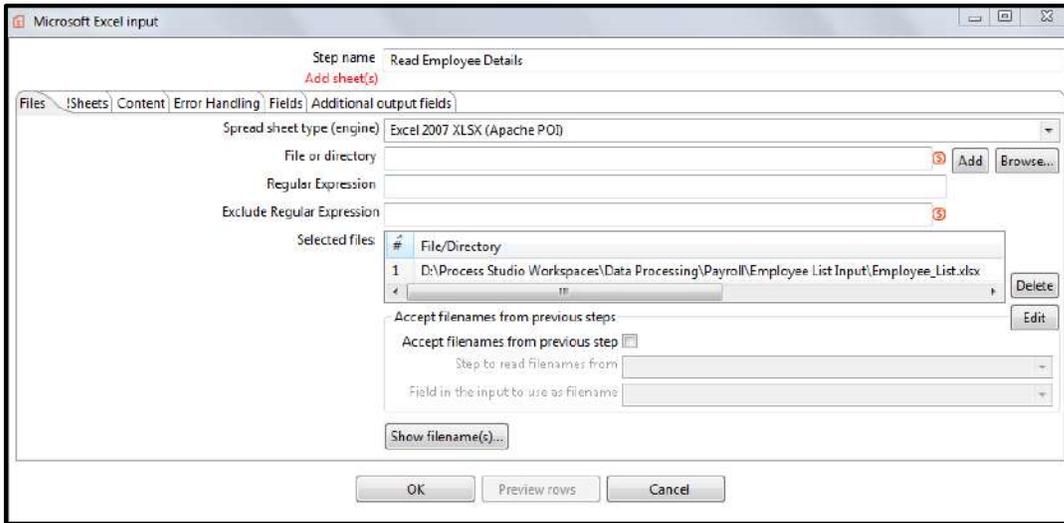
- The ComputeSalary workflow is now open on the canvas.



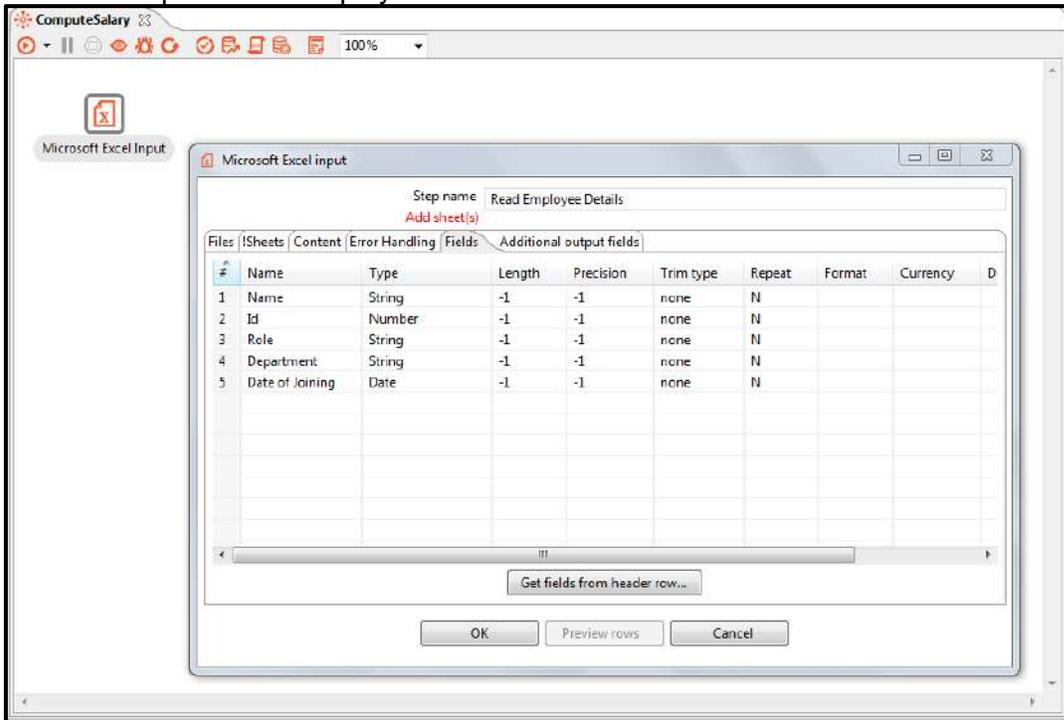
- Drag and drop a Microsoft Excel input step. Name it as Read Employee Details and configure it as follows. Browse the Employee List file.



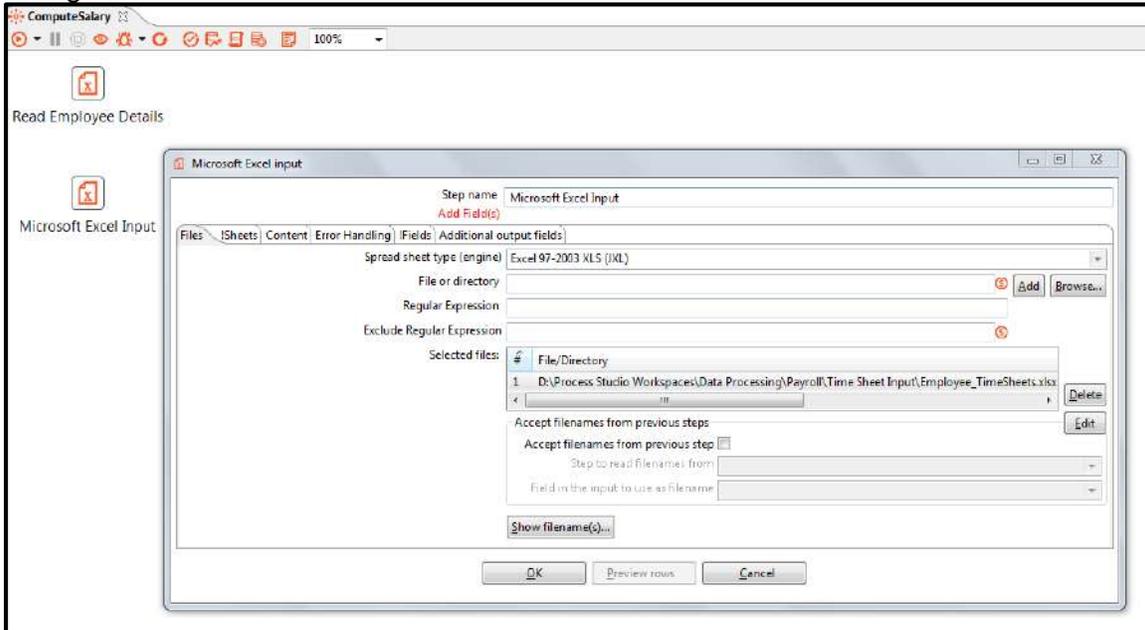
- Click Add to add the file to the Selected files list. Choose appropriate Spread Sheet Engine as seen below.



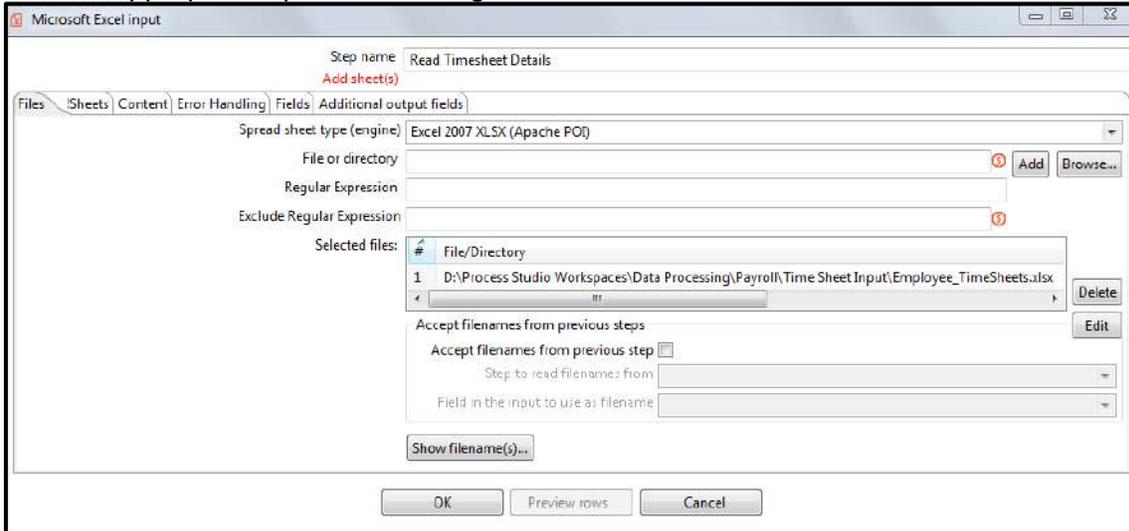
- Rename step to Read Employee Details. Click Get Fields from Header row.



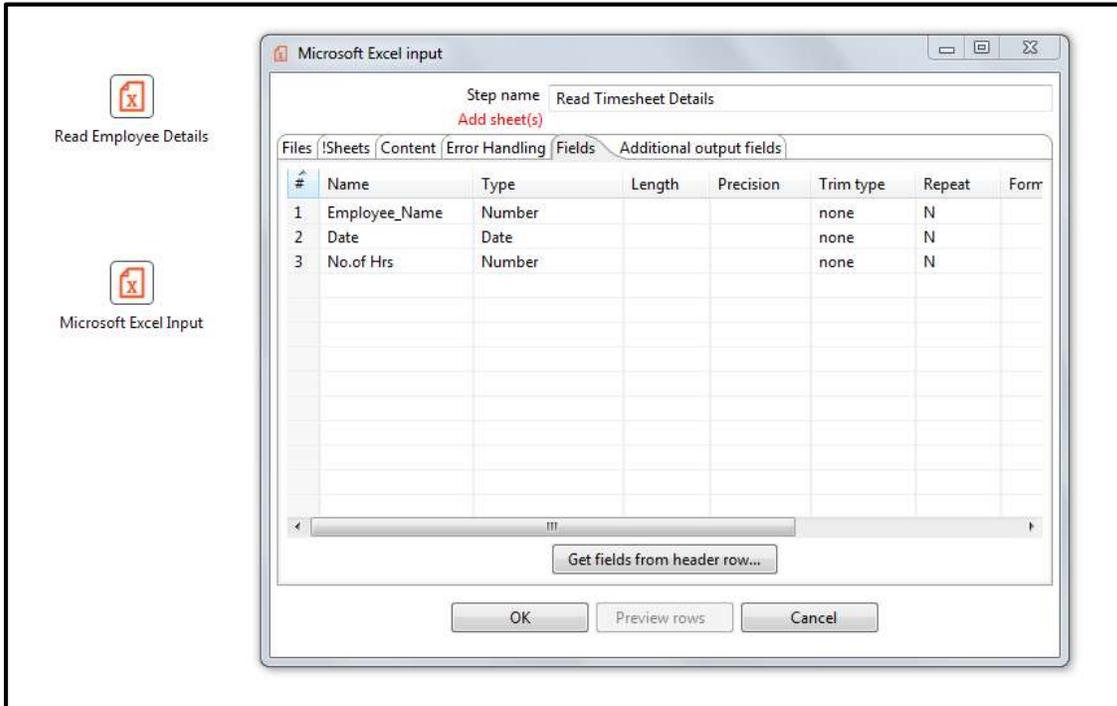
- Next, drag and drop a Microsoft Excel input step. Name it as Read TimeSheet Details and configure it as follows. Browse the Time Sheet file and click Add.



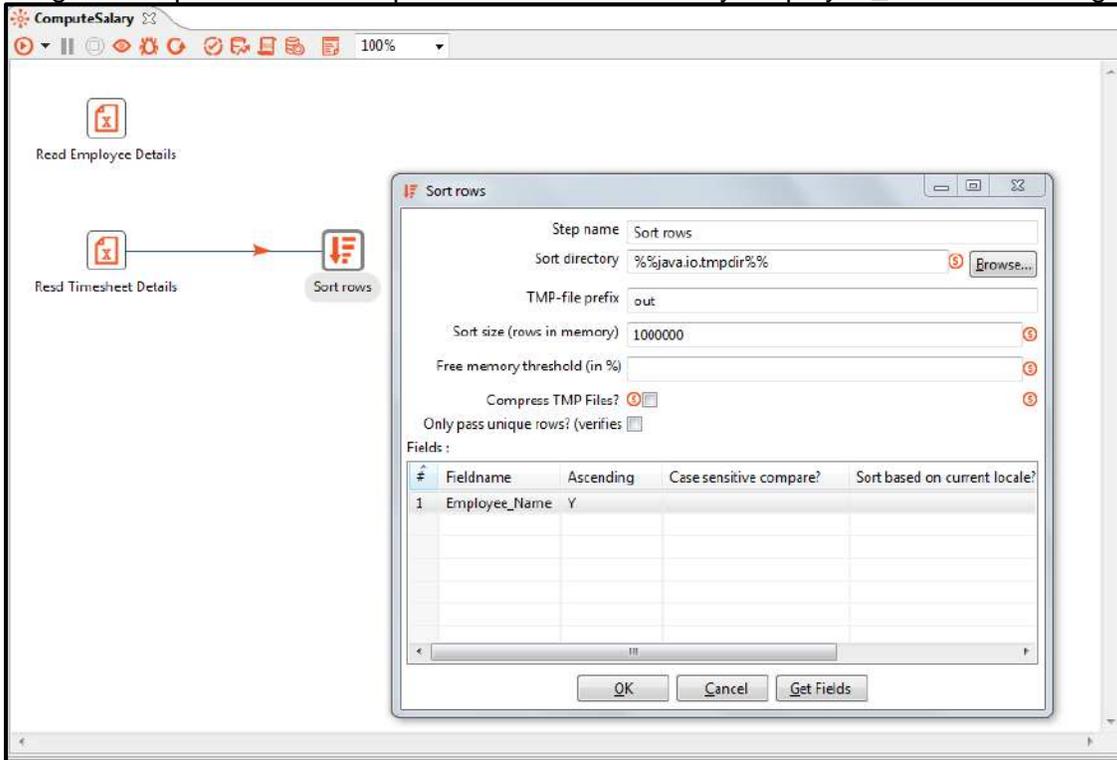
- Choose appropriate spreadsheet engine as shown below.



9. Go to the Fields tab. Click Get fields from header row.



10. Drag and drop a Sort rows step as shown below. Sort by Employee_Name ascending.



11. Drag and drop a Group by step to Group Timesheet Details by employee.

The screenshot shows a workflow diagram on the left with four steps: 'Read Employee Details', 'Read Timesheet Details', 'Sort rows', and 'Group by'. The 'Group by' step is highlighted. On the right, the 'Group By' dialog box is open. It has the following configuration:

- Step name: Group by
- Include all rows?:
- Temporary files directory: %java.io.tmpdir%
- TMP-file prefix: grp
- Add line number, restart in:
- Line number field name: (empty)
- Always give back a result row:
- The fields that make up the group:

#	Group field
1	Employee_Name
- Aggregates:

#	Name	Subject	Type
1	TotalHoursworked	No.of Hrs	Sum

12. Drag and drop a Merge Join step to merge rows from Read Employee Details step and Read TimeSheetDetails after sorting and grouping by employee.

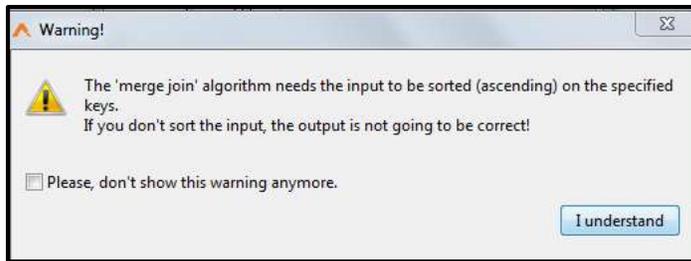
The screenshot shows a workflow diagram on the left with five steps: 'Read Employee Details', 'Read Timesheet Details', 'Sort rows', 'Group by', and 'Merge Join'. The 'Merge Join' step is highlighted. On the right, the 'Merge Join' dialog box is open. It has the following configuration:

- Step name: Merge Join
- First Step: Read Employee Details
- Second Step: Group by
- Join Type: FULL OUTER
- Keys for 1st step:

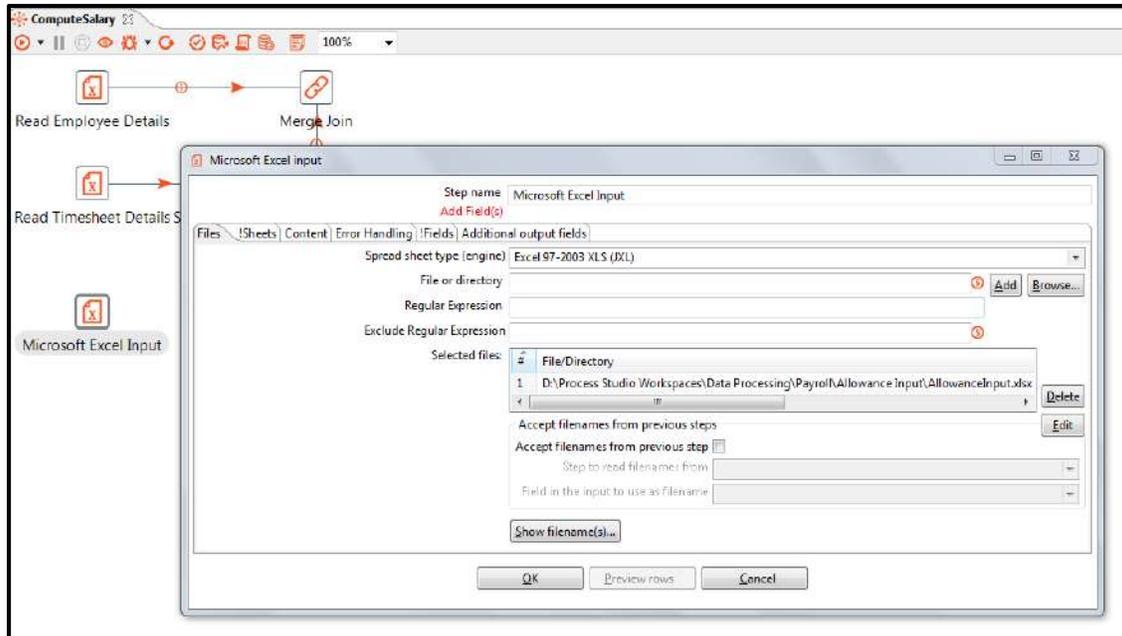
#	Key field
1	Id
- Keys for 2nd step:

#	Key field
1	Employee_Name

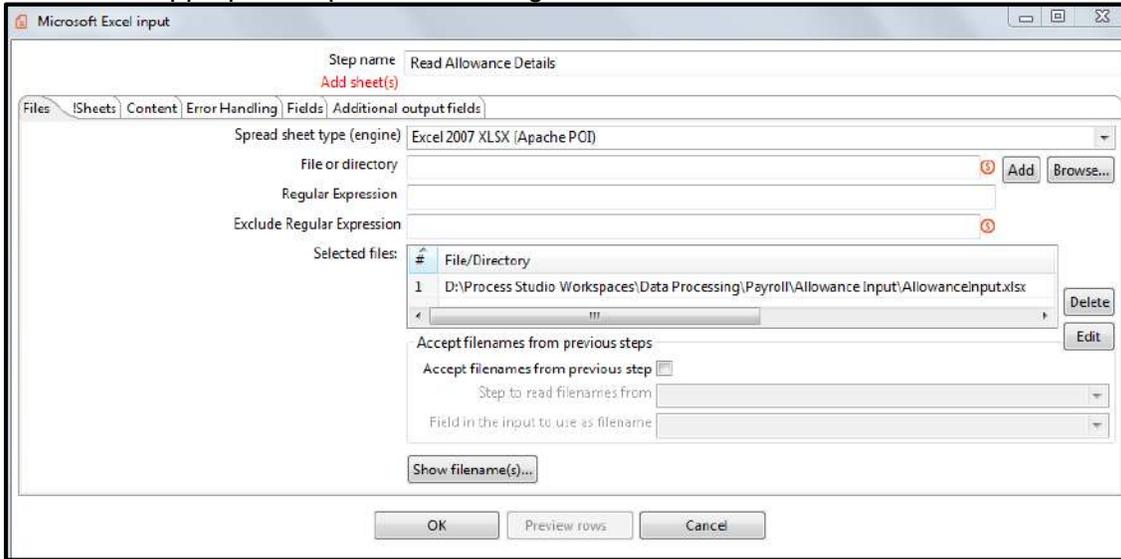
13. A pop up appears to confirm if merge join input is sorted. Acknowledge by clicking I understand.



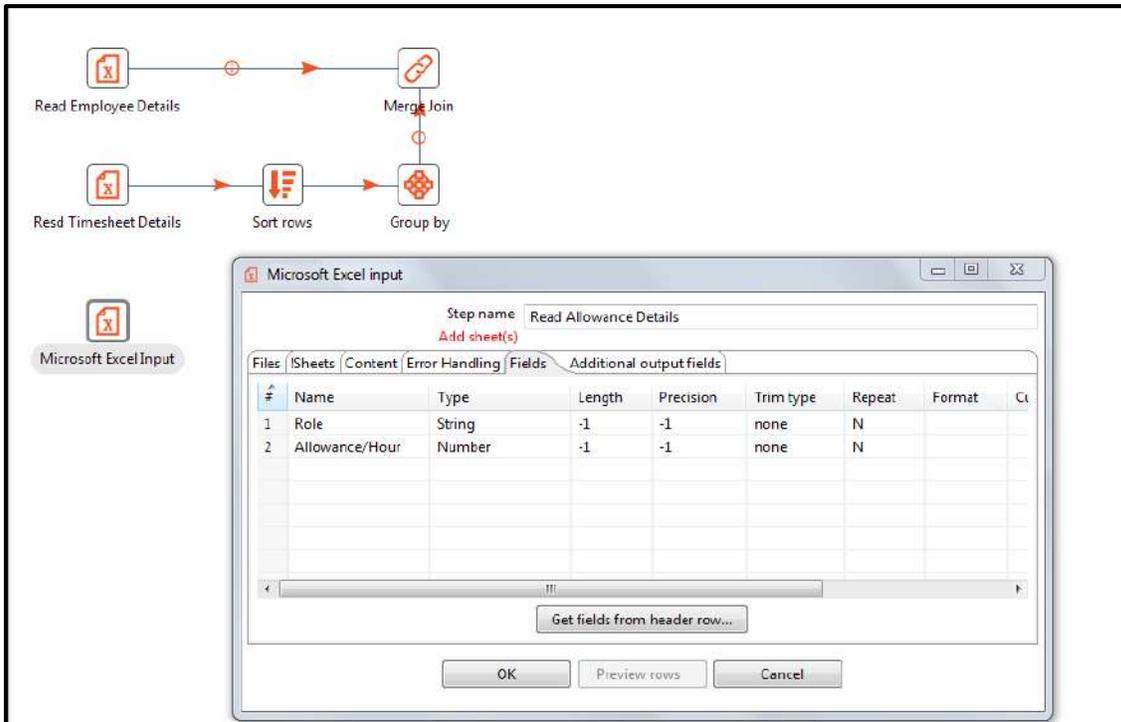
14. Drag and drop a Microsoft Excel input step. Name it as Read Allowance Details and configure it as follows.



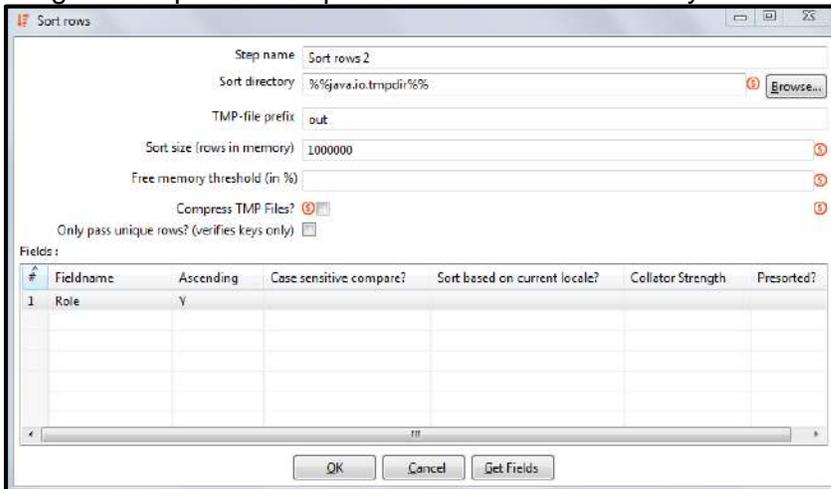
15. Choose an appropriate Spread Sheet engine.



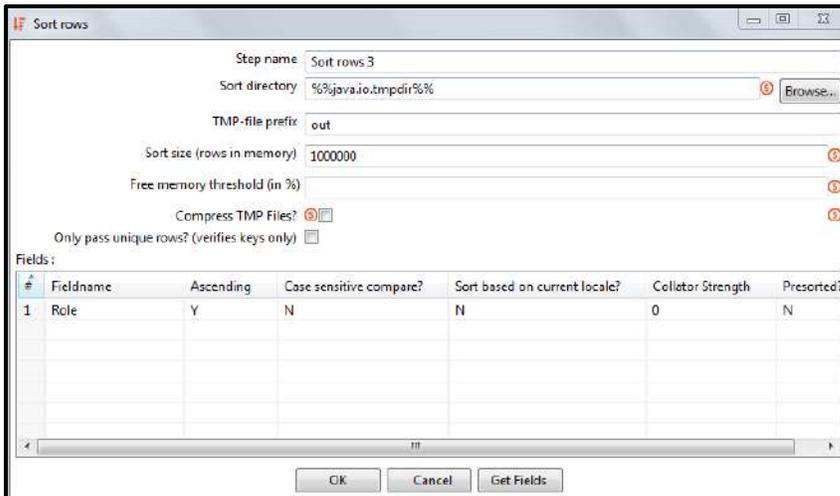
16. Go to Fields tab. Click Get Fields from header row.



17. Drag and drop a Sort step to sort Allowance details by role.



18. Drag and drop a Sort step to sort rows coming from the merge of employee and timesheet details.



19. Merge Employee wise timesheet details with role allowance details by role.

The screenshot shows a workflow diagram and a 'Merge Join' dialog box. The workflow consists of the following steps: Read Employee Details, Merge Join, Sort rows 3, MergeJoin 2, Read Timesheet Details, Sort rows, Group by, Read Allowance Details, and Sort rows 2. The 'Merge Join' dialog box is open, showing the following configuration:

- Step name: Merge Join 2
- First Step: Sort rows 3
- Second Step: Sort rows 2
- Join Type: INNER
- Keys for 1st step: Role
- Keys for 2nd step: Role

20. Create a calculator step to calculate employee salary by computing no. of hours worked and Allowance/hr.

The screenshot shows a workflow diagram and a 'Calculator' dialog box. The workflow consists of the following steps: Read Employee Details, Merge Join, Sort rows 3, MergeJoin 2, Calculator, Read Timesheet Details, Sort rows, Group by, Read Allowance Details, and Sort rows 2. The 'Calculator' dialog box is open, showing the following configuration:

- Step name: Calculator
- Fields table:

#	New field	Calculation	Field A	Field B	Field C	Value type	Length	Precision	Remove
1	Total_Salary	A * B	TotalHoursworked	Allowance/Hour					

21. Sort the employee salary by employee name.

The process flow for step 21 involves reading employee and timesheet details, merging them, sorting by role, reading allowance details, and finally calculating the total salary. The 'Sort rows' dialog box is configured as follows:

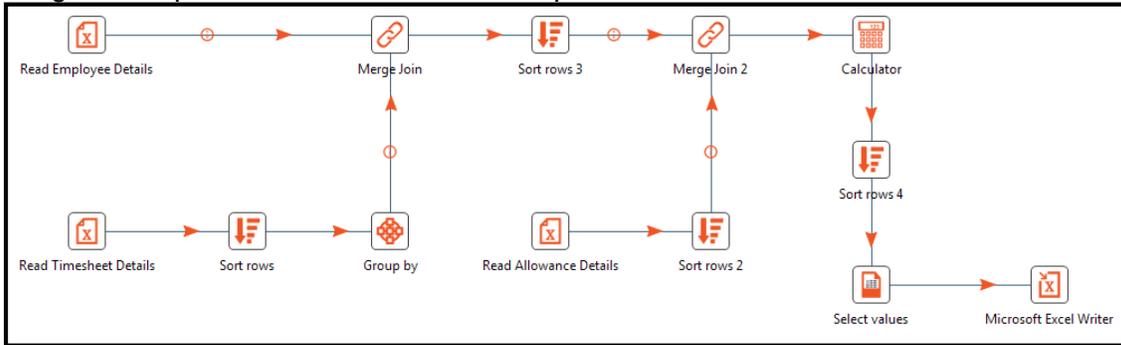
#	Fieldname	Ascending	Case sensitive compare?	Sort based on current locale?	Collator Strength	Presorted?
1	Role	Y	N	N	0	N

22. Use a Rename Fields (earlier Select Values) Step. Select the fields as shown below.

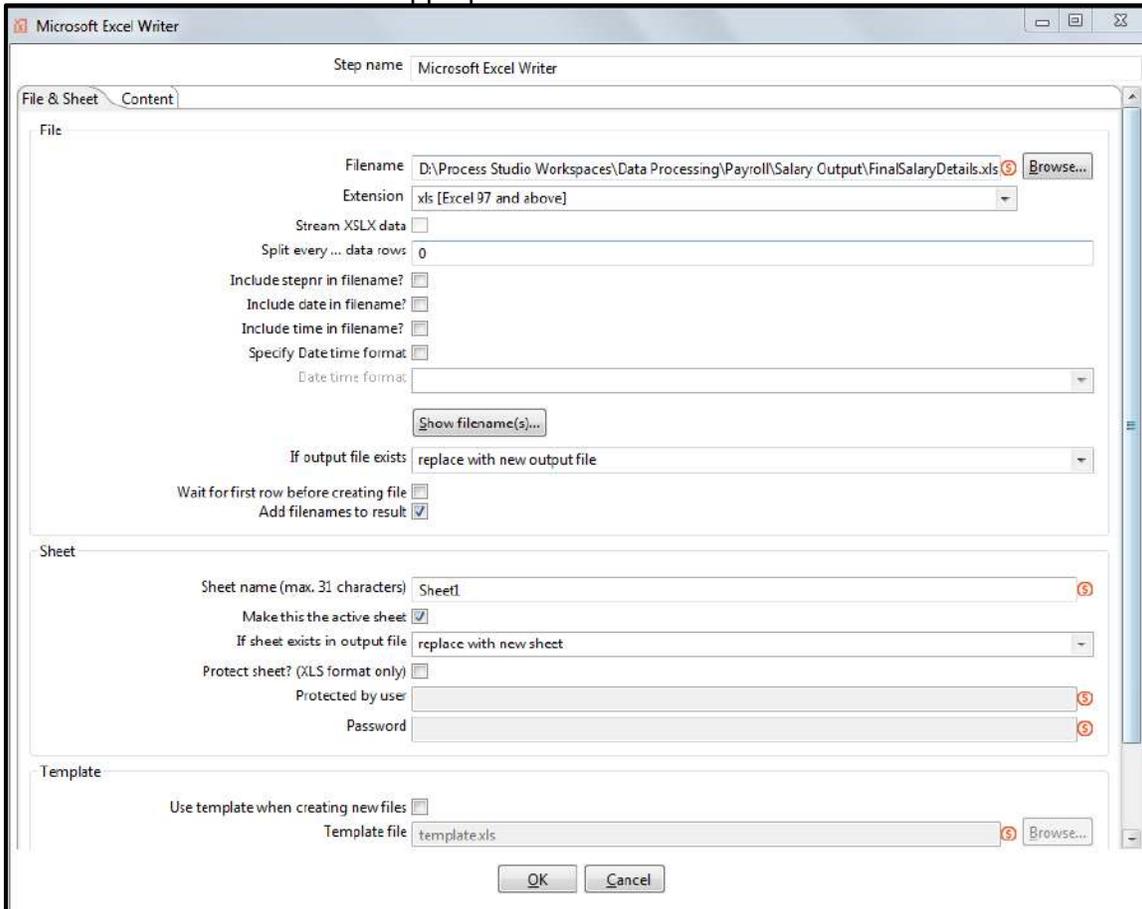
The process flow for step 22 is identical to step 21, but includes a 'Rename Fields' step at the end. The 'Select / Rename values' dialog box is configured with the following fields:

#	Fieldname	Rename to	Length	Precision
1	Name			
2	Id			
3	Role			
4	Department			
5	Date of Joining			
6	TotalHoursWorked			
7	Allowance/Hour			
8	Total_Salary			

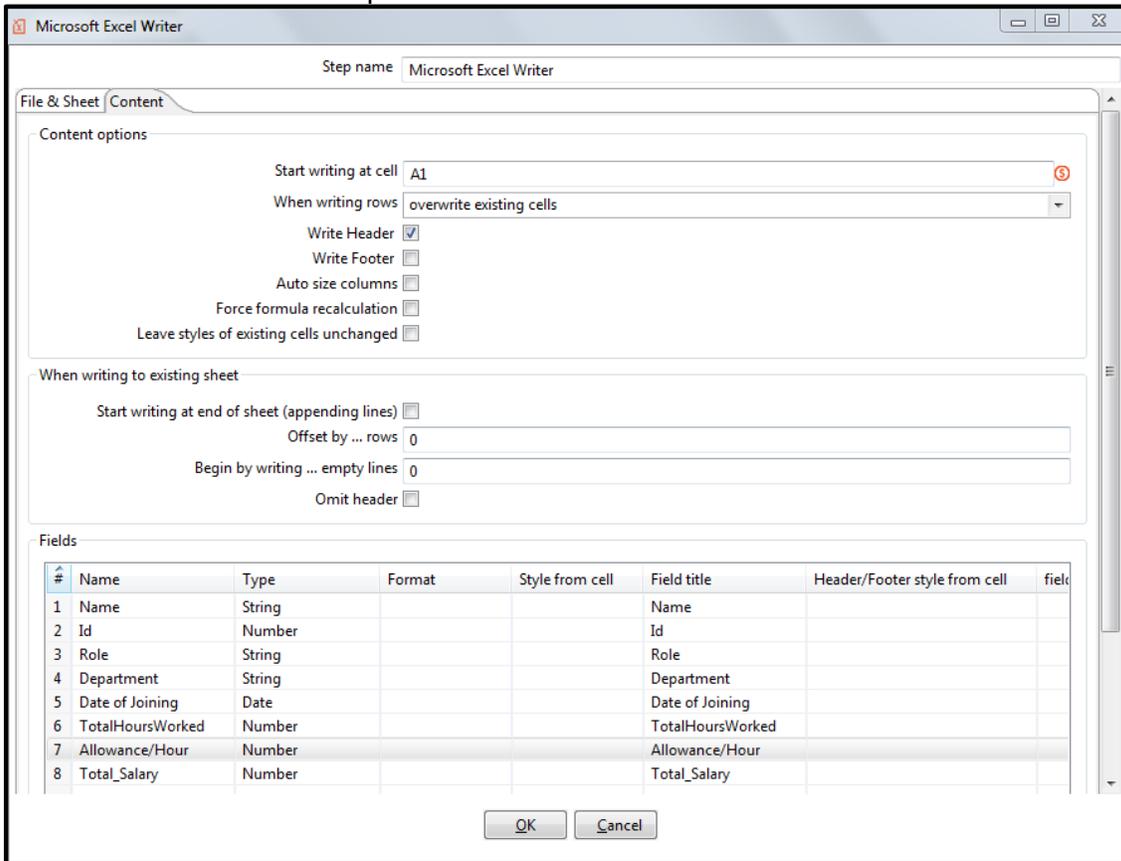
23. Drag and drop a Microsoft Excel Writer step to write to an excel file.



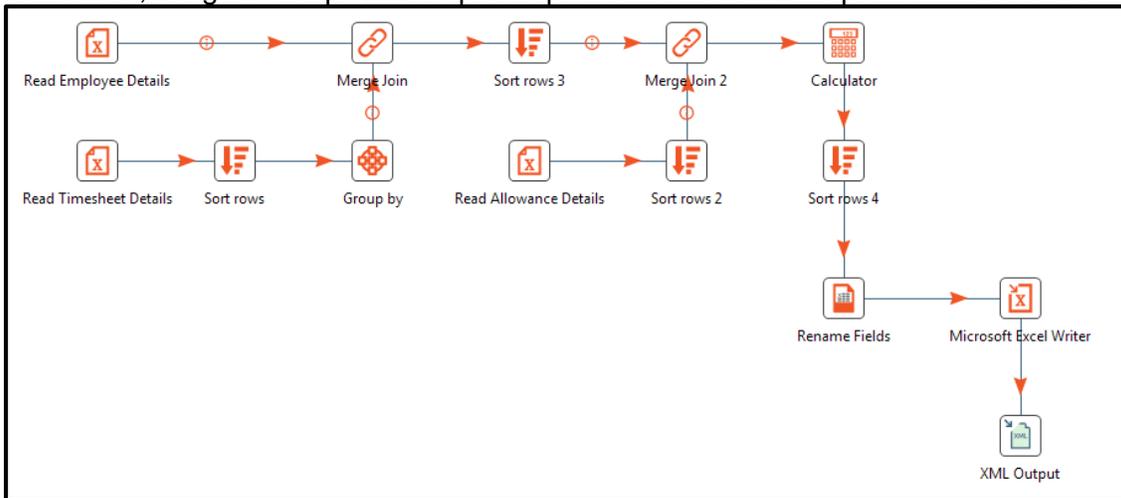
24. Browse a filename. Chose an appropriate extension as seen below.



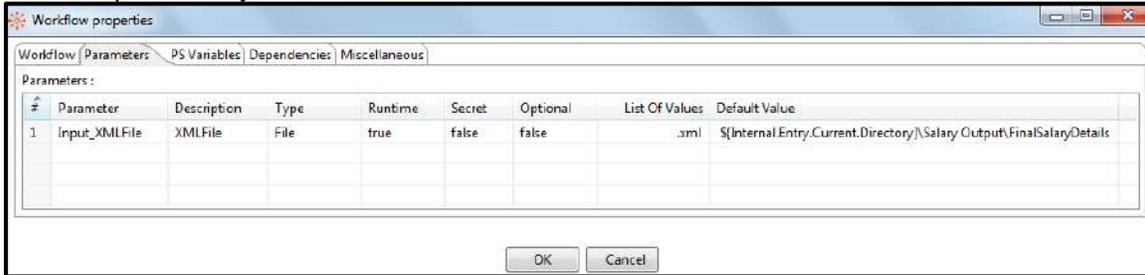
25. In Microsoft Excel Writer step in the Content Tab click on Get Fields.



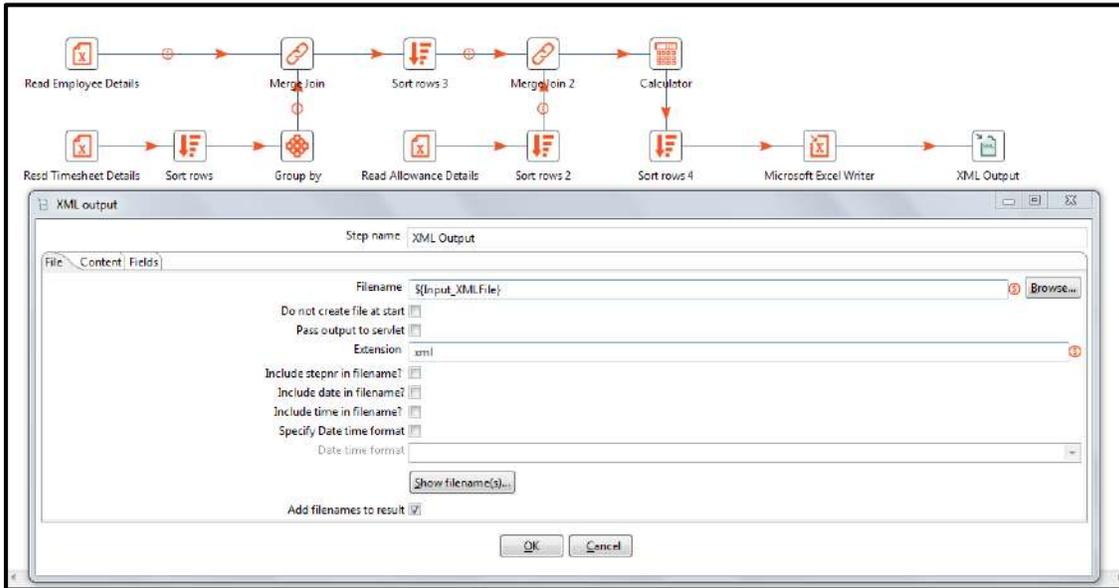
26. In addition, Drag and drop XML output step to create an XML output as seen below.



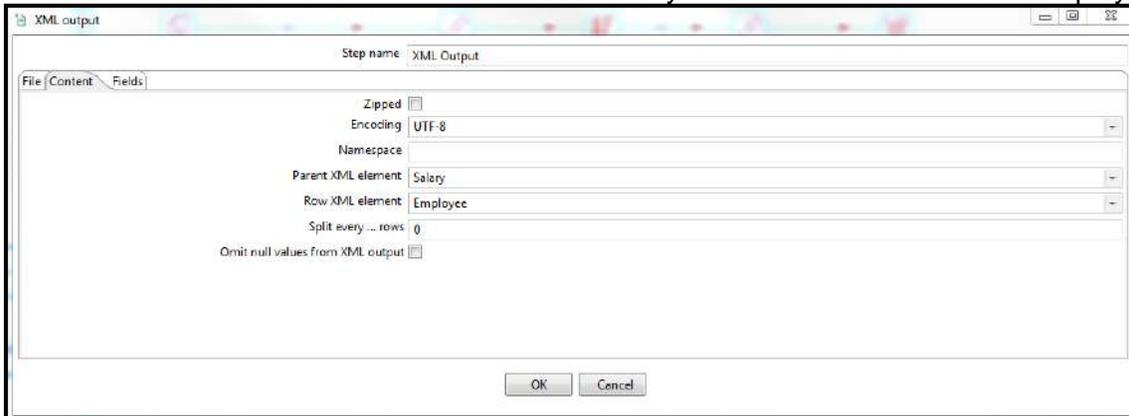
27. Before we configure the XML Output step, configure a workflow parameter as shown below for the ComputeSalary workflow.



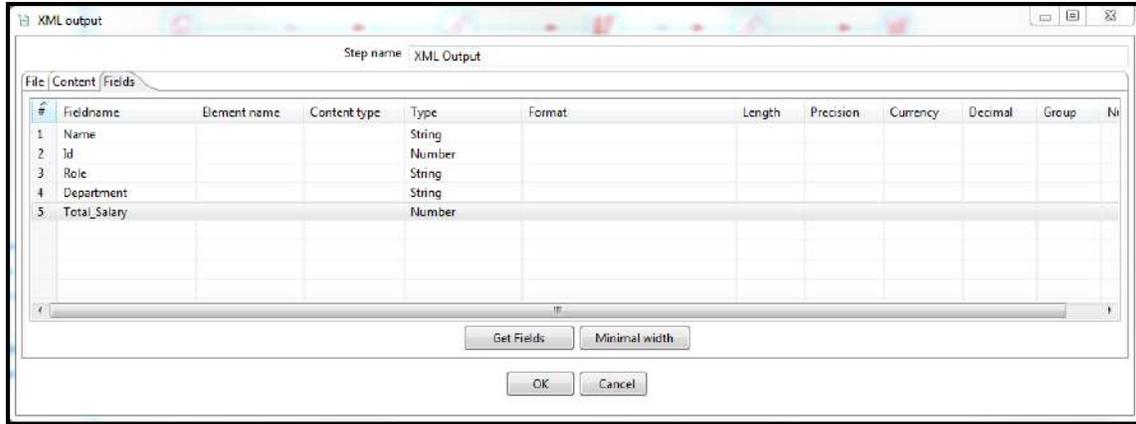
28. Configure XML Output step. Provide a parameterised filename with the parameter created above.



29. In the content tab set Parent XML element as Salary and Row XML element as Employee.



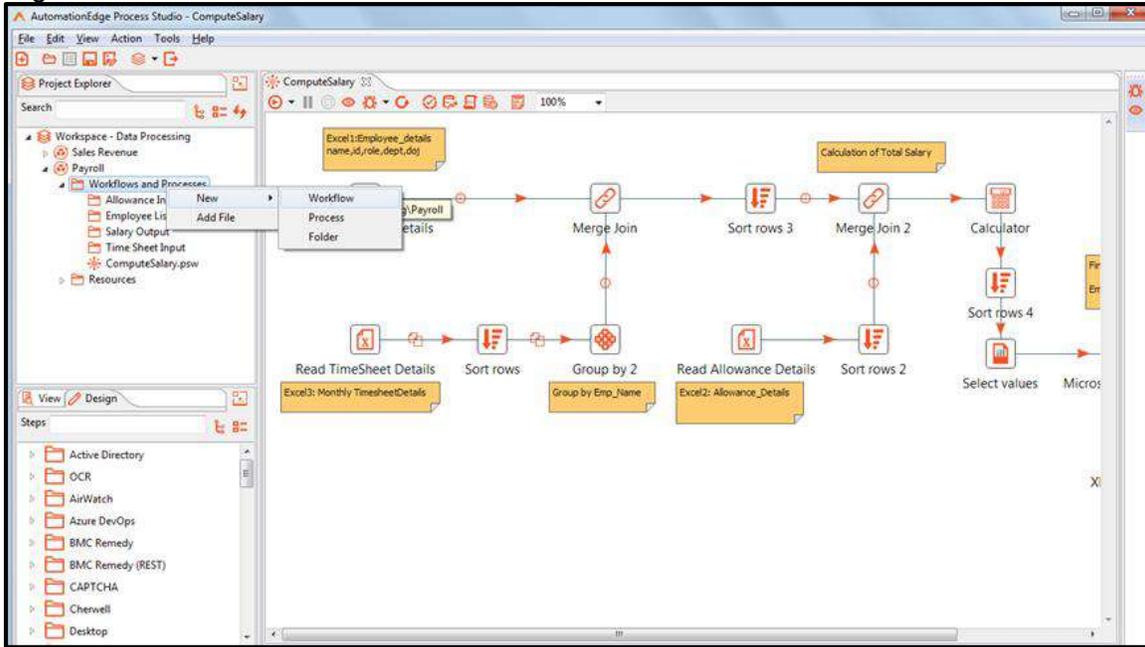
30. Select Get Fields in the Fields tab.



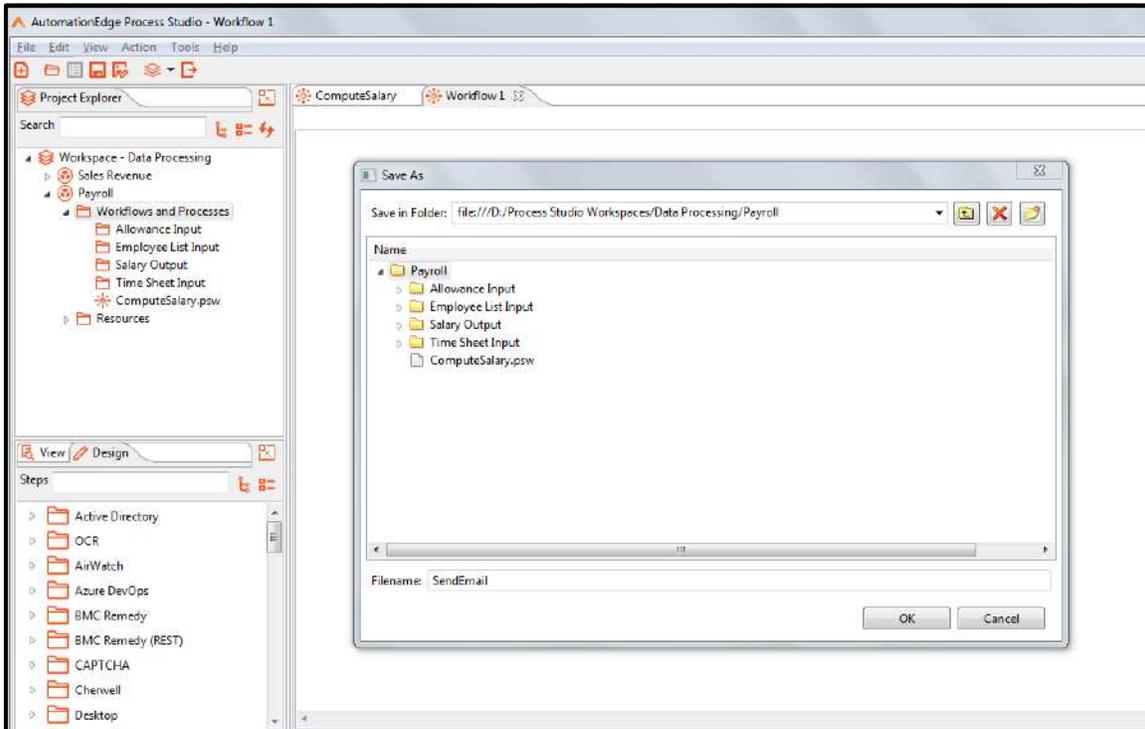
31. This completes Compute Salary workflow.

9.3 Send Mail Workflow

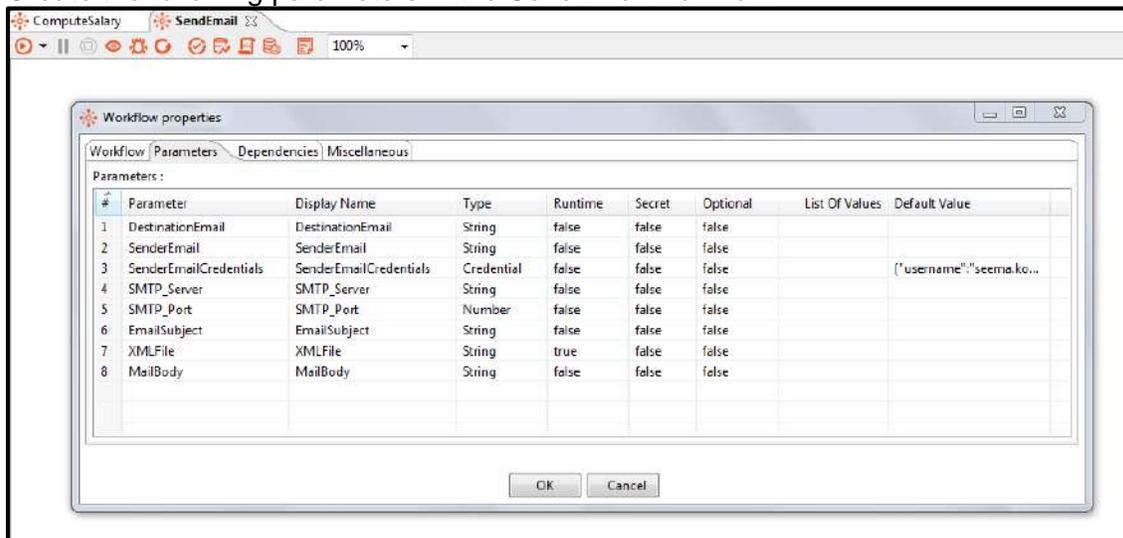
1. The screen shot below shows ComputeSalary workflow with Notes about the steps (You may right click on the canvas to add notes).
2. Right Click on Workflows and Processes to add a workflow.



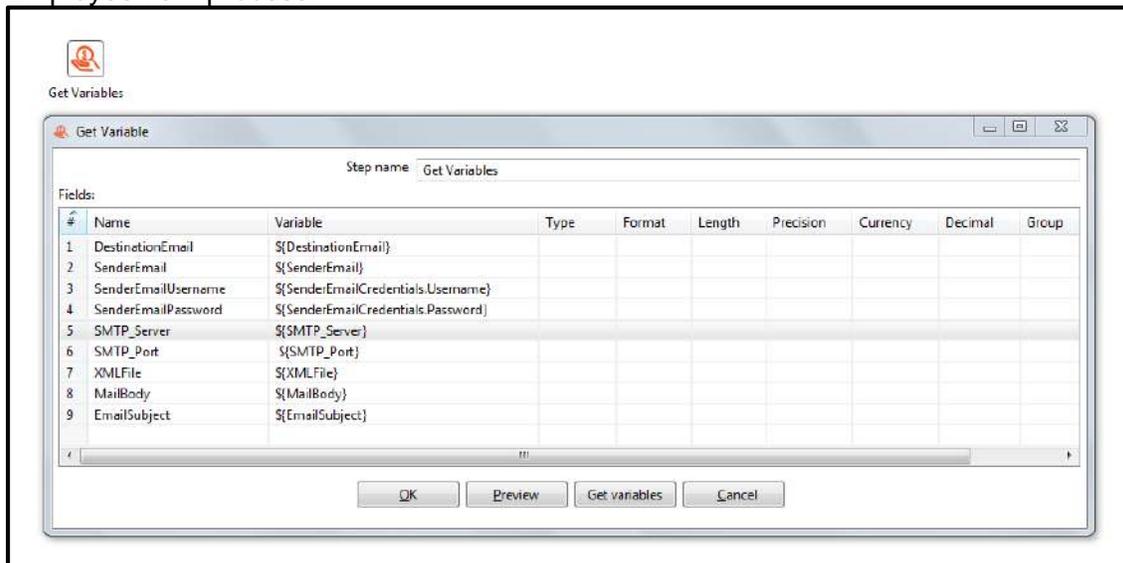
3. Create a workflow named SendMail.



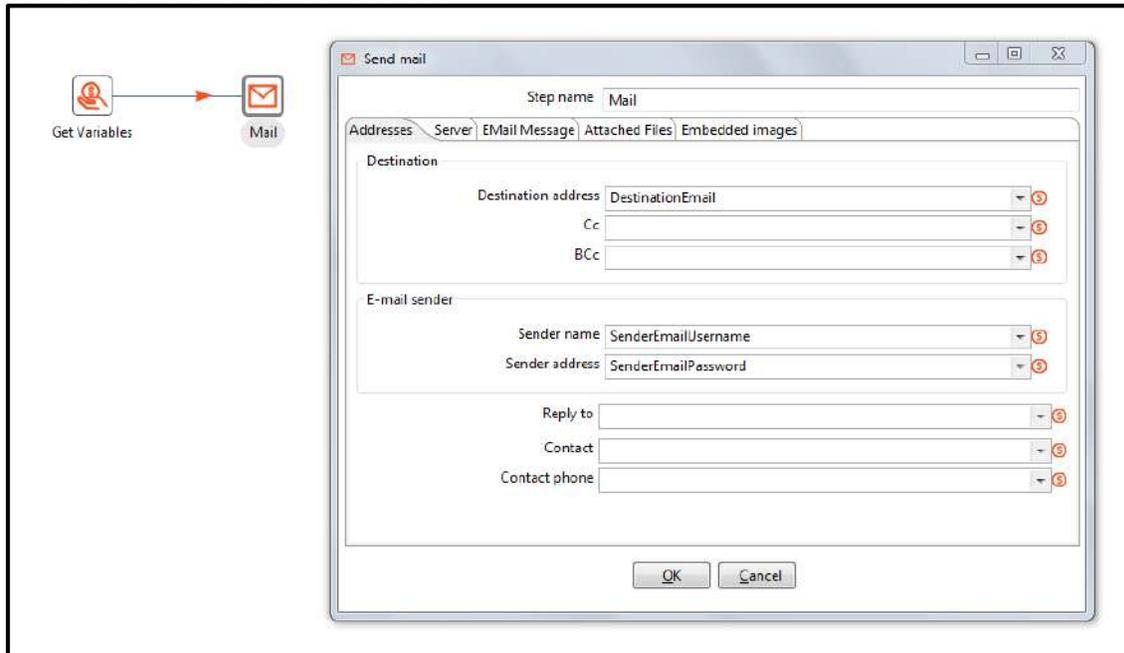
4. Create the following parameters in the SendEmail workflow.



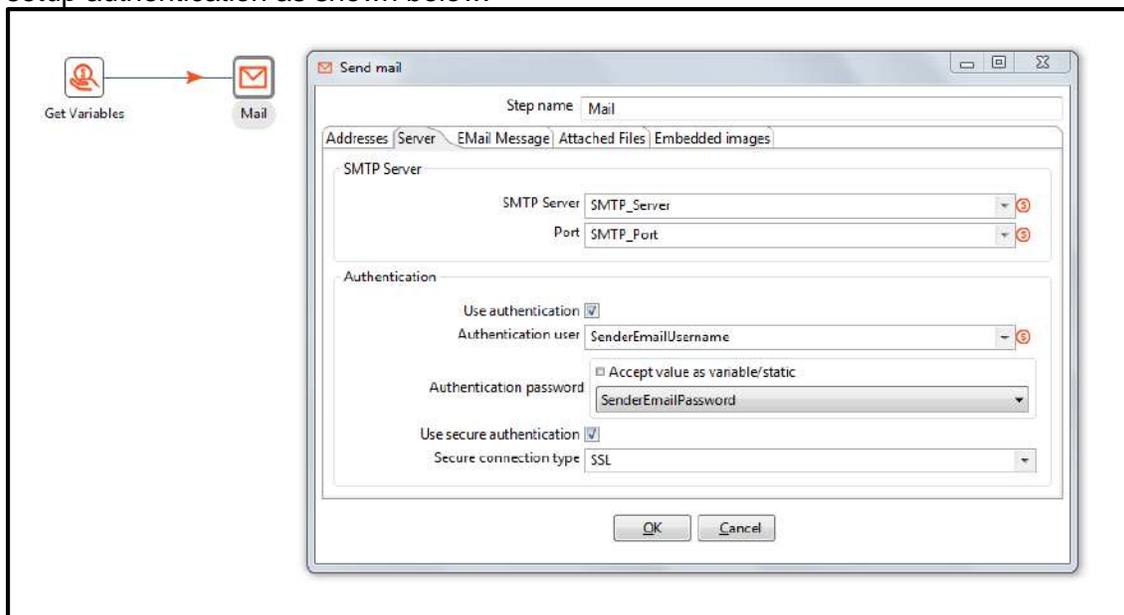
5. Drag and drop Get Variables Step. Click Get Variables to get the variables into the stream as fields. Map the variables to process parameters as shown below. You may complete the mapping after process parameters are defined in the following section in which we create the EmployeeMain process.



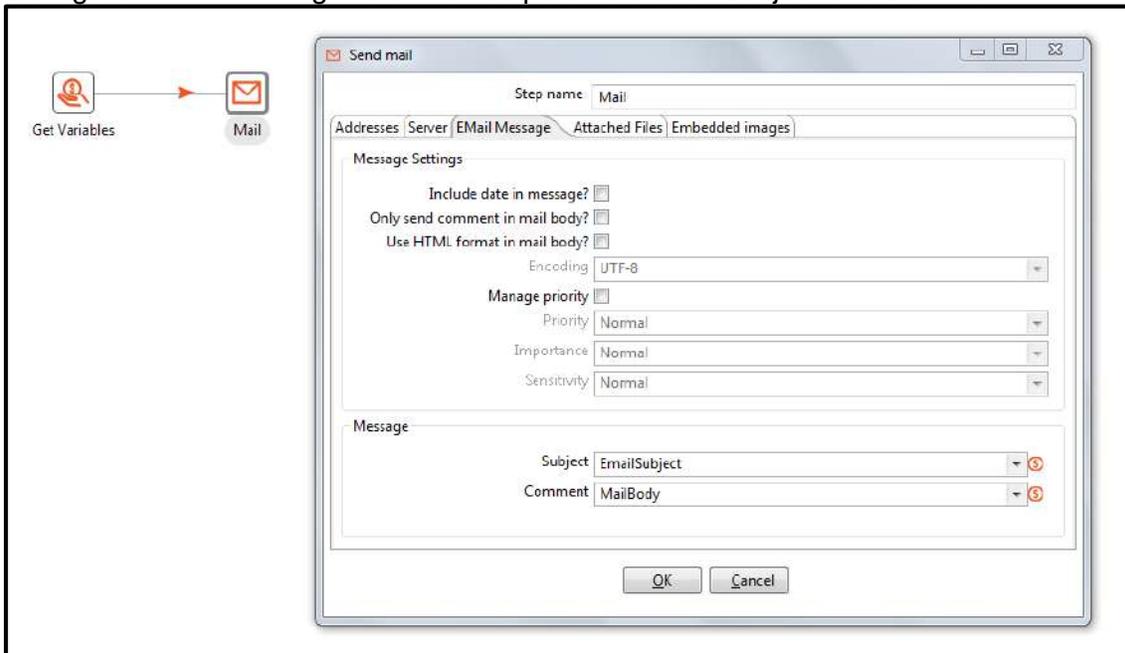
6. Drag and drop Mail step. Provide values in the form of parameters for Destination address, Sender name and Sender address.



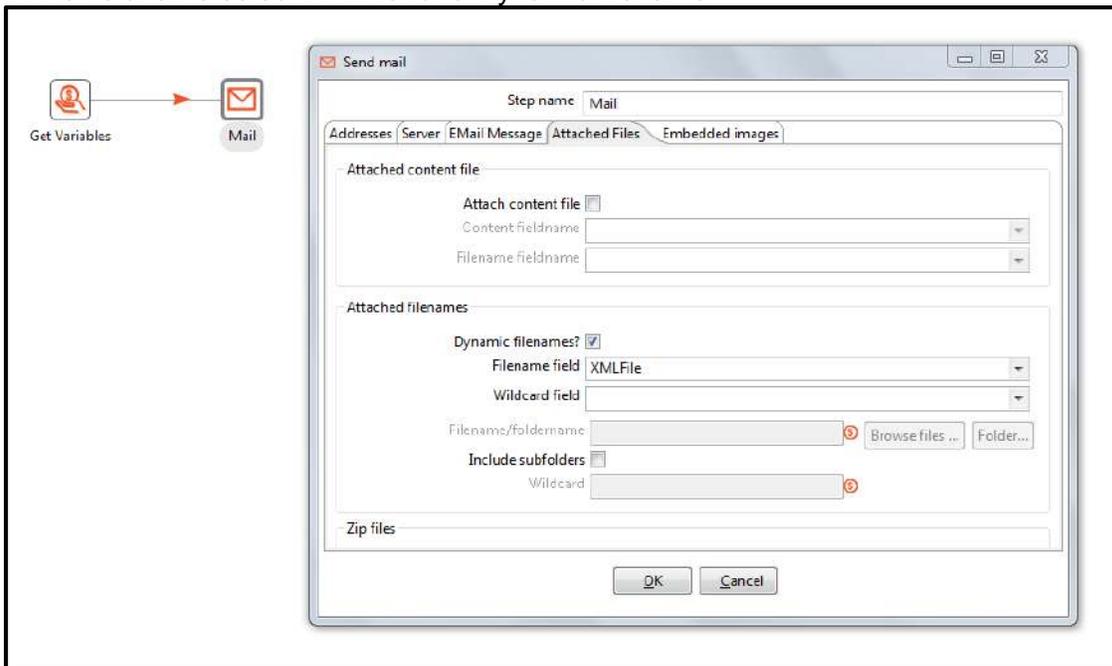
7. Configure Server properties by providing values in the form of parameters as shown below. Also setup authentication as shown below.



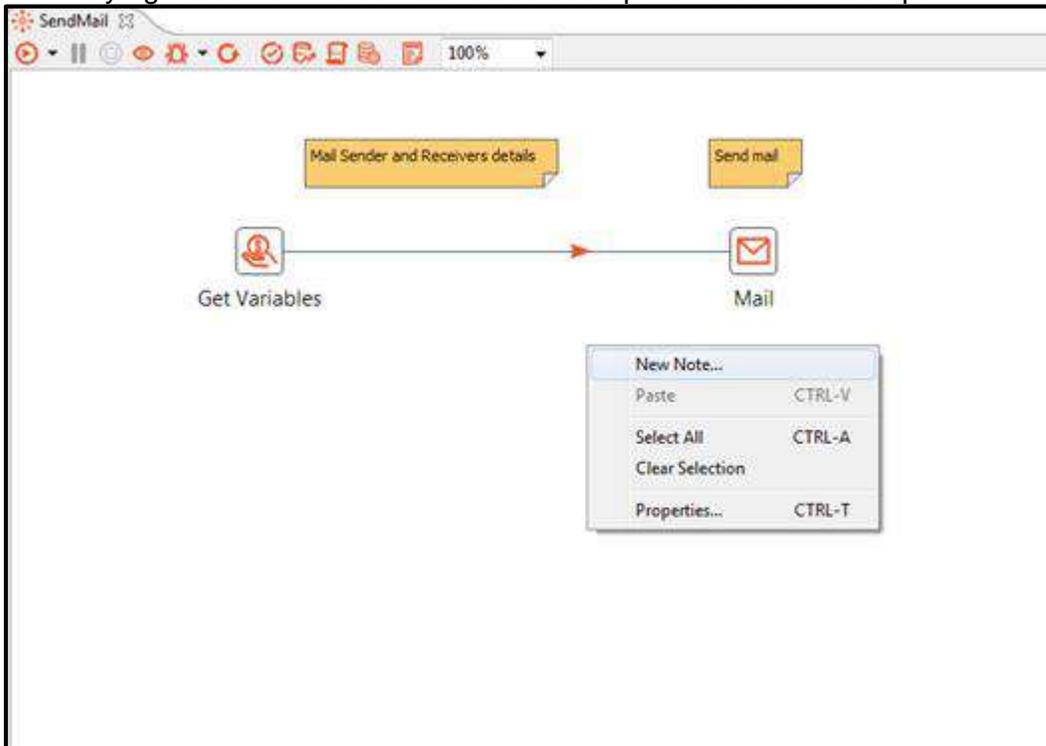
8. Configure 'EMail Message' tab. Provide parameters for Subject and Comment.



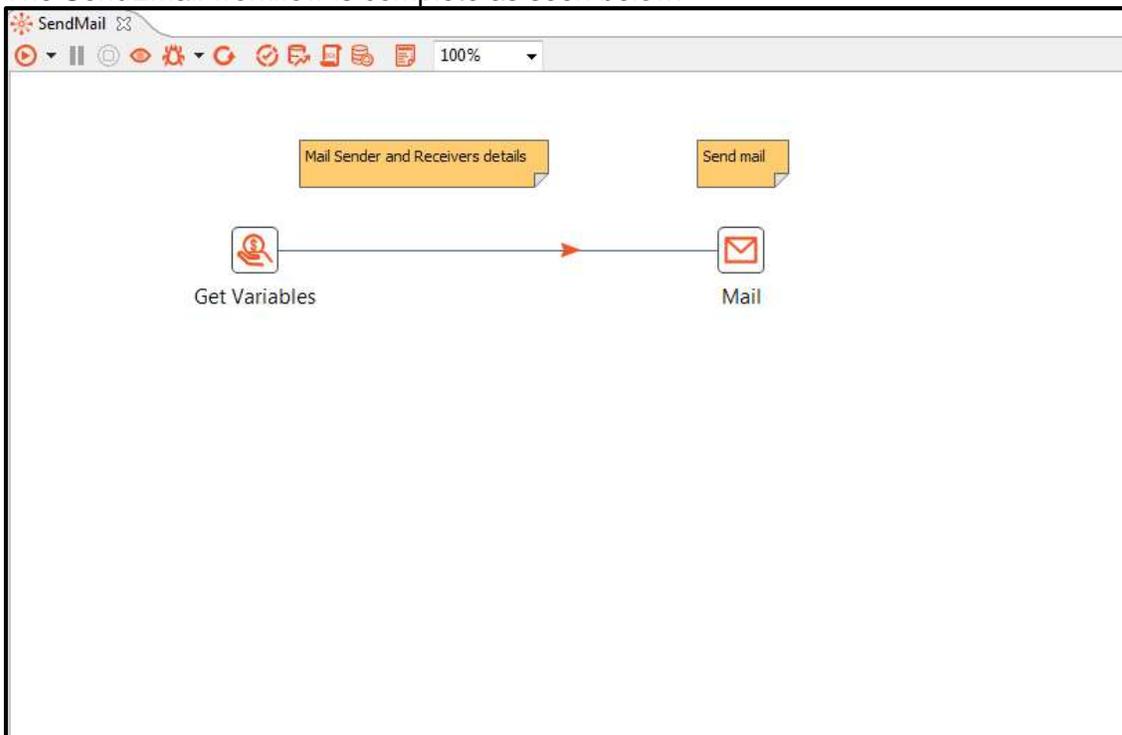
9. In the filename select XMLfile for a Dynamic filename.



10. You may right click on the canvas to add descriptive notes for the steps.

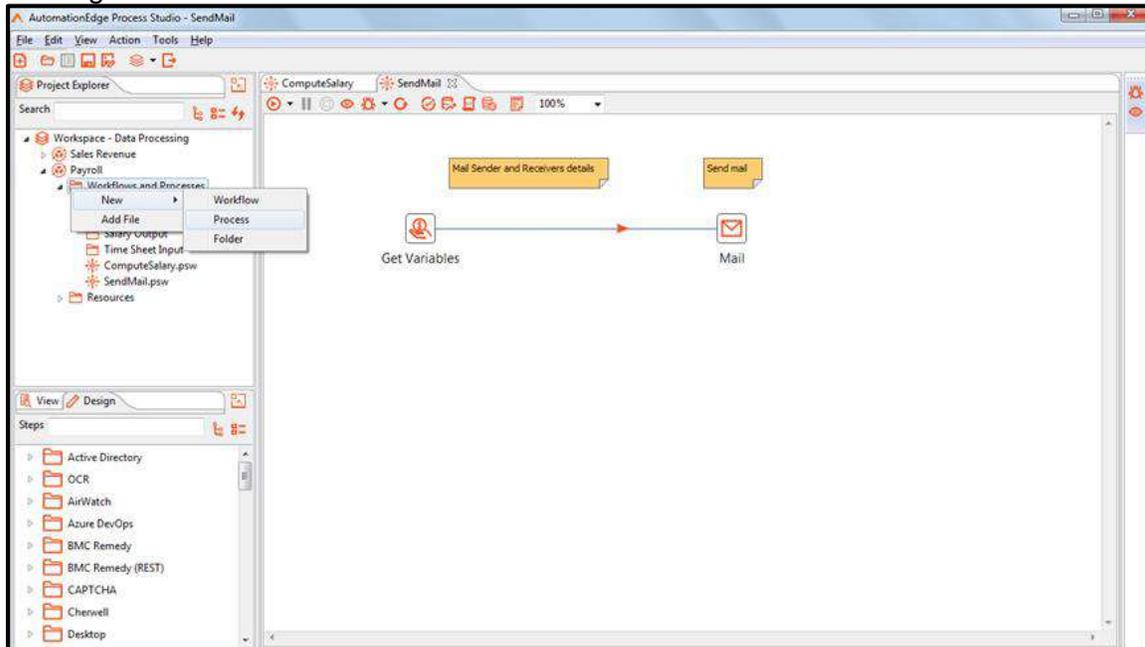


11. The SendEmail workflow is complete as seen below.

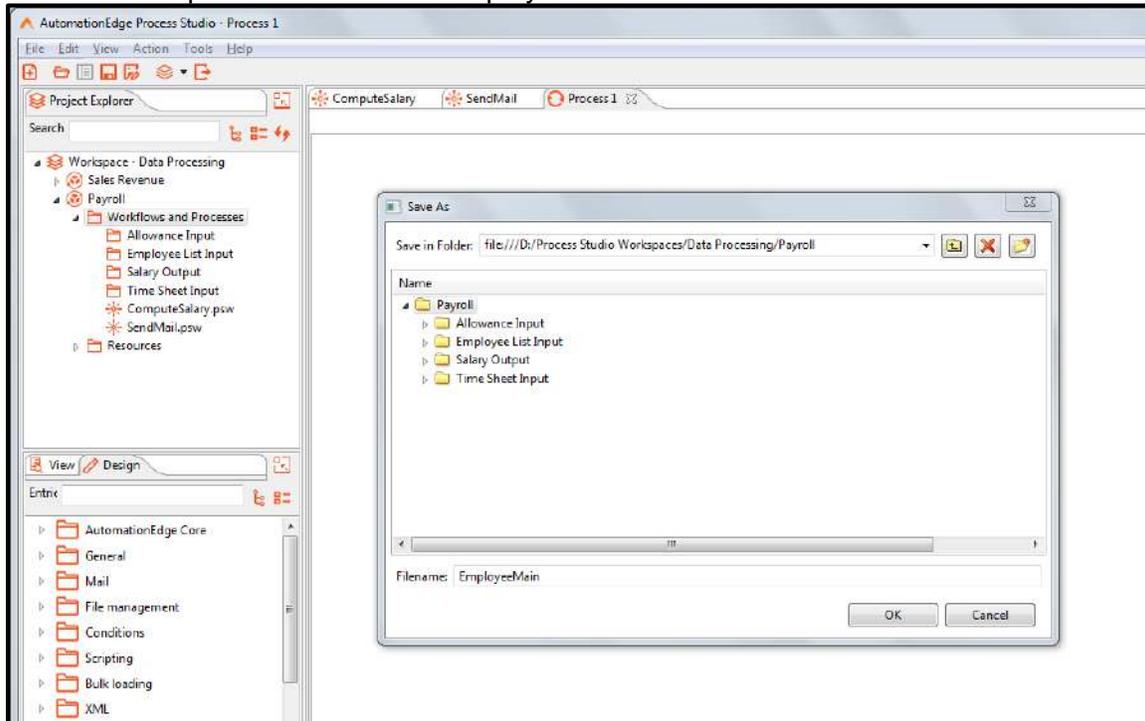


9.4 Employee Main Process

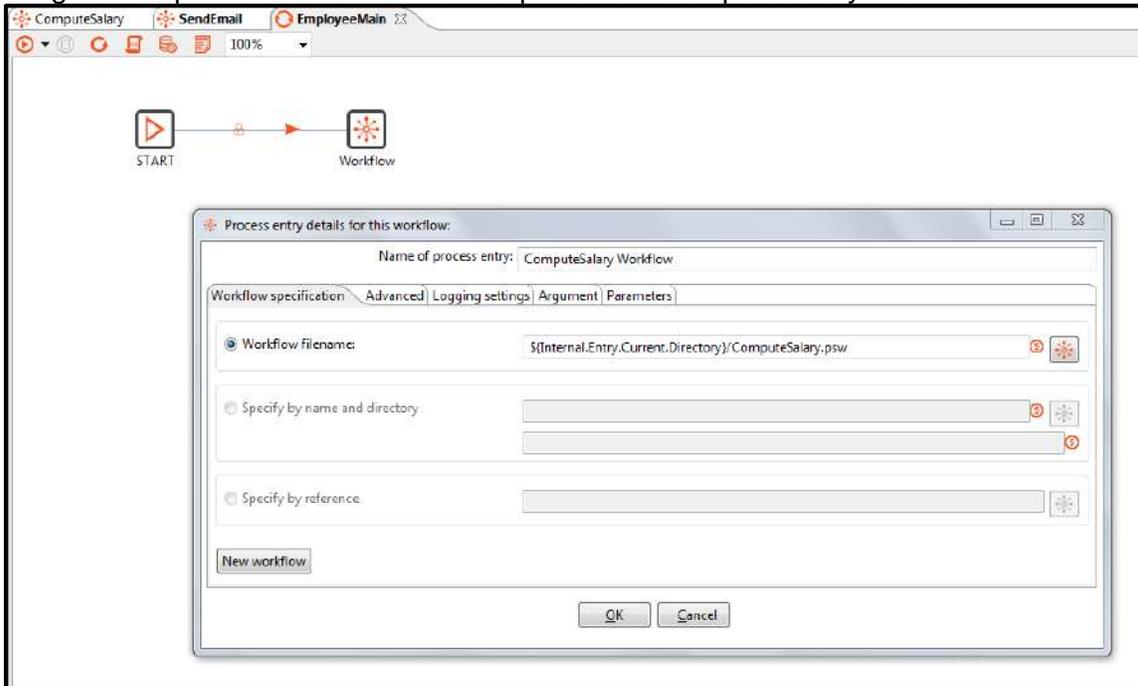
1. Now right click on Workflows and Processes to add a Process.



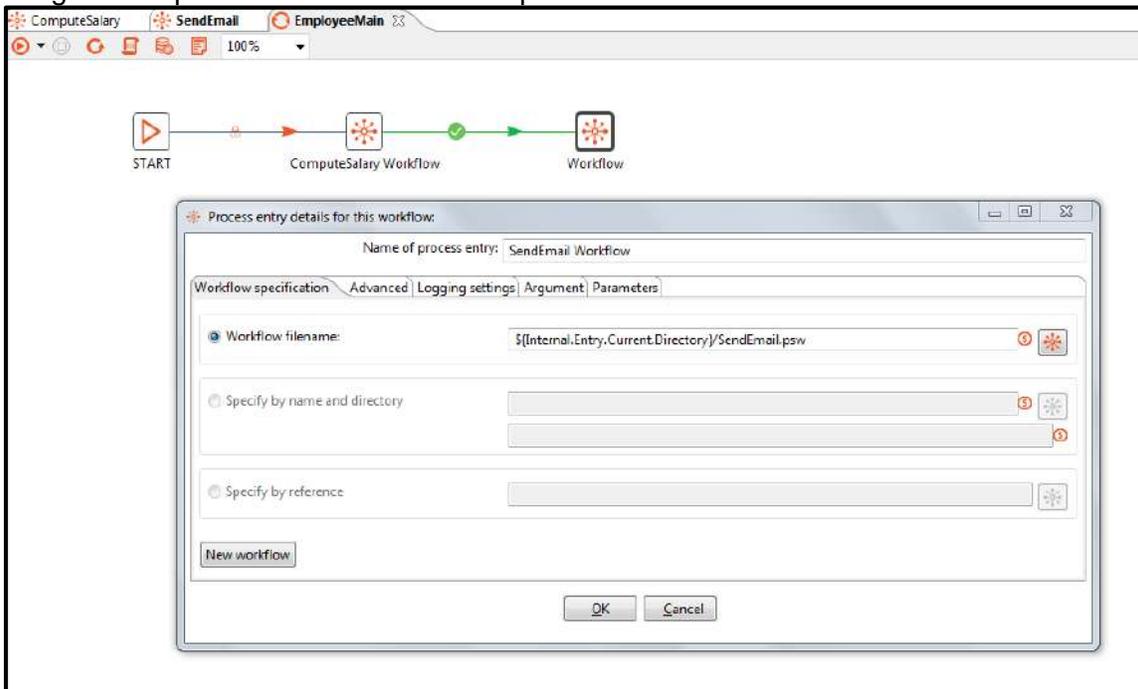
2. Create a new process. Save it as EmployeeMain.



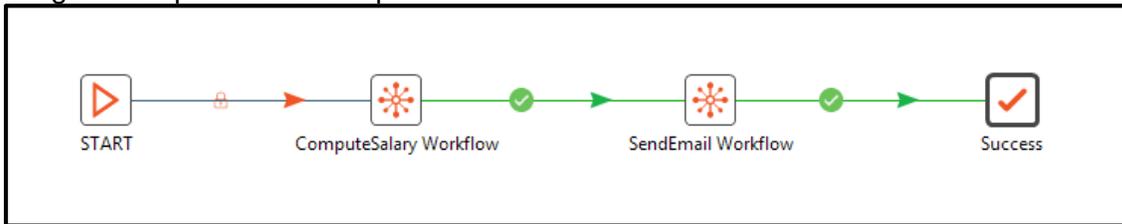
3. Drag and drop a start step.
4. Drag and drop an Execute Workflow step. Name it ComputeSalary Workflow.



5. Drag and drop an Execute Workflow step. Name it SendMail Workflow.



6. Drag and drop a success step.



7. Set process parameters as shown below.

8. This is the main process and all input parameters need to be defined here.

AutomationEdge Process Studio - EmployeeMain

Project Explorer

- Workspace - Data Processing
 - Sales Revenue
 - Payroll
 - Workflows and Processes
 - Allowance Input
 - Employee List Input
 - Salary Output
 - Time Sheet Input
 - ComputeSalary.psw
 - EmployeeMain.psp
 - SendEmail.psw
- Resources

View Design

Process properties

Process [Parameters - PS Variables]

#	Parameter	Description	Type	Runtime	Secret	Optional	List Of Values	Default value
1	DestinationEmail	DestinationEmail	String	true	false	false		seema.kothari@v
2	SenderEmail	SenderEmail	String	true	false	false		seema.kothari@v
3	SenderEmailCredentials	SenderEmailCredentials	Credential	true	false	false		{"username": "see
4	SMTP_Server	SMTP_Server	String	false	false	false		
5	SMTP_Port	SMTP_Port	Number	false	false	false		
6	EmailSubject	EmailSubject	String	false	false	false		
7	XMLFile	XMLFile	String	true	false	false		
8	MailBody	MailBody	String	false	false	false		

Credential Values

Credential Attribute Values

Username: seema.kothari@vvyon

Password: *****

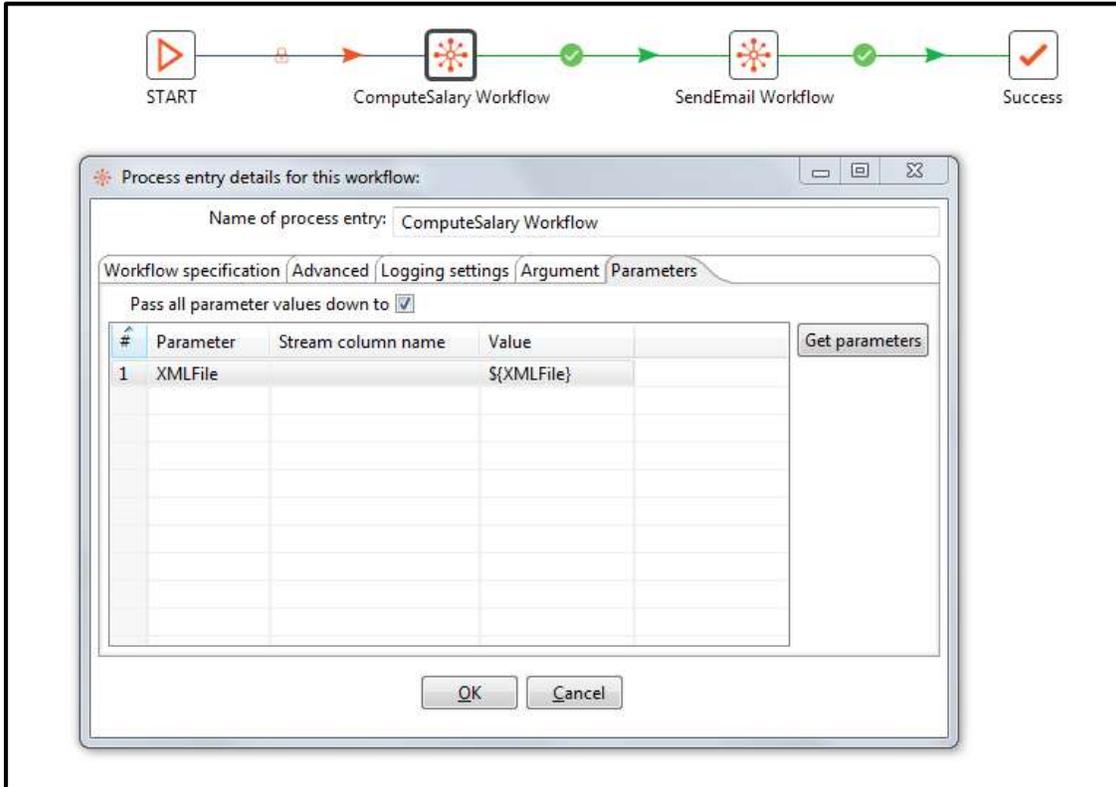
Parameter1:

Parameter2:

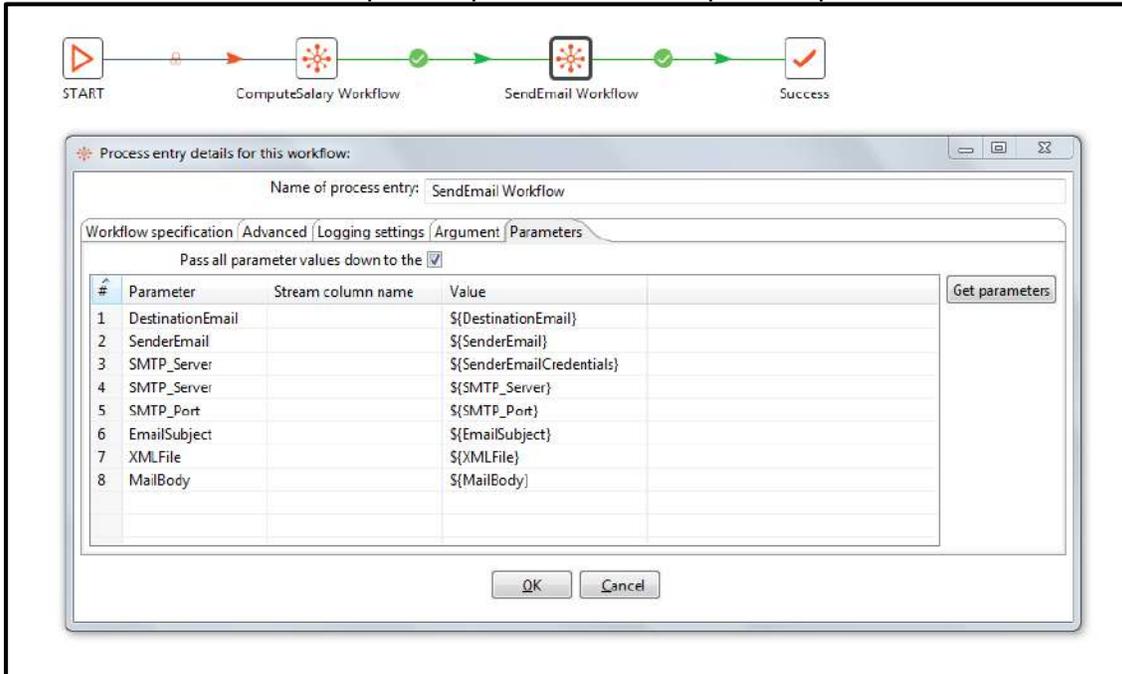
Encrypted1:

Encrypted2:

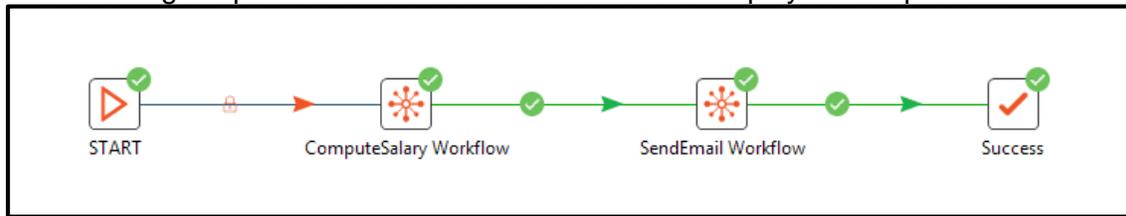
- Double click Compute Salary Workflow. Click Get Parameters and map it to a process parameter as shown below.



- Double click Send Mail Workflow executor step. Click Get parameters. It gets the parameters of the Send Mail Workflow. Map these parameters to the process parameters as shown.



- The following snapshot shows a successful run of the EmployeeMain process.



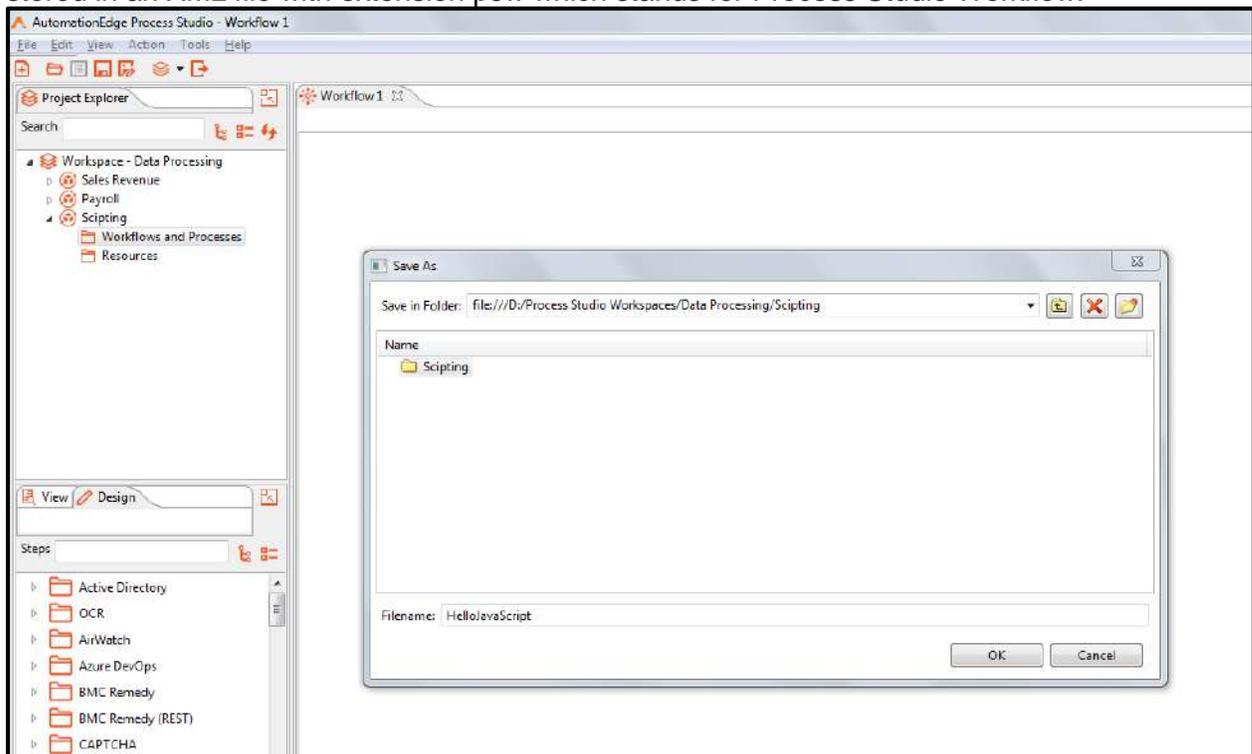
10 Project 7: Scripting

Modified Java Script Value step, is an easy, expression based user interface for building JavaScript expressions. We have introduced this step here as it is used commonly.

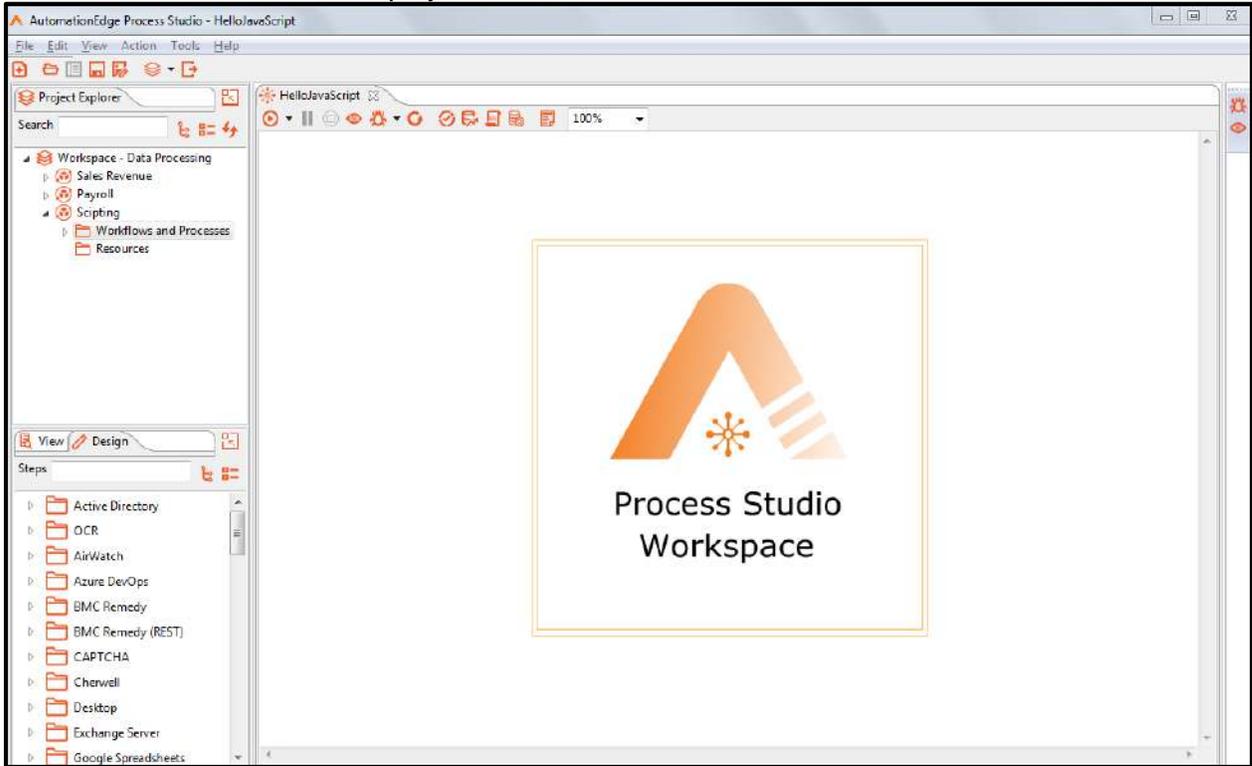
10.1 Building your first scripting workflow HelloJavaScript

We will build a workflow with **Modified Java Script** step to Add Two Numbers and Multiply two numbers. Following, are the steps to create the workflow HelloJavaScript.

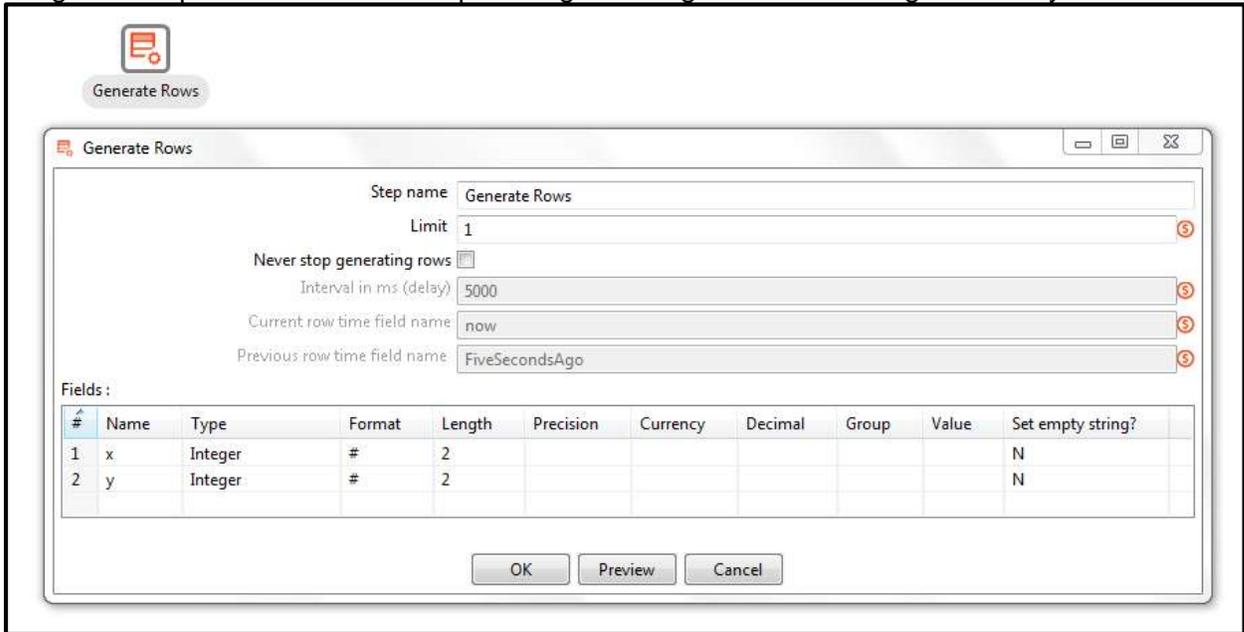
- Create a new Project in the Data Processing Workspace. Name the project Scripting.
- Right click on the project to create a new workflow. Name the workflow as **HelloJavaScript** and save it at an appropriate location. The default location is in the project directory. The Workflow is stored in an XML file with extension psw which stands for Process Studio Workflow.



- Type **Generate Rows** in the text box next to **Steps** in the left pane. All the steps having **Generate Rows** in their name will be displayed.



- Drag and Drop **Generate Rows Step**. Configure it to generate two integers **x** and **y**.



5. Drag and Drop a Modified Java Script Value.



6. Java Script Step shows the two input fields x and y coming from Generate Rows step. Write a java script to multiply two numbers. Click Get Variables button to get all the variables defined in the script in the bottom section fields.

The screenshot shows the 'Script Values / Mod' dialog box. The 'Step name' is 'Modified Java Script Value'. The 'Java script functions' tree on the left shows 'Input fields' with sub-items 'x' and 'y'. The 'Java script' area contains the following code:

```

//Script here
var a=5;
var b=10;

var x=50;
var y=100;

var c = a+b;
var w=x+y;

display(a,b);
display1(x,y);

Alert("Mutiplication: " + multiplication(a,b));
Alert("Mutiplication: " + multiplication(x,y));

function display(a,b)
{
    Alert("A = " + a + "\nB = " + b);
}

function display1(x,y)
{
    
```

Below the script editor, the 'Fields' table is visible:

#	Fieldname	Rename to	Type	Length	Precision	Replace value 'Fieldname' or 'Rename to'
1	a		Number	16	2	N
2	b		Number	16	2	N
3	c		Number	16	2	N
4	w		Number	16	2	N
5	x		Number	16	2	Y

At the bottom of the dialog are buttons for 'OK', 'Cancel', 'Get variables', and 'Test script'.

7. Following is a sample code.

```
//Script here

var a=5;
var b=10;

var x=50;
var y=100;

var c = a+b;
var w=x+y;

display(a,b);
display1(x,y);

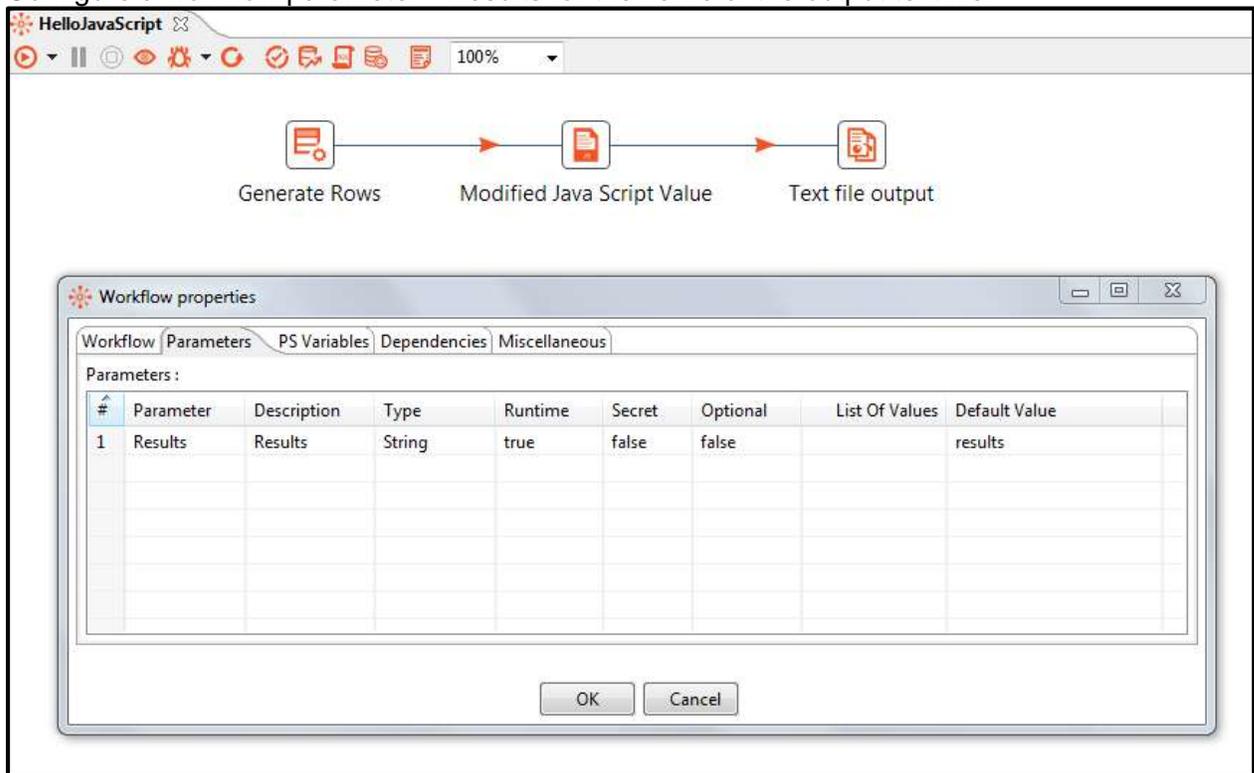
Alert("Mutiplication: " + multiplication(a,b));
Alert("Mutiplication: " + multiplication(x,y));

function display(a,b)
{
    Alert("A = " + a + "\nB = " + b);
}

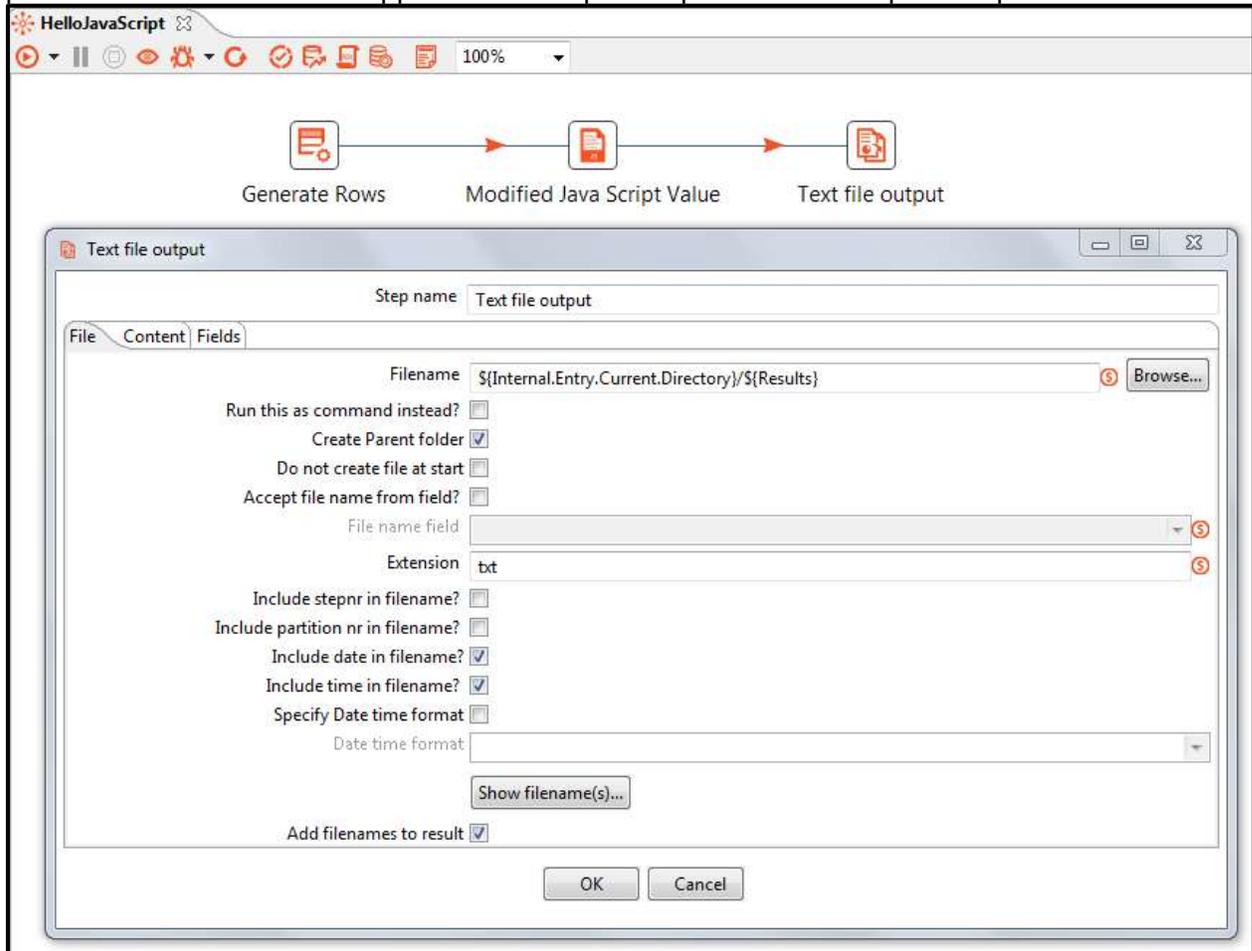
function display1(x,y)
{
    Alert("X = " + x + "\nY = " + y);
}

function multiplication(a,b)
{
    return a * b;
}
```

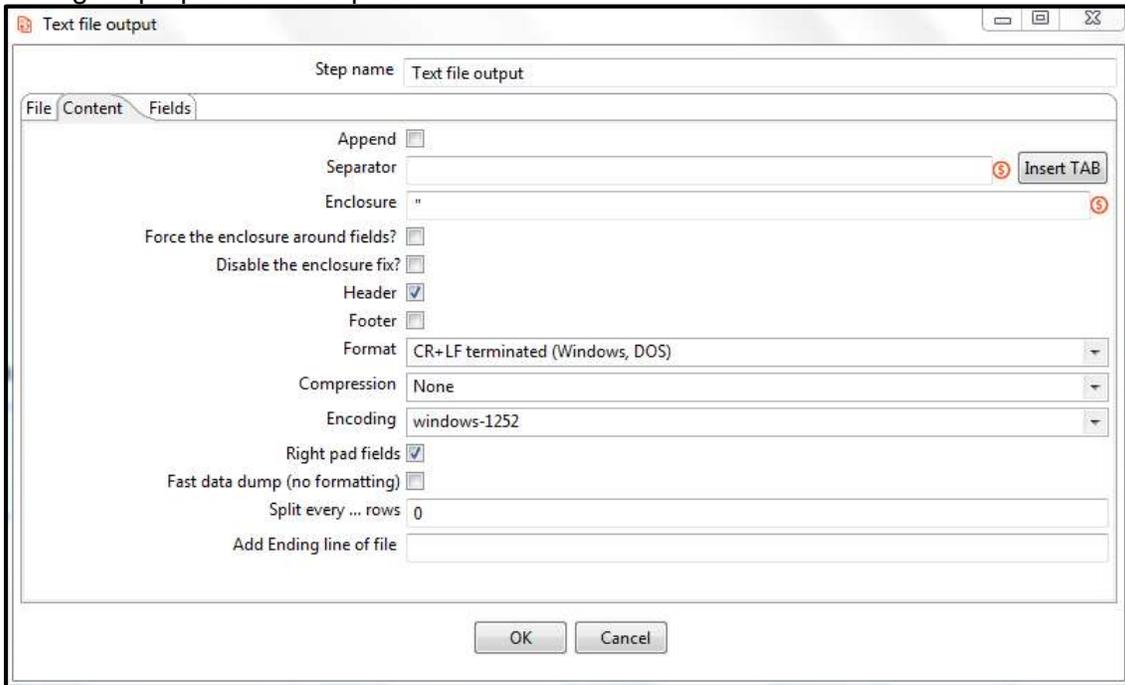
8. Configure a workflow parameter - Results for the name of the output text file.



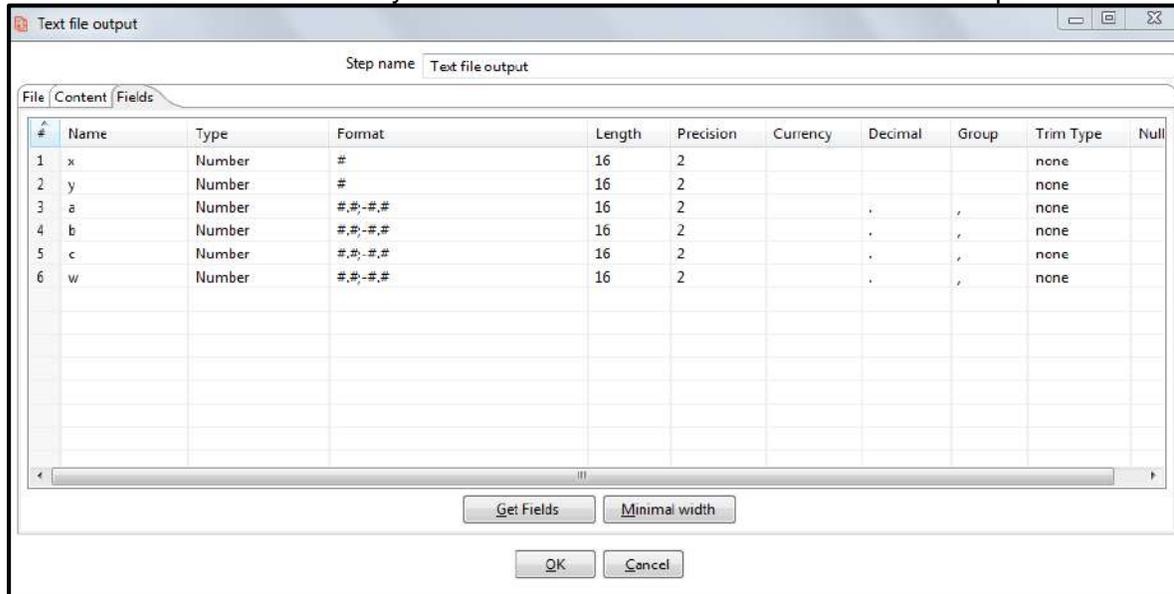
9. Configure the Text file output step to capture all or required fields in the stream. Notice the parameter Results which is appended to the path to provide the complete filepath.



10. Configure properties as required.



11. This screen shot shows x and y variables have been chosen for Text File Output.



12. The screenshot below shows the workflow execution results.

The screenshot displays the execution results of a workflow named 'HelloJavaScript'. The workflow consists of three steps: 'Generate Rows', 'Modified Java Script Value', and 'Text file output', all of which are completed successfully, as indicated by green checkmarks. A 'results_...' window is open, showing the output of the workflow: 'x;y;a;b;c;w' and '50;100;5;10;15;150'. The 'Execution Results' table below shows the data for the first row.

#	x	y	a	b	c	w	display
1	50	100	5	10	15	150	function display(a, b) { Alert("A =...

13. A sample workflow using Java Script step is complete.

V Workspace for Tutorials

In the Tutorial workspace we will create the following projects, with sample workflows.

- Spy
 - GUI Spy for Web
 - GUI Spy for Windows
 - Desktop Spy
 - Surface Spy
- Try-Catch
- Debugging
- Git Repository Integration
- SVN Repository Integration

Follow the steps below, and in the respective sections below,

1. Create a new folder for the workspace – Tutorials in the file system.
2. Start Process studio application. Browse for the Tutorials folder and click Launch button to launch the Tutorials workspace.

11 Project 8: GUI Spy for Web

11.1.1 Sample Workflow with GUI Spy for Web

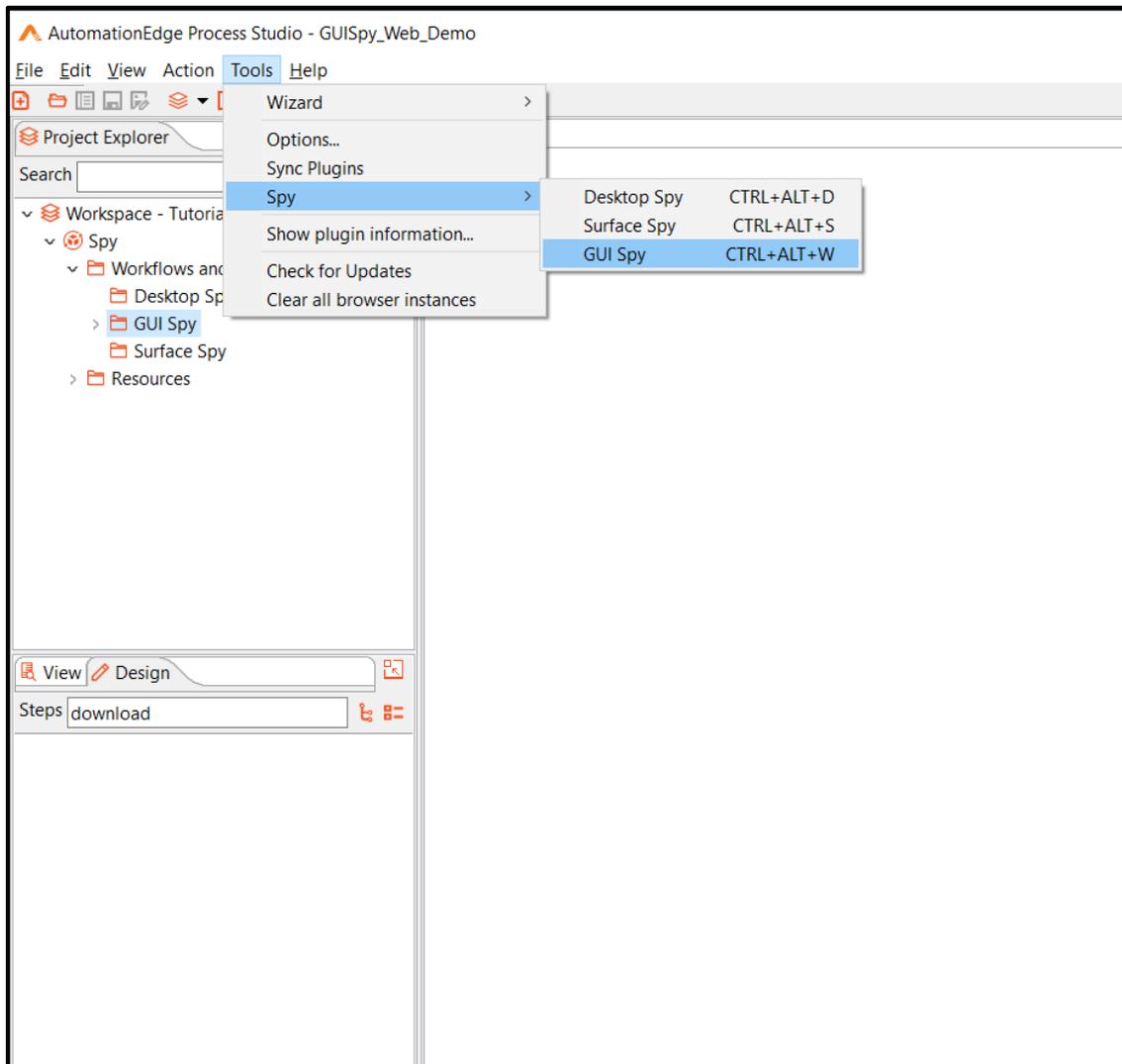
For a sample workflow with GUI Spy for web refer to section Project 1: Stock Value. The tutorial demonstrates GUI Robotic Process Automation to fetch Stock Values

12 Project 8: GUI Spy Recorder

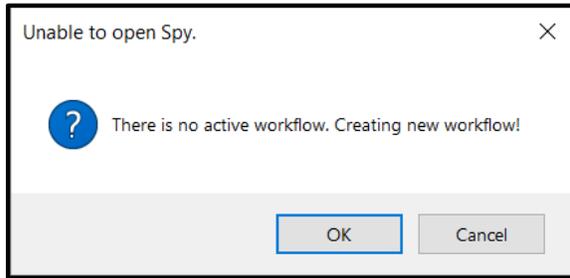
This tutorial demonstrates GUI recording. We will record a use case to open the Adobe website (<https://www.adobe.com/in/acrobat/online/pdf-to-word.html>), upload a pdf to convert to a word document.

GUI recording steps are as follows,

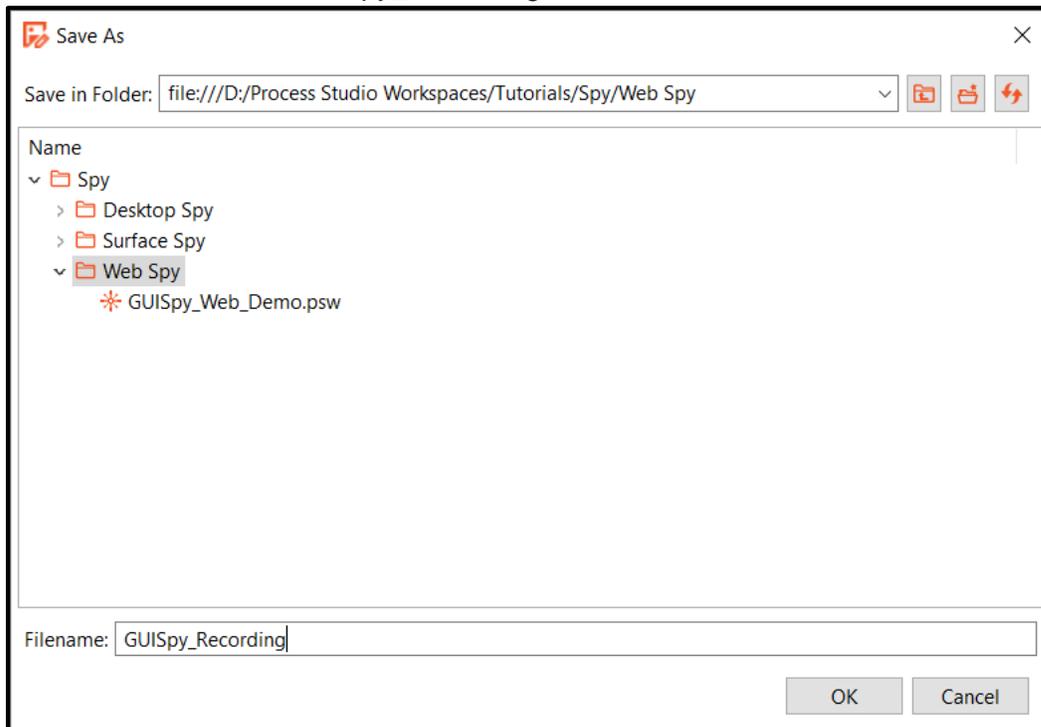
1. In Process Studio navigate to Tools→Spy→GUI Spy menu to start GUI Spy.



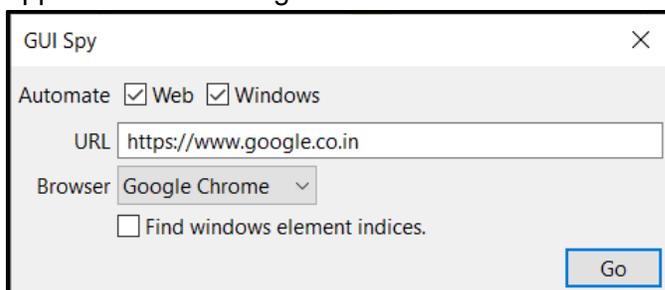
2. If no workflow is open a pop-up message appears. Click OK to create a new workflow.



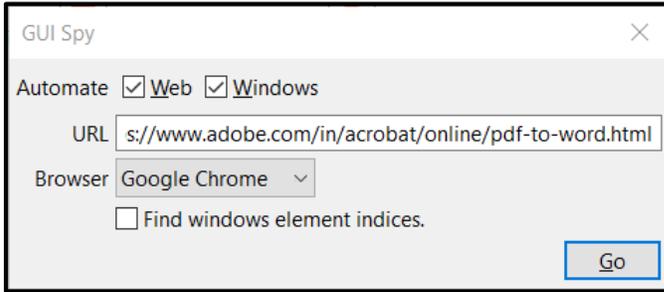
3. Save the workflow as GUISpy_Recording.



4. The GUI Spy dialog appears.
5. For only Web or Windows applications enable only the respective checkboxes. Enable the Find indices checkbox only for robust Windows applications automation.
6. For this use case leave Web and Windows checked to enable GUI Spy for web and windows applications recording.



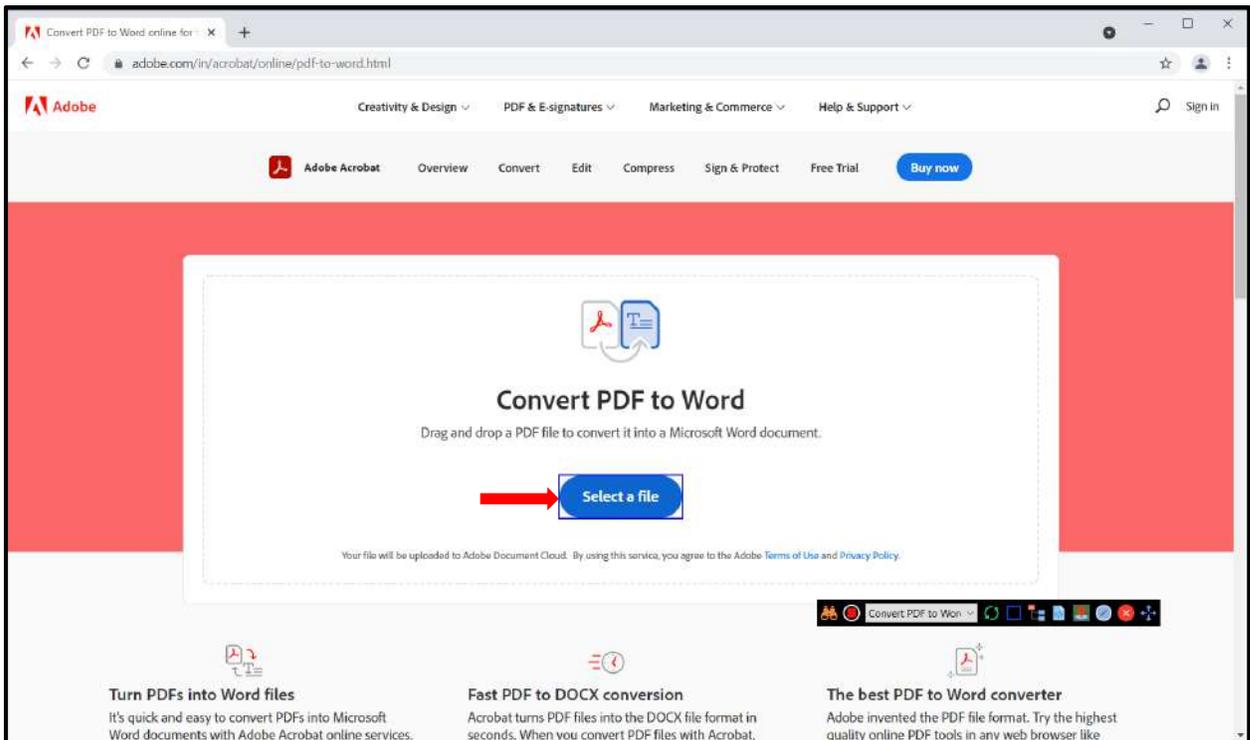
- Change the URL to the Adobe site - <https://www.adobe.com/in/acrobat/online/pdf-to-word.html> as seen below.



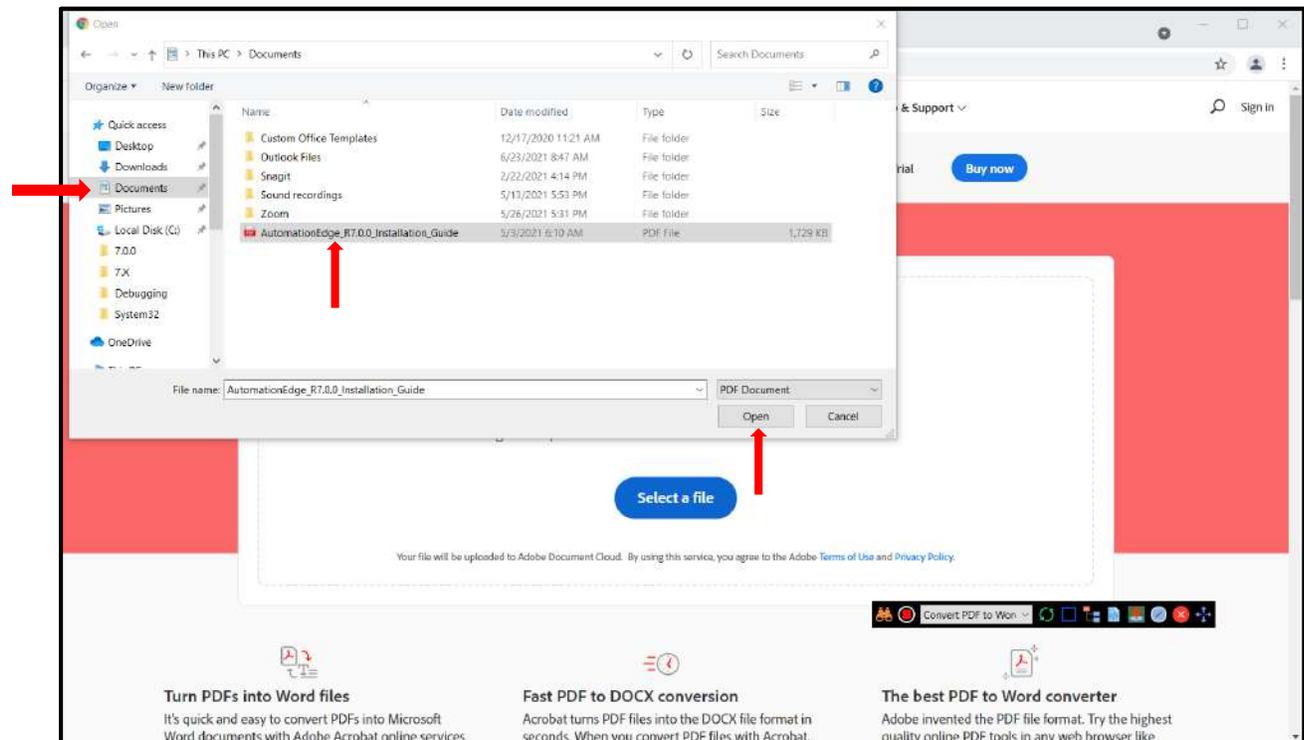
- The Adobe website opens and the the GUI Spy toolbar for Web and Windows opens as below.



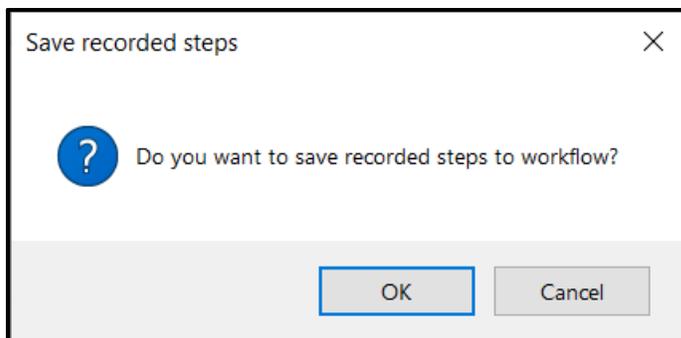
- Start recording by clicking the Start Recording icon.
- Hover over the 'Select a File', button to see the blue box. Click the button.



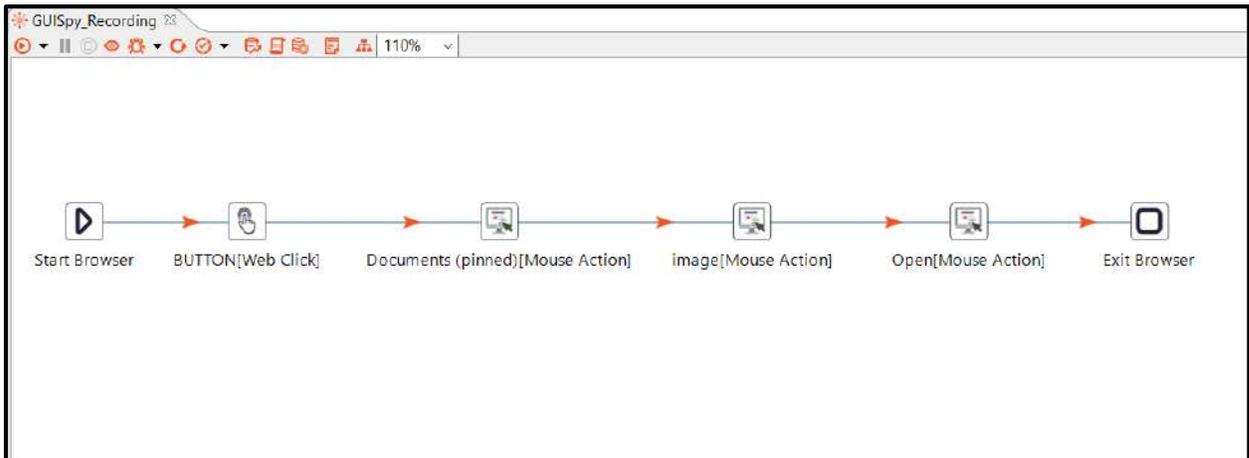
11. The Windows Explorer opens. Click the appropriate folder in the left navigation and click the pdf in the right navigation.
12. In this case we clicked the Documents folder and a pdf file as visible below.
13. Click the Open Button.



14. Click the Stop Recording icon (🛑) to stop the recording.
15. Confirm the pop-up to save the recorded steps.



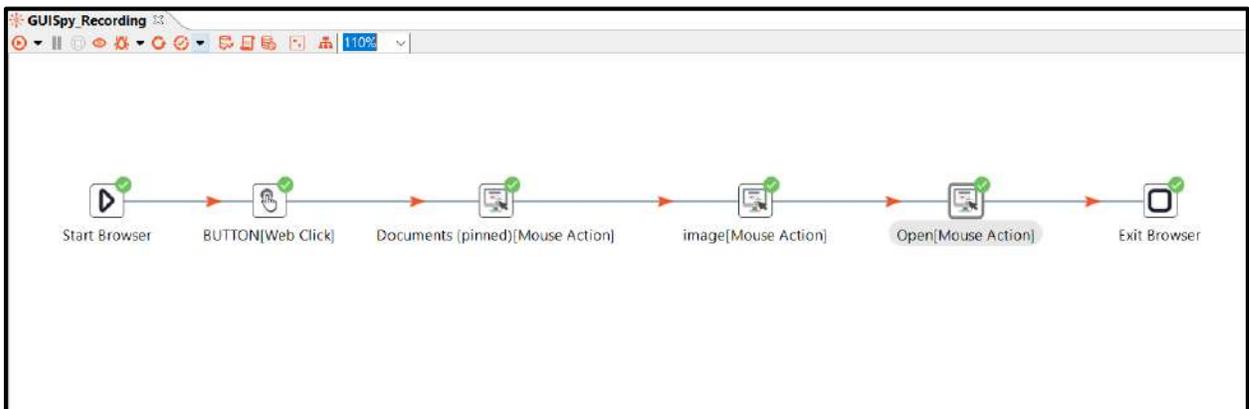
16. The recorded workflow in created in Process studio.
17. GUI for Web and Windows steps are added as seen below.



18. Execute the workflow with Run/Run Options. Accept the pop-up to Save files.



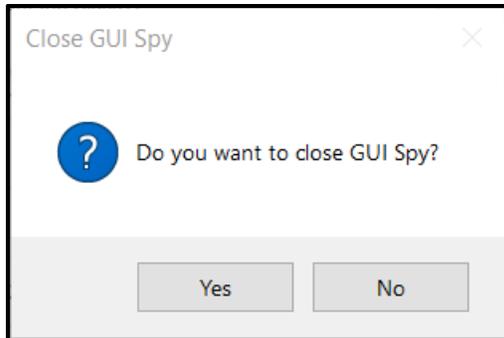
19. Save the workflow to verify it and execute the workflow.
20. Confirm that the image is Downloaded in the download folder.



21. The GUI recording feature is for ease and quick workflow creation. Some steps in the recorded workflow can be added or modified as expected if required. To launch other applications, stop recording by clicking the Stop Recording icon on the GUI toolbar. Add a Launch step to the workflow and once again Start Recording.

 **Note** In some cases, if GUI Recorder fails to highlight/record a few elements, use GUI Spy to generate steps for such element actions.

22. For now, this use case is over, Click the close GUI Spy icon. Confirm to close GUI spy.



23. The tutorial demonstrated Web and Windows recording with GUI Spy.

13 Project 8: GUI Spy for Windows

13.1 Sample workflow with GUI Spy for Windows

13.1.1 Update Opportunity in CRM Application with GUI Automation Plugin

In this section we will create a workflow to get an Opportunity, update the opportunity quantity and fetch the old and new Quantity and Total Price values using GUI Automation Plugin steps. Additionally, the intent of this workflow is to use and demonstrate several GUI Automation for Windows Plugin steps.

Compatibility:

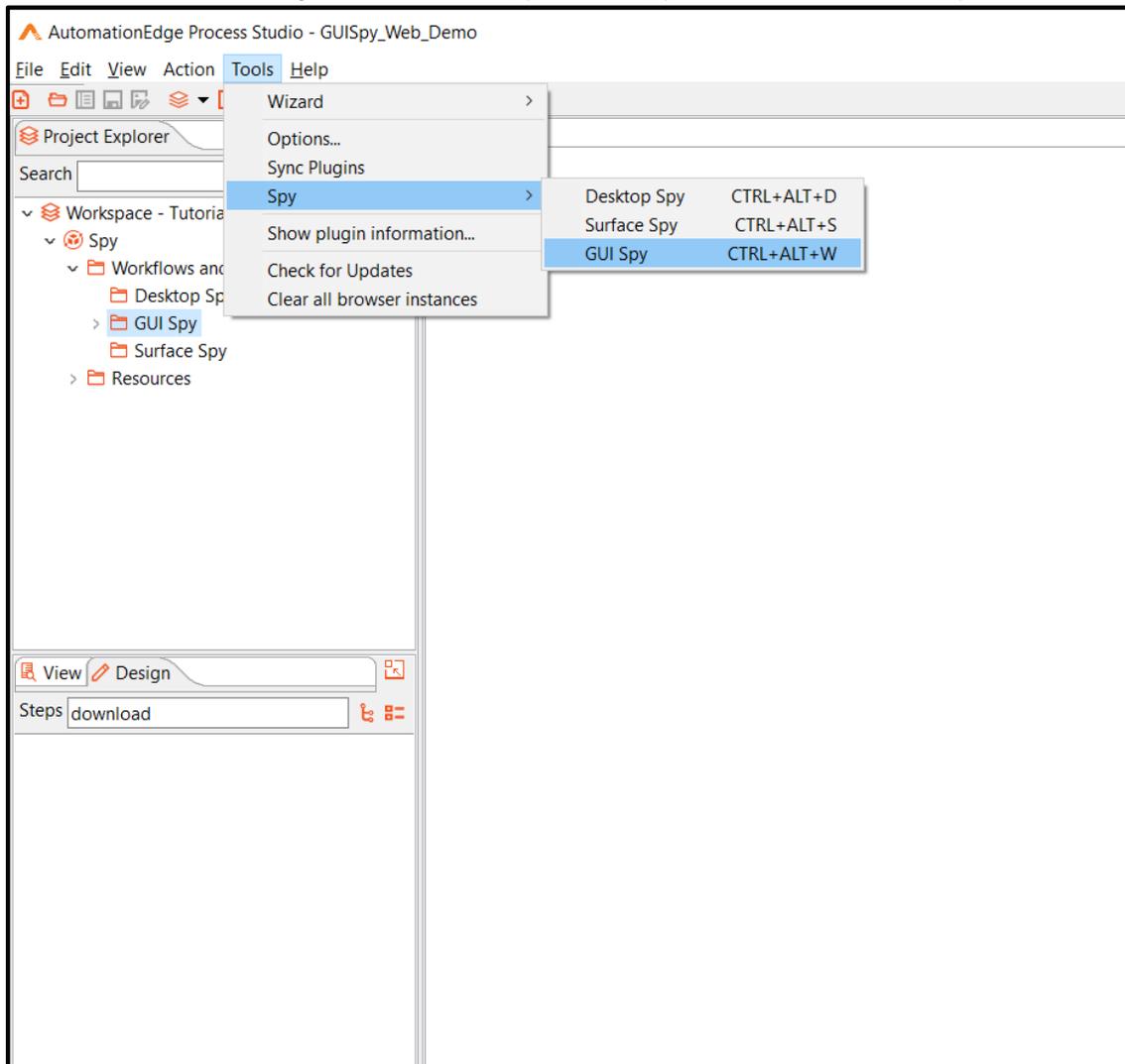
- You need to install Telerik WPF CRM application from the URL below, https://demos.telerik.com/wpf/crm/?_ga=2.126881805.1610052580.1594203121-2096267343.1594033569
- Typically following is the location of the Telerik CRM executable:
C:\Users\aeadmin\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Progress Software Corporation\Telerik UI for WPF\Telerik CRM for WPF.appref-ms
- You may Start or launch Telerik WPF CRM application from the shortcut on your desktop.
- This workflow has been created on Window 10. If you are working on any other Windows OS, please incorporate necessary variations.

Note:

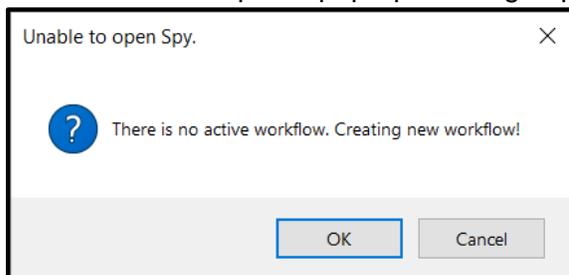
AutomationEdge Workflows with GUI Windows steps are marked Sequential when publishing to AutomationEdge UI.

Following are the steps to create 'Update Opportunity' workflow along with other functionality.

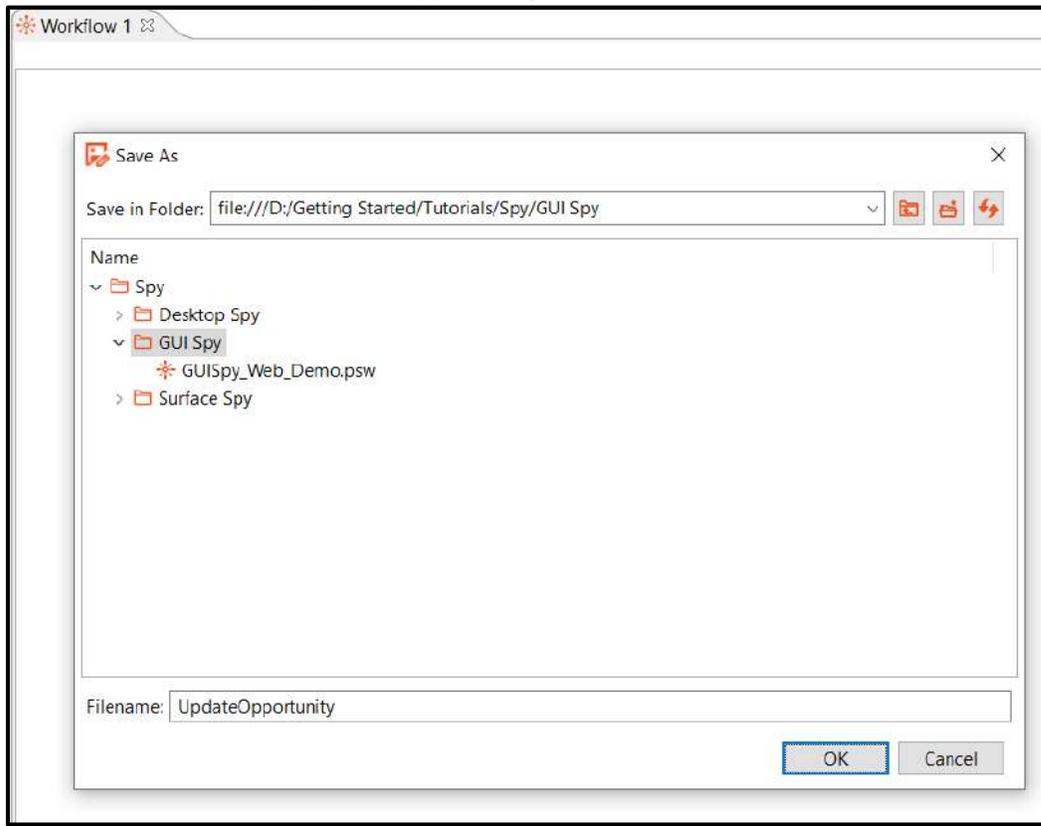
1. In Process Studio navigate to Tools→Spy→GUI Spy menu to start GUI Spy.



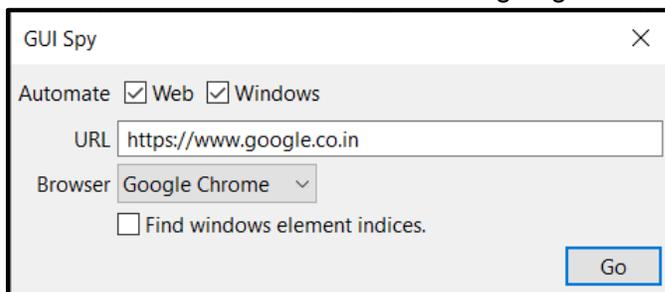
2. If no workflow is open a pop-up message appears. Click OK to create a new workflow.



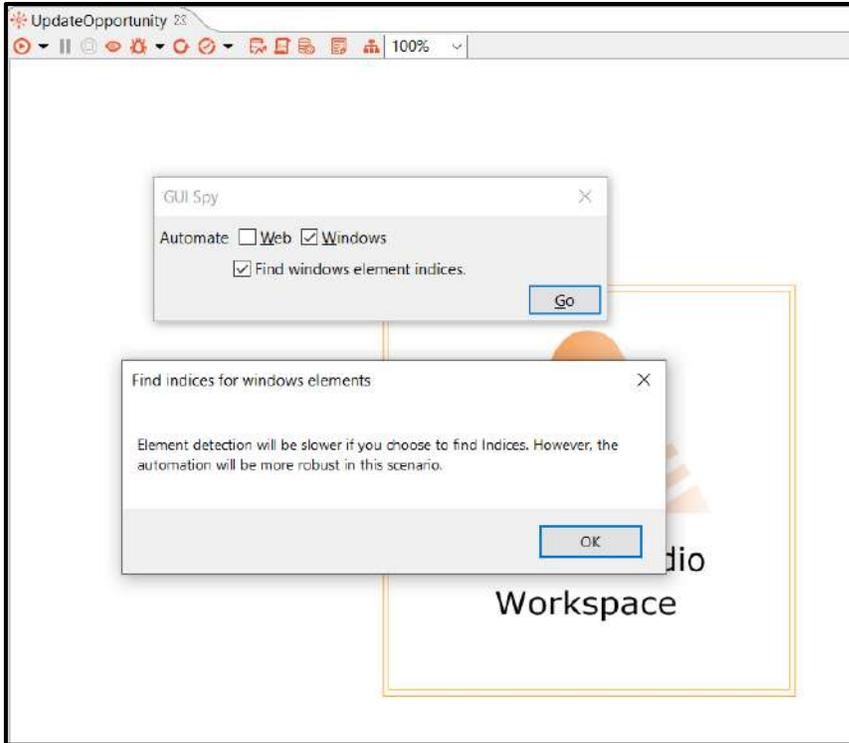
3. Save the workflow as Update Opportunity.



4. The GUI Spy toolbar for Windows popup appears.
5. In this cases uncheck Web as we are going to Gui Spy a purely Windows application.



6. The GUI Spy popup for Windows appears as below.
7. Enable the Find indices for windows elements, for robust automation for windows applications.

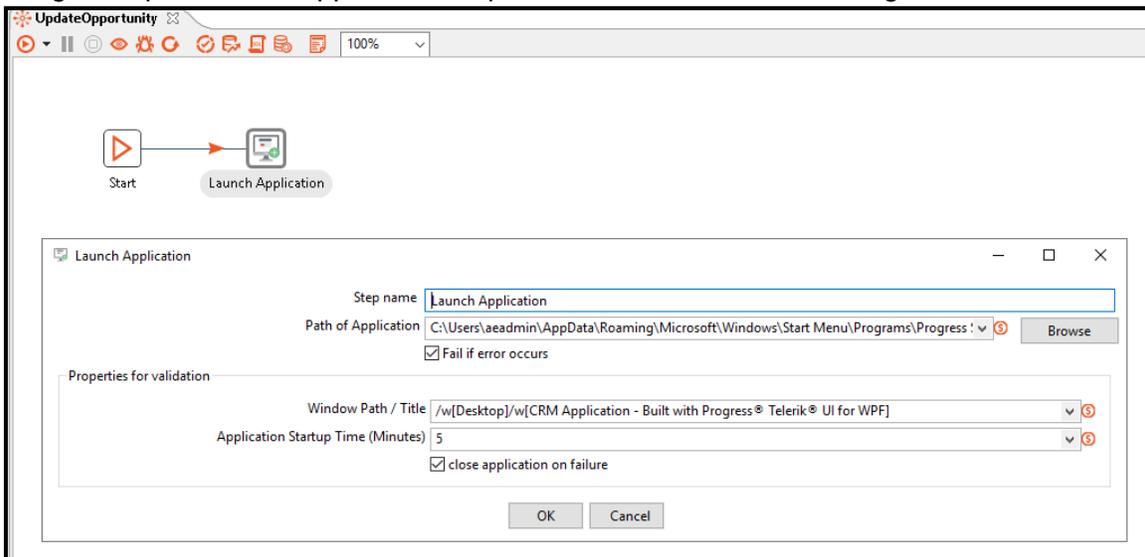


8. The GUI Spy toolbar for windows opens as below.



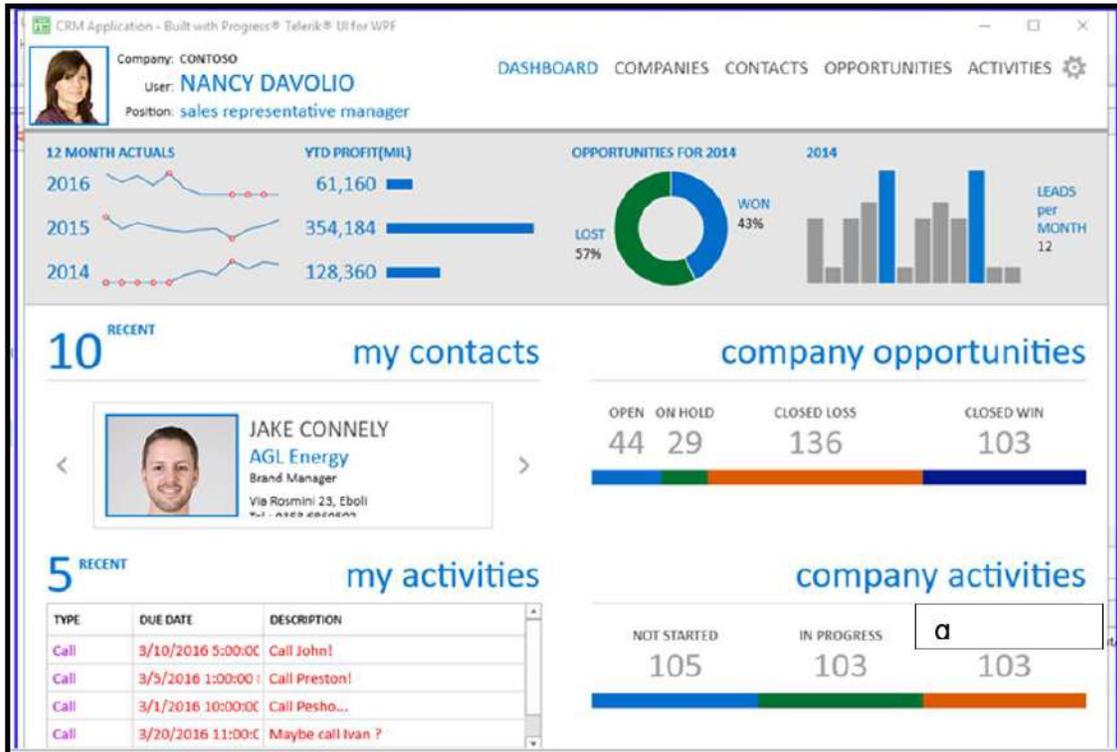
9. Drag and drop a Start step.

10. Drag-n-drop a Launch Application step. Double click to see the configuration.

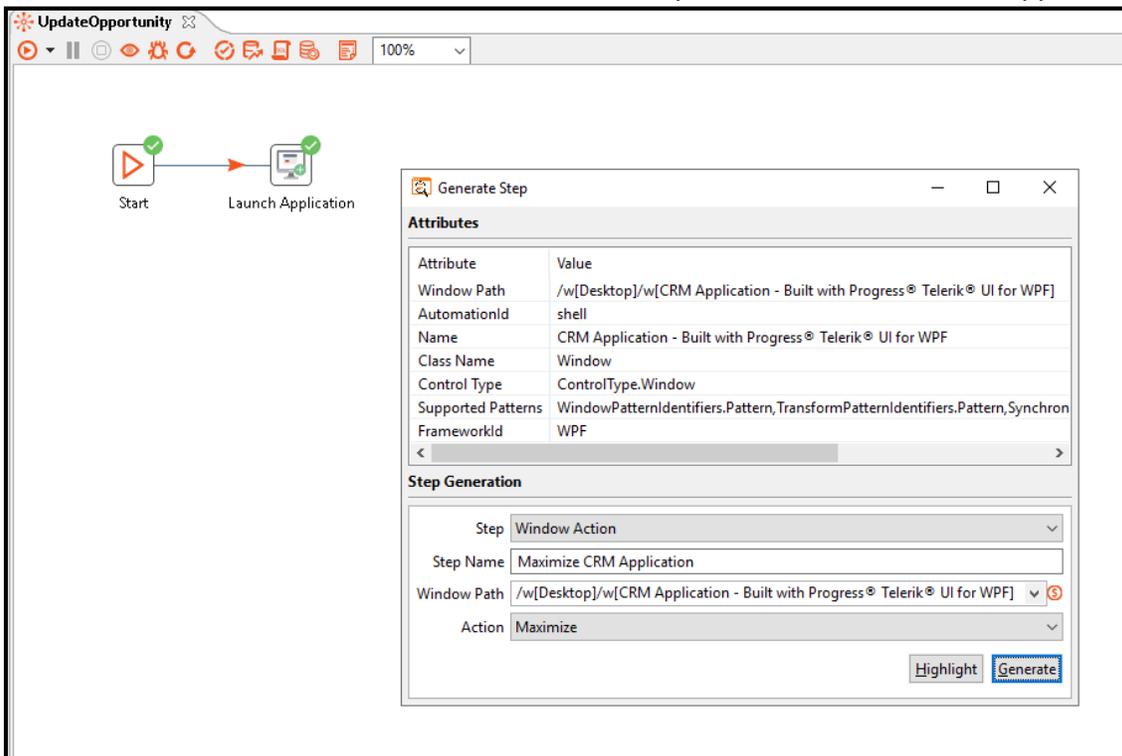


11. Launch the CRM application manually or using this workflow with the two steps. Once the CRM application is launched, maximize it and then perform other operations.

12. Hover over the edges of the application and press Shift. The window is highlighted as seen below



13. Generate Step window is displayed as seen below. Configure a Windows Action step to maximize the window. Click Generate to add the step named Maximize CRM Application.



14. Next we wish to click OPPORTUNITIES. Windows Spy on OPPORTUNITIES as seen below.

Company: CONTOSO
User: NANCY DAVOLIO
Position: sales representative manager
08 Jul 20 WEDNESDAY

DASHBOARD COMPANIES CONTACTS **OPPORTUNITIES** ACTIVITIES

12 MONTH ACTUALS
2016
2015
2014

YTD PROFIT(MIL)
61,160
354,184
128,360

OPPORTUNITIES FOR 2014
LOST 57% WON 43%

2014
LEADS per MONTH 12

10 RECENT my contacts
JAKE CONNELLY
AGL Energy
Brand Manager
Via Rosmini 23, Eboli
Tel.: 0353 6860502
jake.connelly@mailinator.com

company opportunities
OPEN 44 ON HOLD 29 CLOSED LOSS 136 CLOSED WIN 103

5 RECENT my activities
TYPE DUE DATE DESCRIPTION
Call 3/10/2016 5:00:00 PM Call John!
Call 3/5/2016 1:00:00 PM Call Preston!
Call 3/1/2016 10:00:00 AM Call Pesho...
Call 3/20/2016 11:00:00 AM Maybe call Ivan ?

company activities
NOT STARTED 105 IN PROGRESS 103 DONE 103

15. Configure Generate Step for clicking on OPPORTUNITIES. Click Generate.

UpdateOpportunity

Start → Launch Application → Maximize CRM Application

Generate Step

Attributes

Attribute	Value
Window Path	/w[Desktop]/w[CRM Application - Built with Progress® Telerik® UI for WPF]
Absolute XPath	/ProgressBar[0]/Custom[2]/RadioButton[3]
XPath	//ProgressBar[@AutomationId="busyIndicator"]/Custom[2]/RadioButton[3]
Class Name	RadRadioButton
Control Type	ControlType.RadioButton
HelpText	RadRadioButton
Supported Patterns	SelectedItemPatternIdentifiers.Pattern, TogglePatternIdentifiers.Pattern, SynchronizedInputPat
FrameworkId	WPF

Step Generation

Step: Mouse Action

Step Name: Click Opportunity

Window Path: /w[Desktop]/w[CRM Application - Built with Progress® Telerik® UI for WPF]

Action: Click

Criteria: Absolute XPath

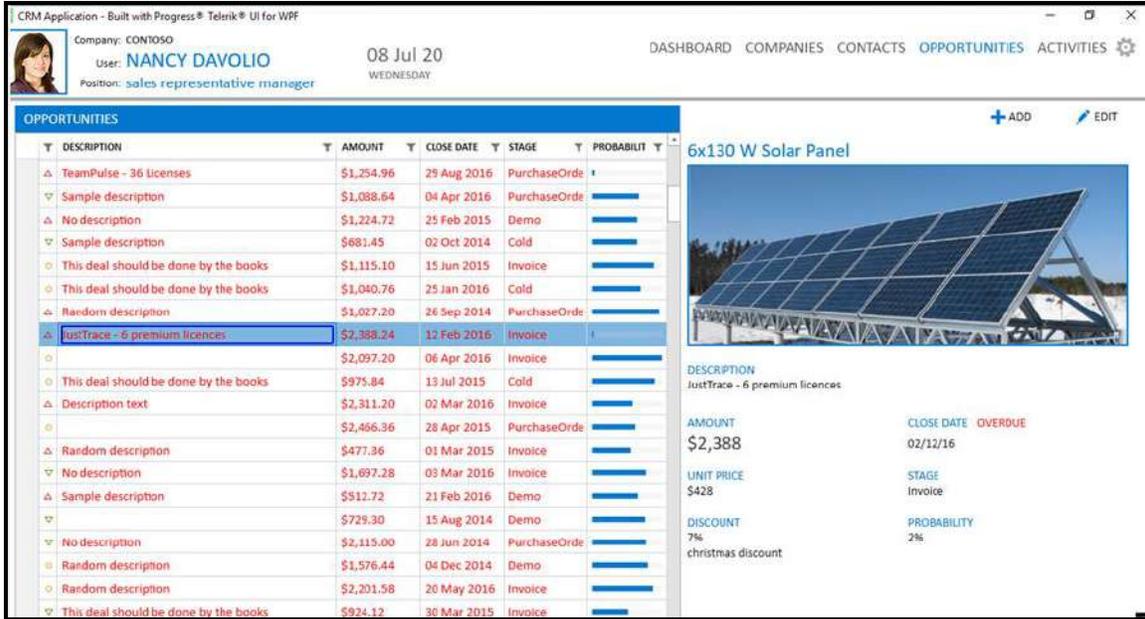
Criteria Value: /ProgressBar[0]/Custom[2]/RadioButton[3]

Element Index (Starts with 0): 0

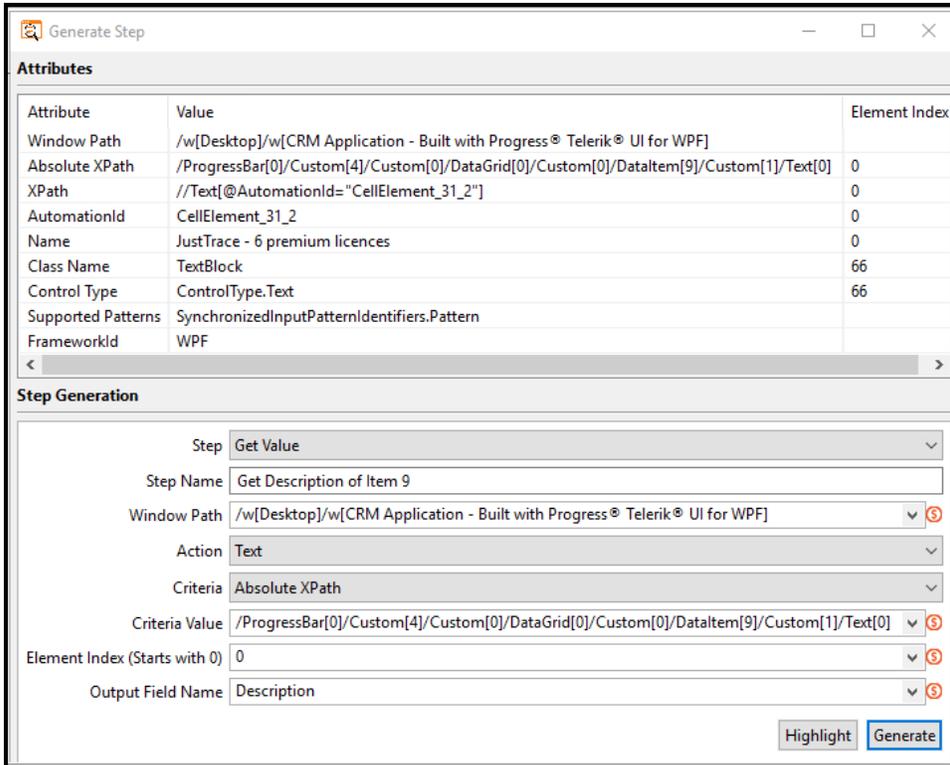
Highlight Generate

16. Step [Click Opportunity] generated message ppears as seen below.

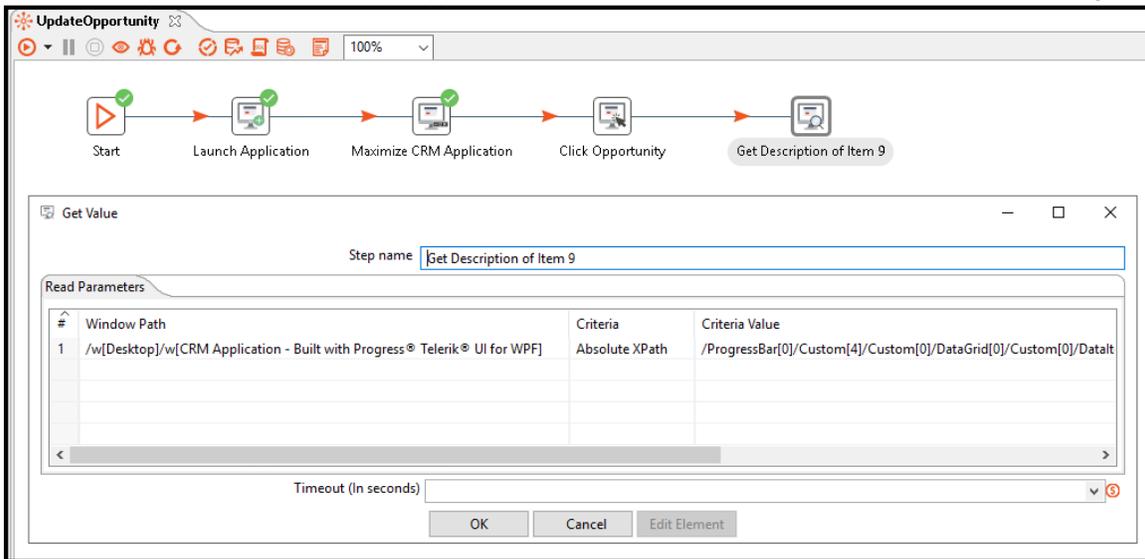
17. Next Spy to capture Opporunity description. ‘JustTrace – 6 premium licences’ is seen with a blue border below.



18. The Generation window appears as seen below. Click Generate.

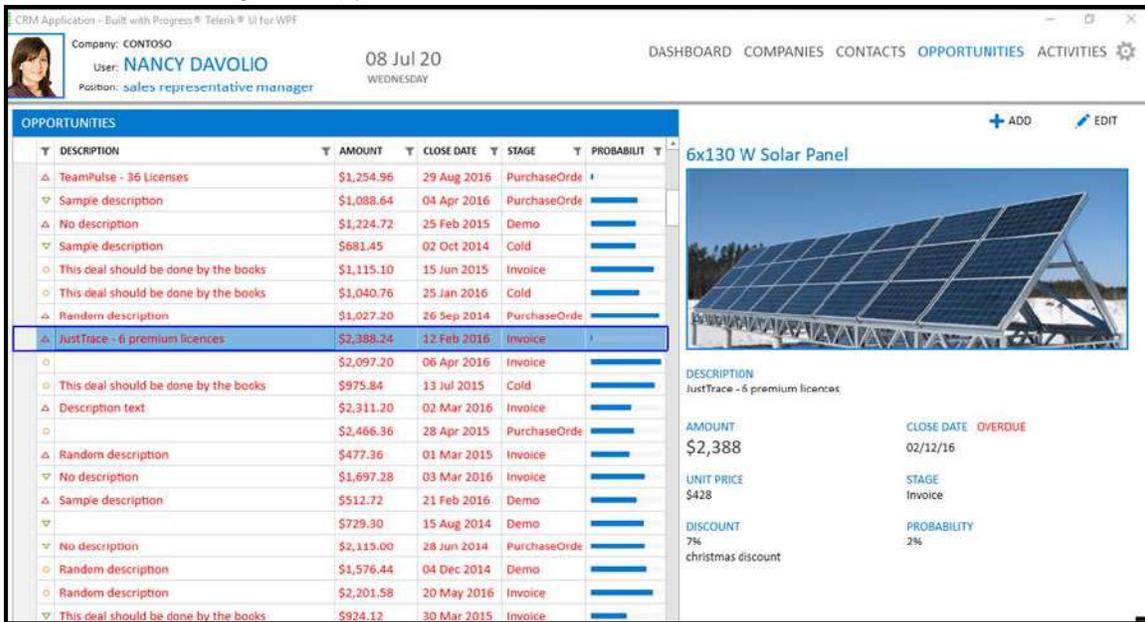


19. 'Get Description of Item 9' step appears as seen below. Double click to see the configuration.



20. Next we wish to click on the item record.

21. Hover over the edges to spy the entire record.



22. Configure Generate step window for click on the item record as seen below. Click Generate.

Attribute	Value
Window Path	/w[Desktop]/w[CRM Application - Built with Progress® Telerik® UI for WPF]
Absolute XPath	/ProgressBar[0]/Custom[4]/Custom[0]/DataGrid[0]/Custom[0]/DataItem[9]/DataItem[0]
XPath	//DataItem[@AutomationId="Row_31"]/DataItem[0]
AutomationId	CRM.Modules.Repository.Services.Opportunity
Name	CRM.Modules.Repository.Services.Opportunity
Class Name	CRM.Modules.Repository.Services.Opportunity
Control Type	ControlType.DataItem
HelpText	CRM.Modules.Repository.Services.Opportunity
Supported Patterns	InvokePatternIdentifiers.Pattern, SelectionItemPatternIdentifiers.Pattern, ScrollItemPatternIdentifiers.Patter
FrameworkId	WPF

Step Generation

Step: Mouse Action

Step Name: Click on Item 9

Window Path: /w[Desktop]/w[CRM Application - Built with Progress® Telerik® UI for WPF]

Action: Click

Criteria: Absolute XPath

Criteria Value: /ProgressBar[0]/Custom[4]/Custom[0]/DataGrid[0]/Custom[0]/DataItem[9]/DataItem[0]

Element Index (Starts with 0): 0

Buttons: Highlight, Generate

23. Click on Item 9 is generated as seen below. Double click to see the configuration.

UpdateOpportunity

Start → Launch Application → Maximize CRM Application → Click Opportunity → Get Description of Item 9 → Click on Item 9

Click on Item 9 Configuration

Step name: Click on Item 9

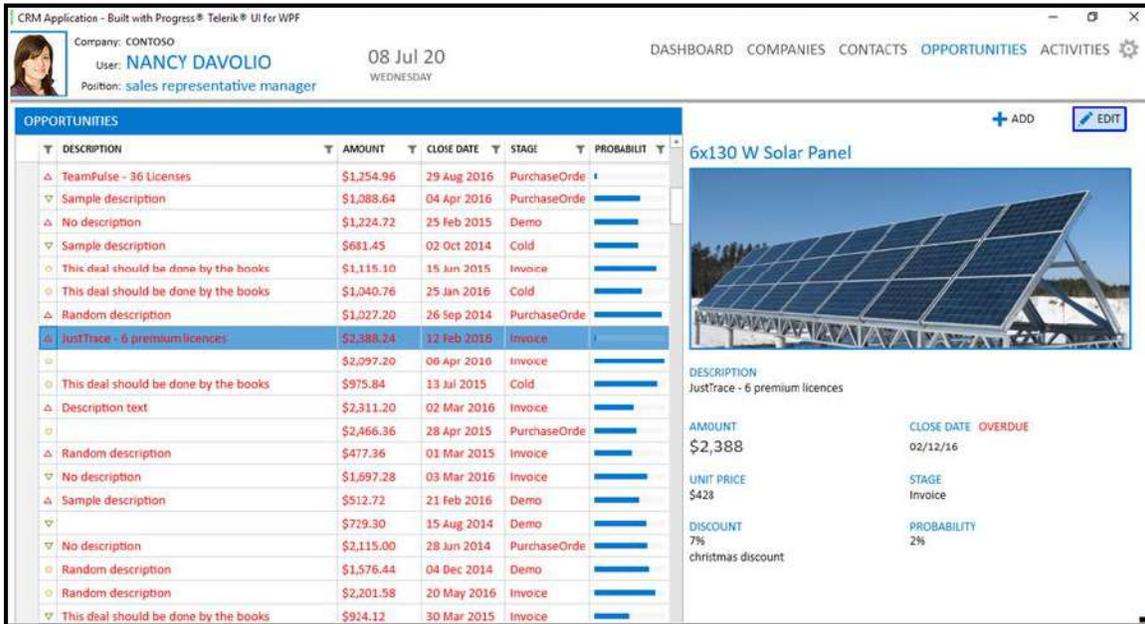
Action Type: Click

Source Element:

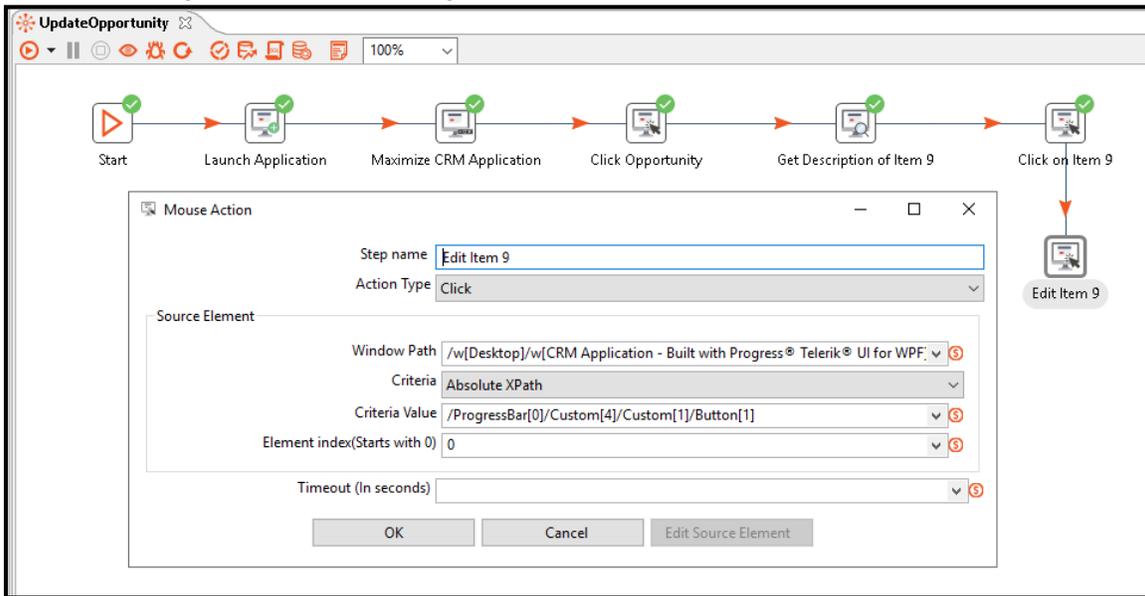
- Window Path: /w[Desktop]/w[CRM Application - Built with Progress® Telerik® UI for WPF]
- Criteria: Absolute XPath
- Criteria Value: /ProgressBar[0]/Custom[4]/Custom[0]/DataGrid[0]/Custom[0]/DataItem[9]/DataItem[0]
- Element index(Starts with 0): 0
- Timeout (In seconds):

Buttons: OK, Cancel, Edit Source Element

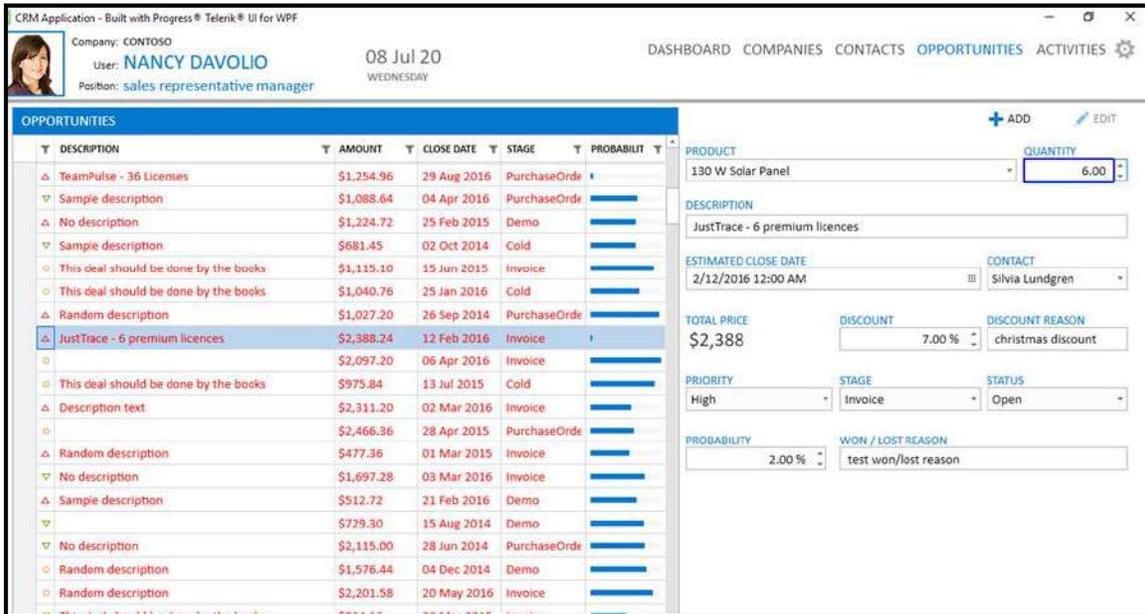
24. Next Spy on EDIT.



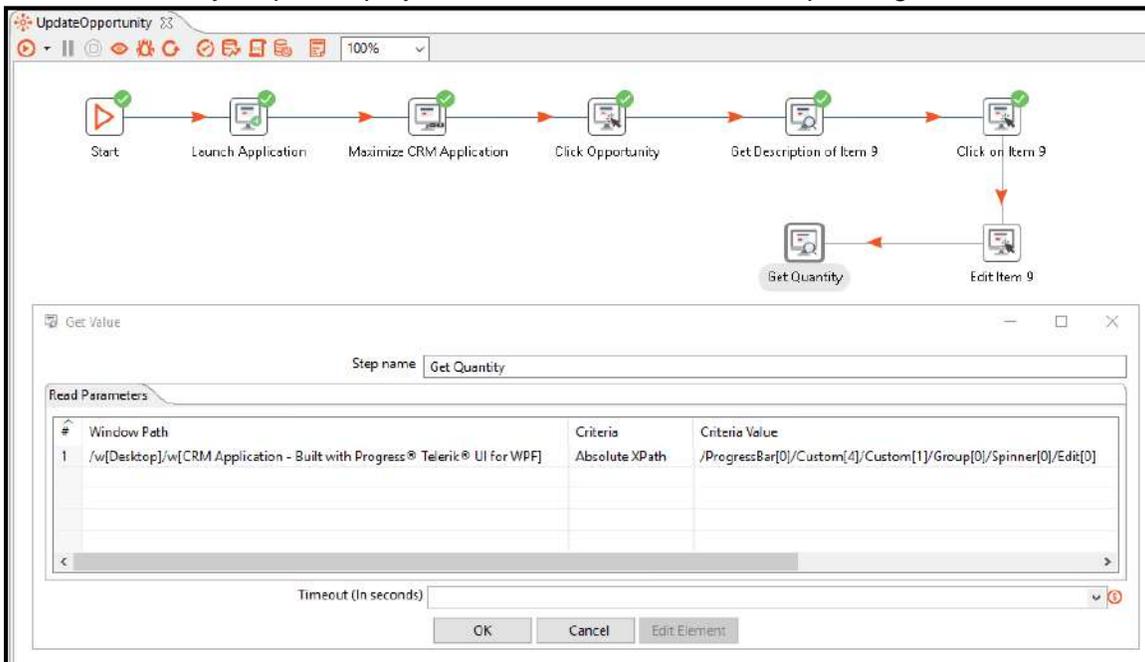
25. Edit Item 9 is generated with configurations as seen below.



26. Now we will capture the quantity on this opportunity. Spy Quantity and Generate step.



27. The Get Quantity step is displayed. Double click to see the step configuration as below.



28. Now add a 'Modified Java Script value' step as seen below. Write a script as seen below to increment the quantity captured in the step above by 2.

The screenshot shows a workflow in Process Studio with the following steps: Start, Launch Application, Maximize CRM Application, Click Opportunity, Get Description of Item 9, Click on Item 9, Calculate Quantity, Get Quantity, and Edit Item 9. A 'Script Values / Mod' window is open for the 'Calculate Quantity' step, showing the following JavaScript code:

```

Step name: Calculate Quantity

Java script functions:
- Transform Scripts
- Transform Constants
- Transform Functions
- Input fields
  - Description
  - Quantity
- Output fields
  - Please use the '

Java script:
Script 1
//Script here
var NewQuantity;
NewQuantity = str2num(Quantity) + 2;

Linens: 0
Compatibility mode:  Optimization level: 9
    
```

#	Fieldname	Rename to	Type	Length	Precision	Replace value 'Fieldname' or 'Rename to'
1	NewQuantity		Number			N

29. Again Spy Quantity. This time we will set the new quantity calculated in the Java Script step. Generate a step for Set value.

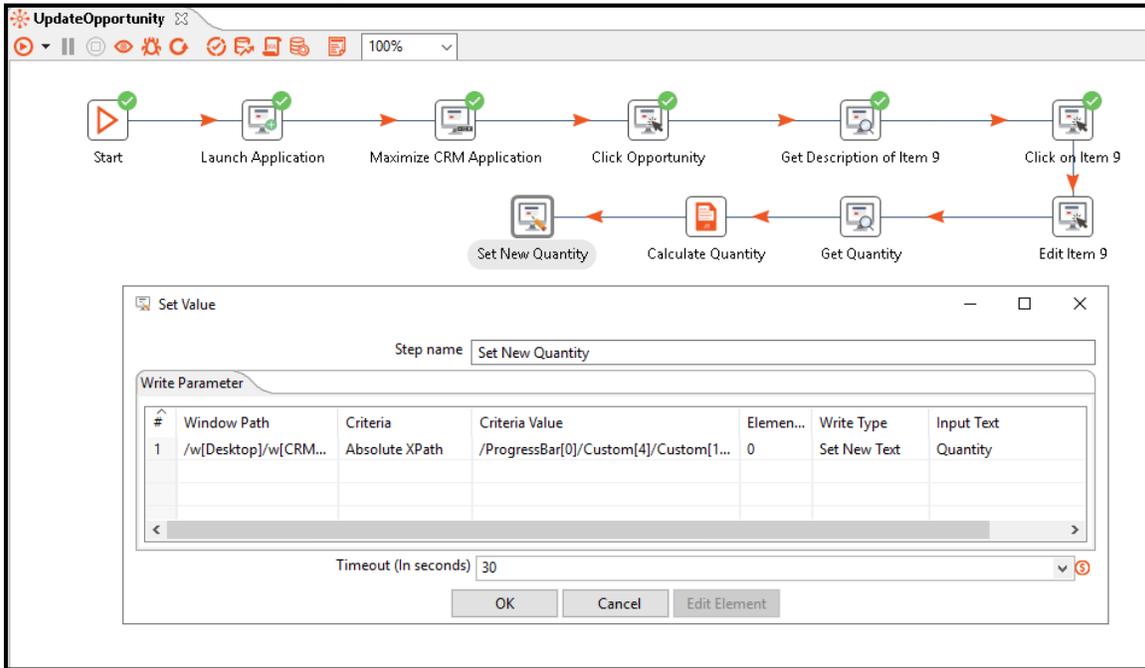
The screenshot shows a CRM application interface. The 'OPPORTUNITIES' table is visible with the following data:

DESCRIPTION	AMOUNT	CLOSE DATE	STAGE	PROBABILITY
TeamPause - 36 licenses	\$1,254.96	29 Aug 2016	PurchaseOrder	
Sample description	\$1,088.64	04 Apr 2016	PurchaseOrder	
No description	\$1,224.72	25 Feb 2015	Demo	
Sample description	\$681.45	02 Oct 2014	Cold	
This deal should be done by the books	\$1,115.10	15 Jan 2015	Invoice	
This deal should be done by the books	\$1,040.76	25 Jan 2016	Cold	
Random description	\$1,027.20	26 Sep 2014	PurchaseOrder	
JustTrace - 6 premium licences	\$2,388.24	12 Feb 2016	Invoice	
	\$2,097.20	06 Apr 2016	Invoice	
This deal should be done by the books	\$975.84	13 Jul 2015	Cold	
Description text	\$2,311.20	02 Mar 2016	Invoice	
	\$2,466.36	28 Apr 2015	PurchaseOrder	
Random description	\$477.36	01 Mar 2015	Invoice	
No description	\$1,697.28	03 Mar 2016	Invoice	
Sample description	\$512.72	21 Feb 2016	Demo	
	\$729.30	15 Aug 2014	Demo	
No description	\$2,115.00	28 Jan 2014	PurchaseOrder	
Random description	\$1,576.44	04 Dec 2014	Demo	
Random description	\$2,201.58	20 May 2016	Invoice	
This deal should be done by the books	\$924.12	30 Mar 2015	Invoice	

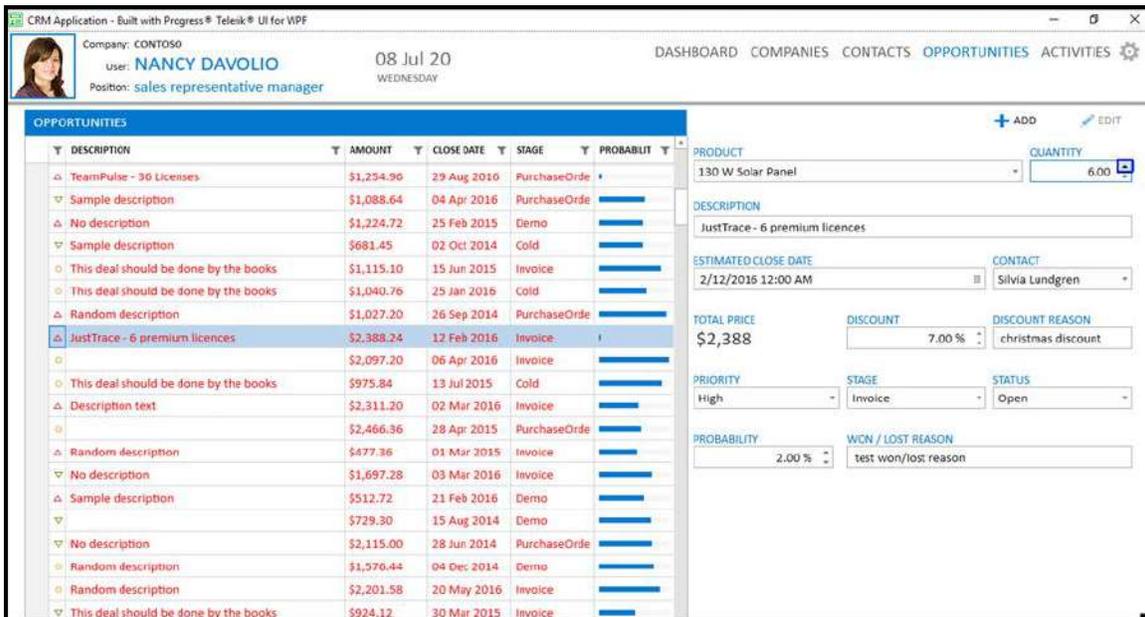
The detailed view on the right shows the following fields:

- PRODUCT: 130 W Solar Panel
- QUANTITY: 6.00
- DESCRIPTION: JustTrace - 6 premium licences
- ESTIMATED CLOSE DATE: 2/12/2016 12:00 AM
- CONTACT: Silvia Lundgren
- TOTAL PRICE: \$2,388
- DISCOUNT: 7.00%
- DISCOUNT REASON: christmas discount
- PRIORITY: High
- STAGE: Invoice
- STATUS: Open
- PROBABILITY: 2.00%
- WON / LOST REASON: test won/lost reason

30. Set New Quantity step is generated with configurations as seen below.



31. Next we shall click the increment arrow on Quantity so that the new Total Price is reflected on the screen.



32. Generate the Confirm Increase step with configurations as seen below.

The screenshot shows a workflow in Process Studio titled 'UpdateOpportunity'. The workflow consists of the following steps: Start, Launch Application, Maximize CRM Application, Click Opportunity, Get Description of Item 9, Click on Item 9, Edit Item 9, Get Quantity, Calculate Quantity, Set New Quantity, and Confirm Increase. A 'Mouse Action' dialog is open, showing the configuration for the 'Confirm Increase' step. The configuration includes:

- Step name: Confirm Increase
- Action Type: Click
- Source Element:
 - Window Path: /w[Desktop]/w[CRM Application - Built with Progress® Telerik® UI for WPF]
 - Criteria: Absolute XPath
 - Criteria Value: /ProgressBar[0]/Custom[4]/Custom[1]/Group[0]/Spinner[0]/Button[0]
 - Element index(Starts with 0): 0
 - Timeout (In seconds):

33. We can now capture the new Total Price.

34. Spy and Generate the step.

The screenshot shows a CRM application interface. On the left, there is a table of 'OPPORTUNITIES' with columns: DESCRIPTION, AMOUNT, CLOSE DATE, STAGE, and PROBABILIT. The table contains several rows of data. On the right, there is a detailed view of an opportunity. The 'TOTAL PRICE' field is highlighted with a red box, showing a value of \$3,184. Other fields in the detailed view include PRODUCT (130 W Solar Panel), QUANTITY (8.00), DESCRIPTION (JustTrace - 6 premium licences), ESTIMATED CLOSE DATE (2/12/2016 12:00 AM), CONTACT (Sivia Lundgren), DISCOUNT (7.00%), DISCOUNT REASON (christmas discount), PRIORITY (High), STAGE (Invoice), STATUS (Open), and PROBABILITY (2.00%).

35. The Get New Value for Total Price is generated with configurations as seen below.

The screenshot shows a workflow in Process Studio titled 'UpdateOpportunity'. The workflow steps are: Start, Launch Application, Maximize CRM Application, Click Opportunity, Get Description of Item 9, Click on Item 9, Edit Item 9, Get Quantity, Calculate Quantity, Set New Quantity, Confirm Increase, and Get New Value. Below the workflow is a configuration dialog for the 'Get Value' step.

Get Value Configuration Dialog

Step name: Get New Value

#	Window Path	Criteria	Criteria Value	Element index(...)	Read Type	Output field name
1	/w[Desktop]w[CRM Application - ...	Absolute XPath	/ProgressBar[0]/Custom[4]/Custom[1]/Gr...	0	Text	Get New Value

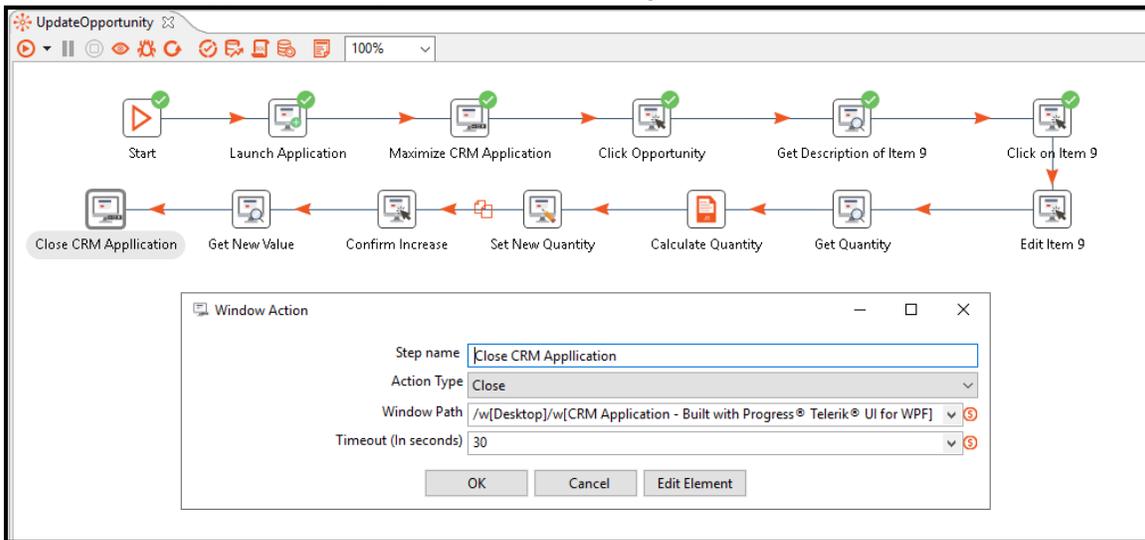
Timeout (In seconds): []

Buttons: OK, Cancel, Edit Element

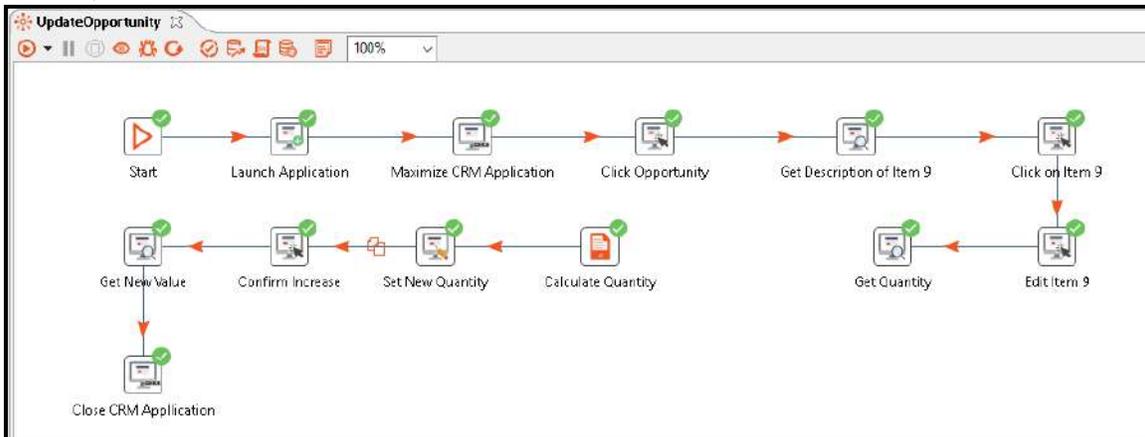
36. Finally we shall close the CRM application window. Spy the window by hovering over the edges. Generate the step.

The screenshot shows a CRM application window titled 'CRM Application - Built with Progress® Telerik® UI for WPF'. The user is Nancy Davolio, a sales representative manager, on 08 Jul 20. The main view shows a table of 'OPPORTUNITIES' with columns for Description, Amount, Close Date, Stage, and Probability. The selected opportunity is 'JustTrace - 6 premium licences' with an amount of \$3,184.32 and a close date of 12 Feb 2016. The detailed view on the right shows fields for Product (130 W Solar Panel), Quantity (8.00), Description (JustTrace - 6 premium licences), Estimated Close Date (2/12/2016 12:00 AM), Contact (Silvia Lundgren), Total Price (\$3,184), Discount (7.00%), Discount Reason (christmas discount), Priority (High), Stage (Invoice), Status (Open), and Probability (2.00%).

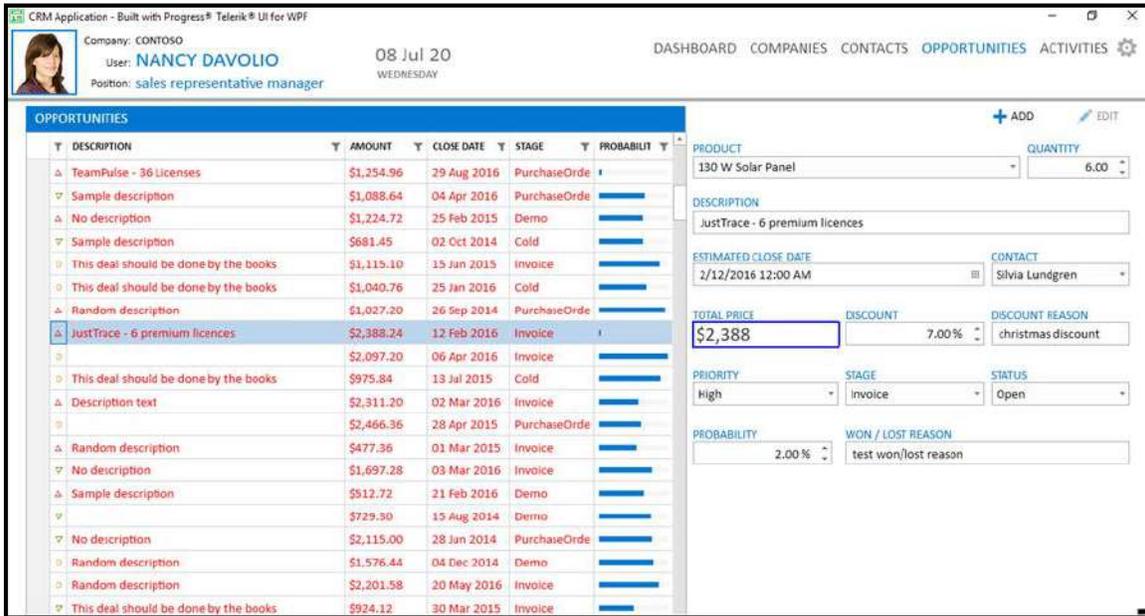
37. A Close CRM Application step is added with configurations as seen below.



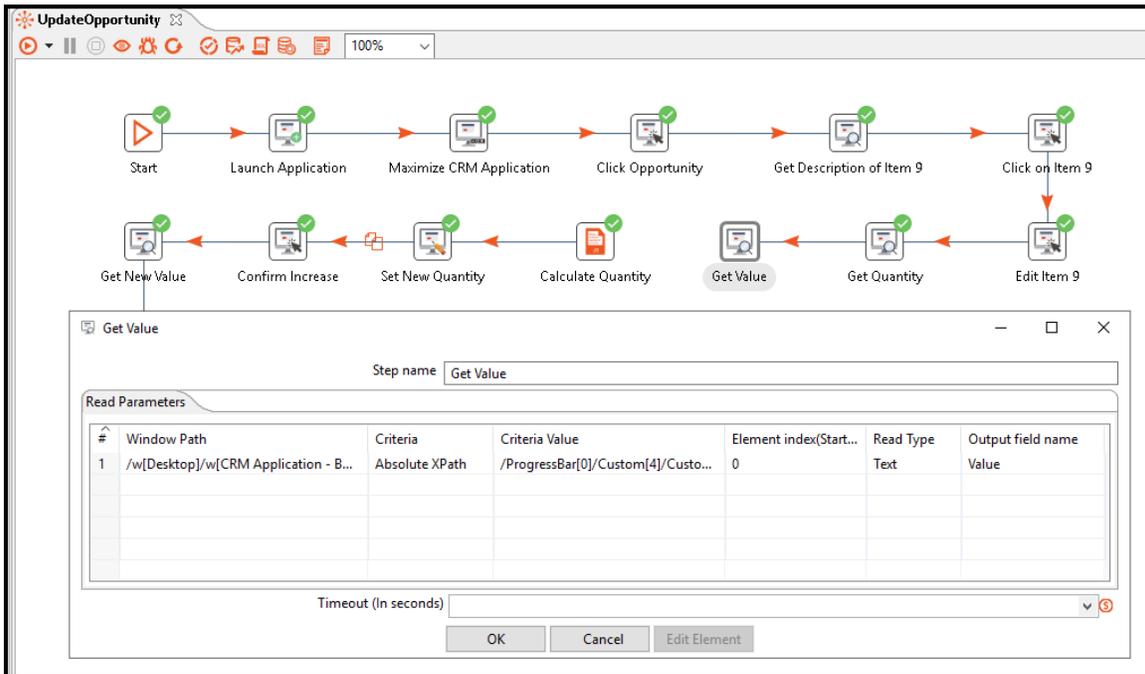
38. We will add a step in between Get Quantity and Calculate Quantity to Get Value before changing quantity. Delete the hop between these two steps.



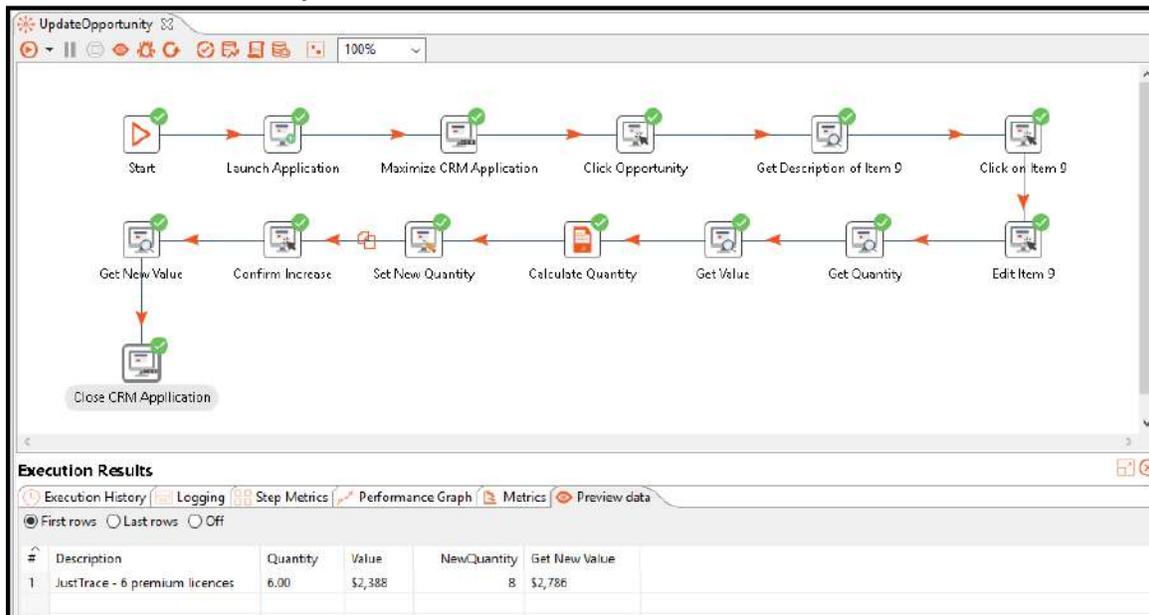
39. Spy Total Price and Generate the step.



40. A Get Value step has been inserted with configurations as seen below. Draw the hops on both sides.



- 41. The screenshot below shows the complete workflow.
- 42. It has been successfully executed and Execution Results can be seen.



- 43. A sample workflow creation tutorial with GUI Automation plugin steps for Windows, is complete.

14 Project 8: Desktop Spy

14.1 Sample Workflow with Desktop Spy

14.1.1 Remote Desktop Connection

In this section we will create a workflow called 'RemoteDesktopConnection'. In this workflow we shall demonstrate automation comprising Remote Desktop Connection Windows.

In the workflow we will use Desktop plugins.

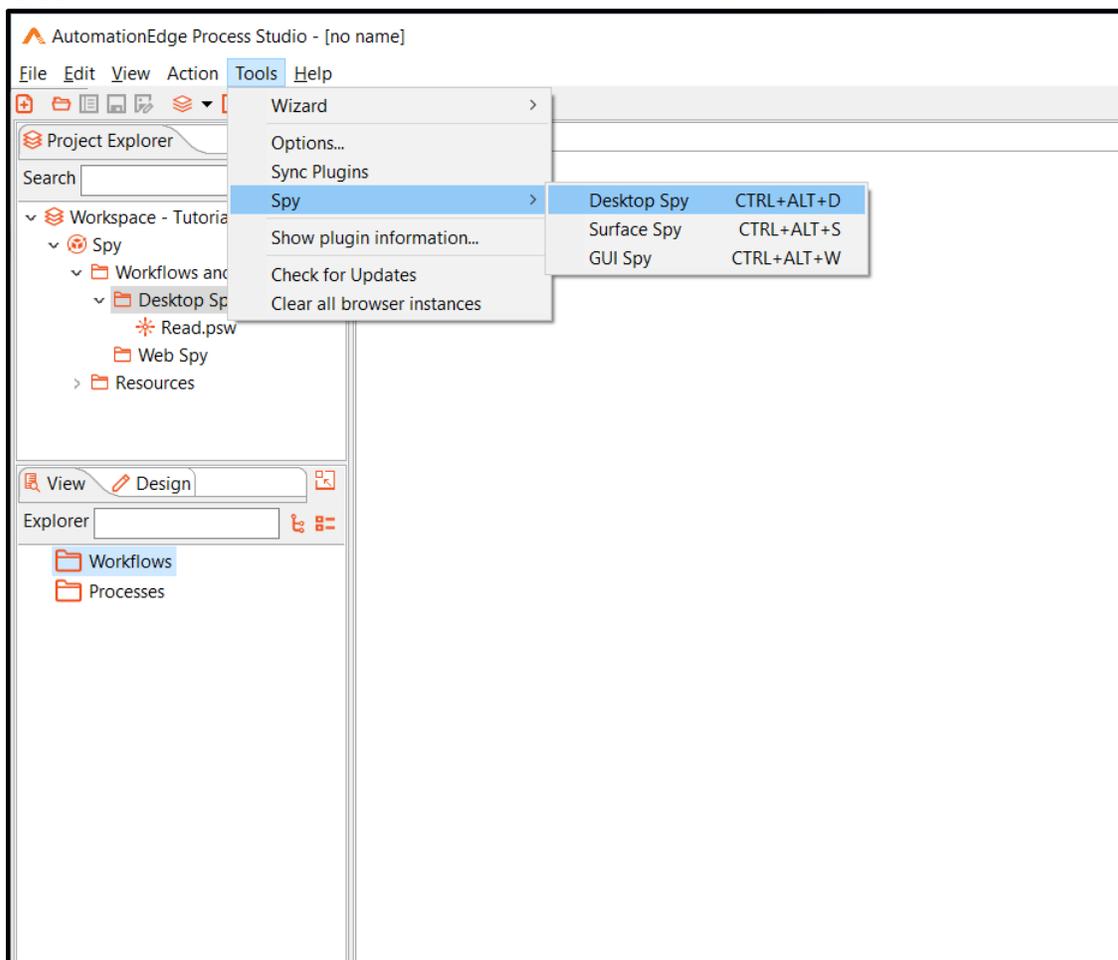
Desktop Spy can detect Windows application elements. Mouse over any element of the Windows Desktop Application and click keyboard left ctrl to populate element attributes.

If a workflow contains both GUI Automation Plugin steps for web based applications; and Desktop Plugin steps for Java applets and/or Windows Applications take care of the following requirements while designing a workflow,

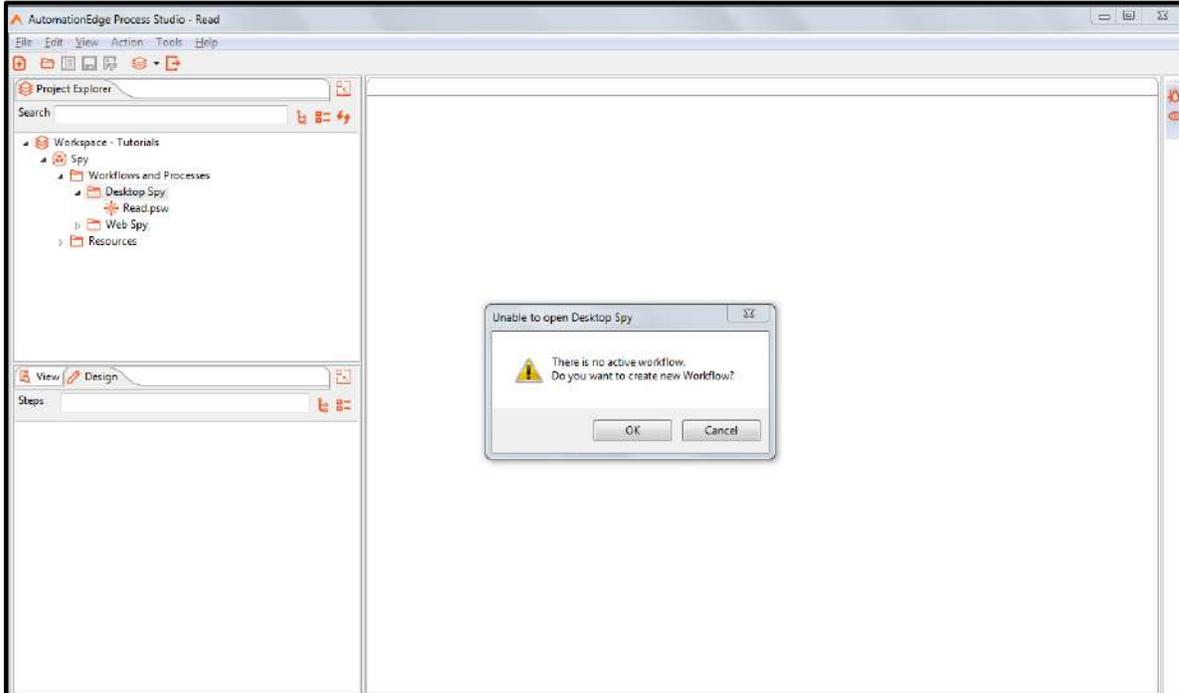
1. Launch Desktop Spy.
2. In Desktop Plugin steps for Windows application, select WINDOWS mode in the steps. In WINDOWS mode Desktop Start step can be placed before any Desktop plugin step.

Following are the steps to create workflow 'RemoteDesktopConnection'.

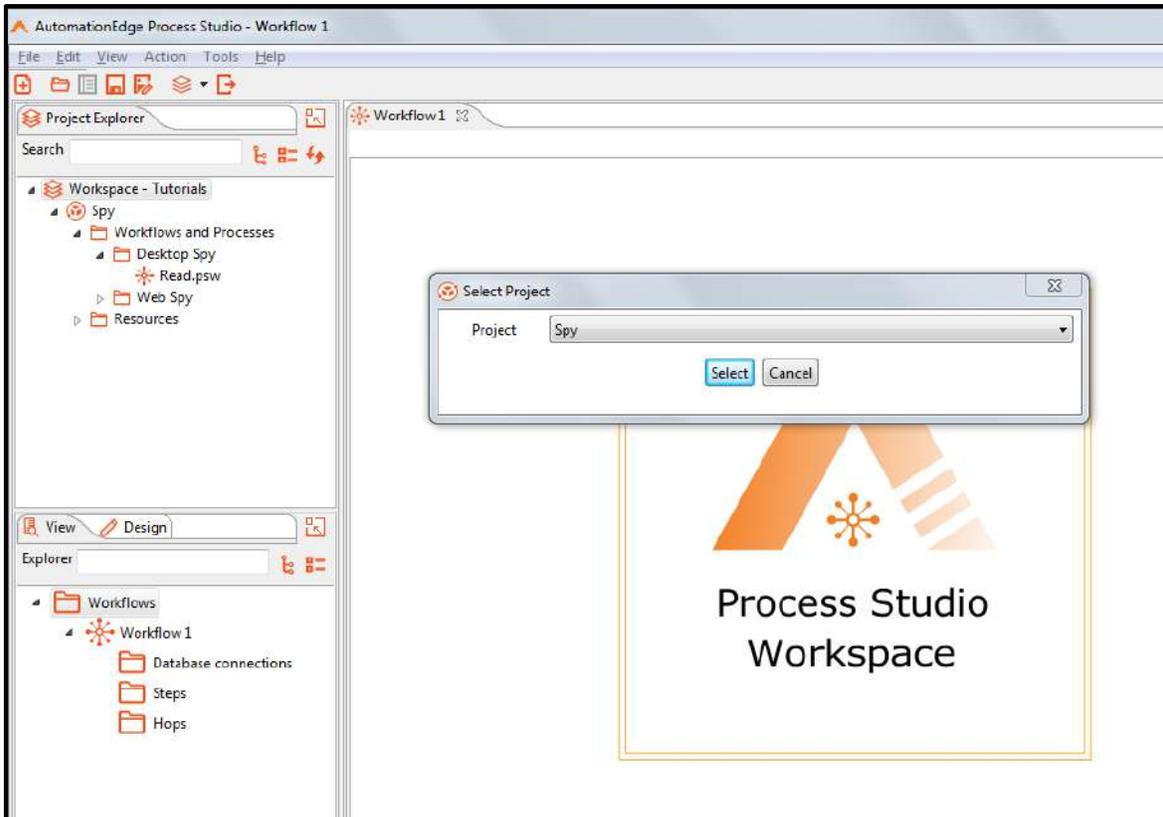
1. Start Process Studio. Launch the Tutorials workspace. In the Spy project select the Desktop Spy folder.



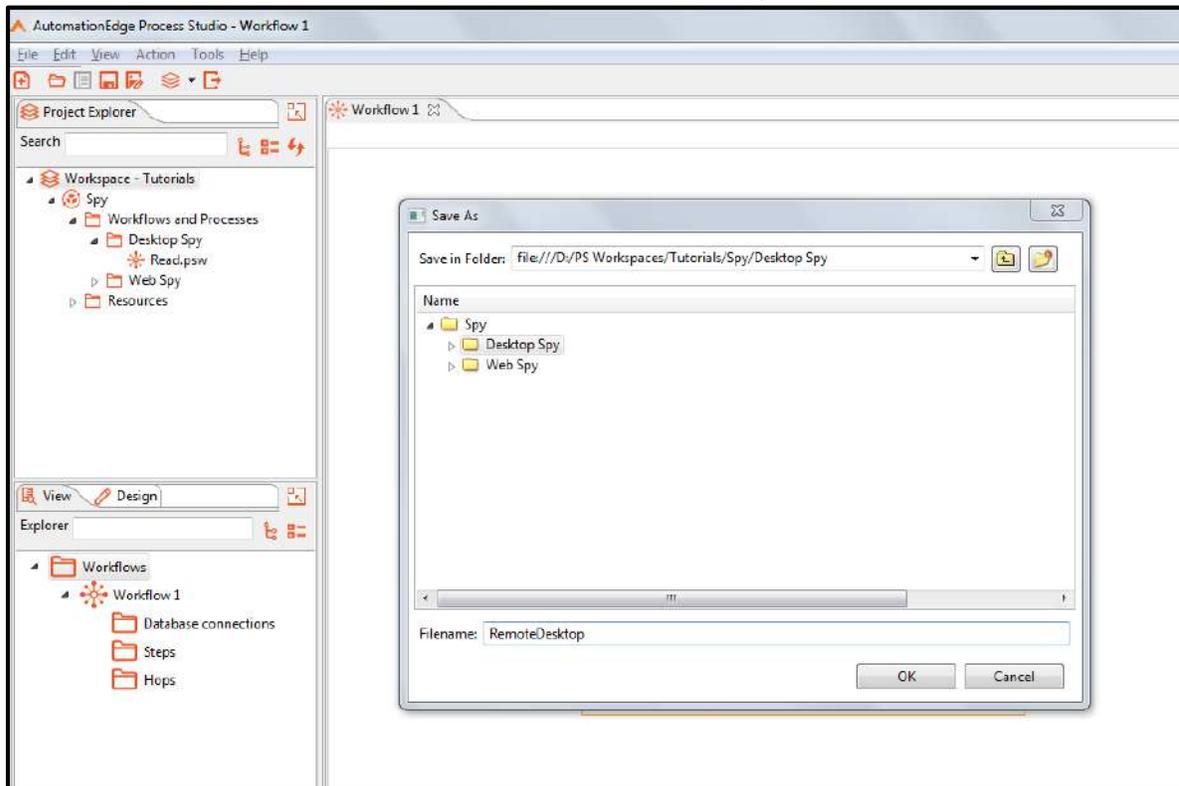
2. On the warning message click OK to create a Workflow.



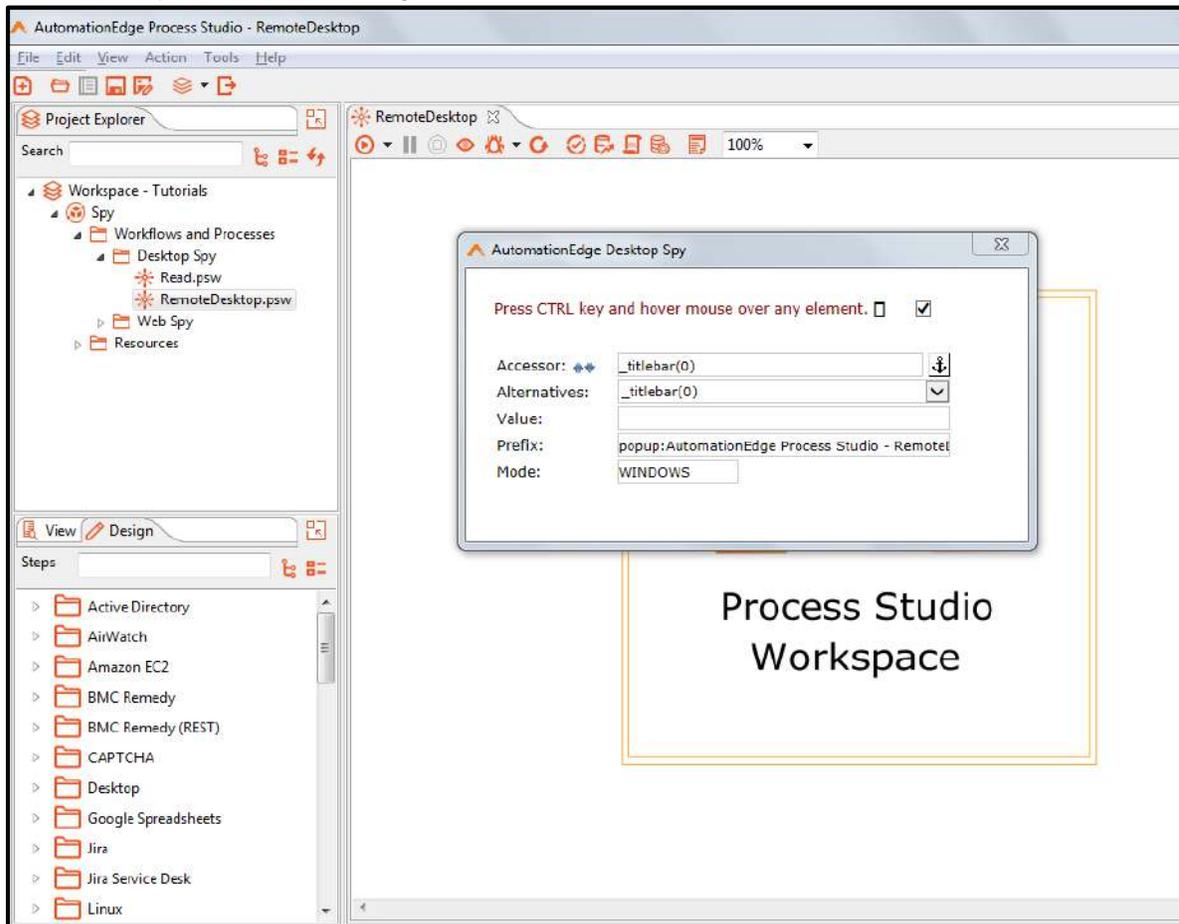
3. Select a project from the drop down list for to store the workflow.



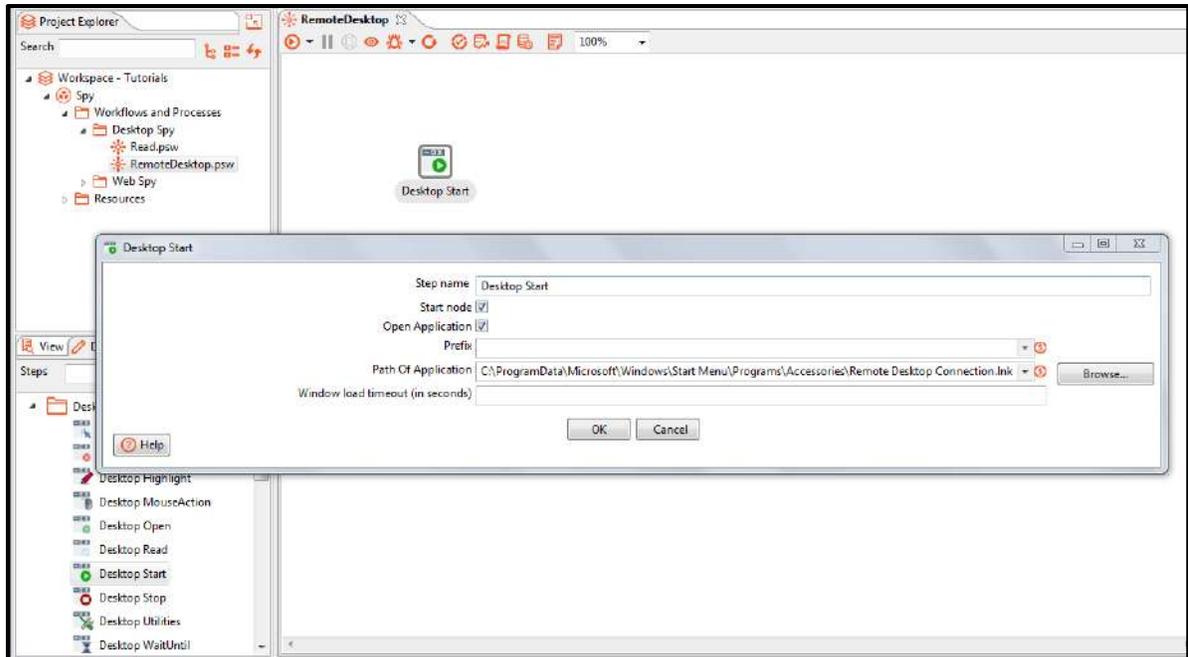
4. Save the workflow as 'RemoteDesktop', in the Desktop Spy folder.



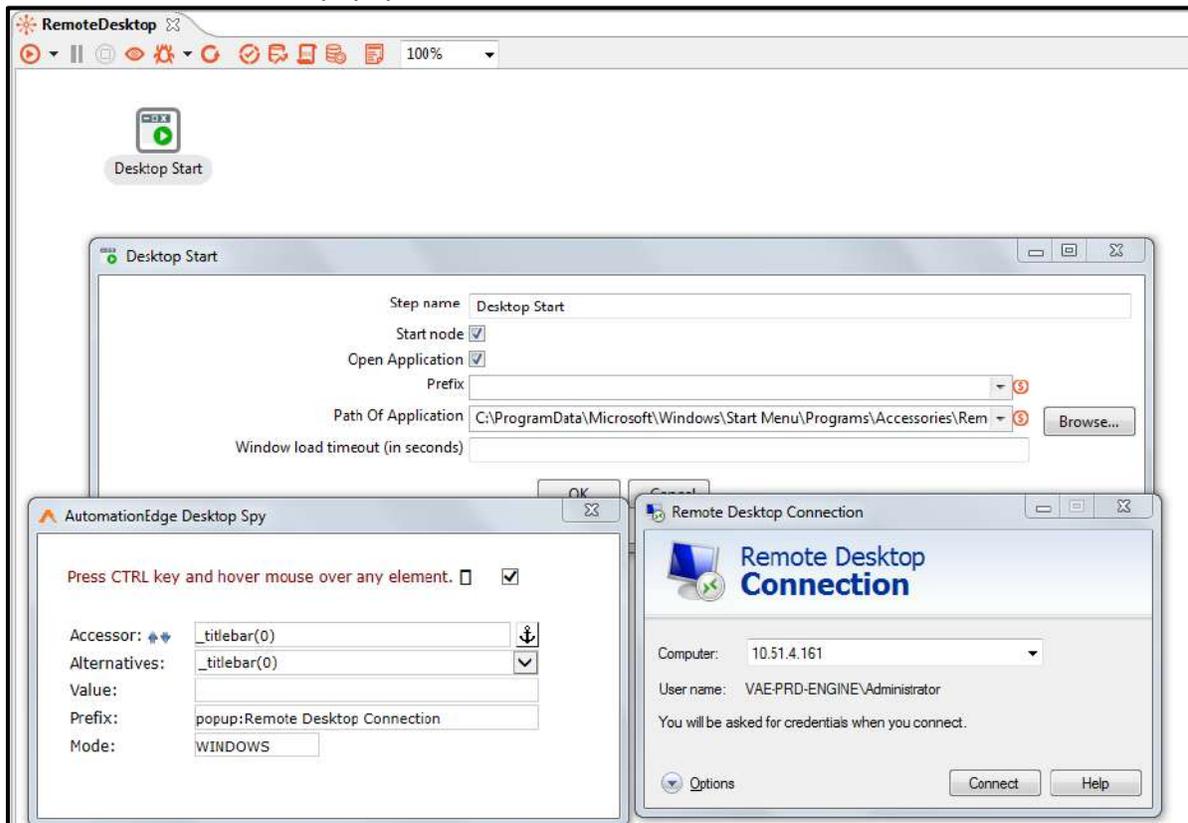
5. Desktop Spy opens in the foreground.



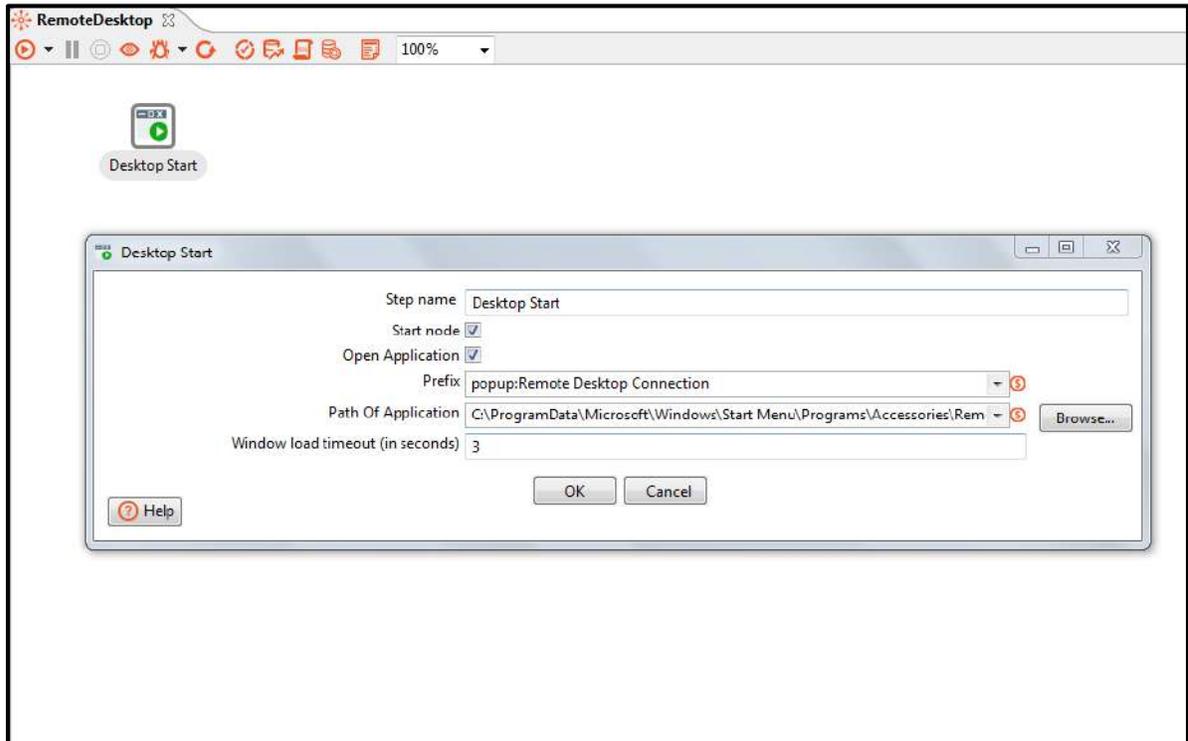
6. Drag and Drop Desktop Start step as the first step in the workflow.
7. In Desktop Start step we shall provide the path of 'Remote Desktop Connection' window -
C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Accessories\Remote Desktop Connection.lnk



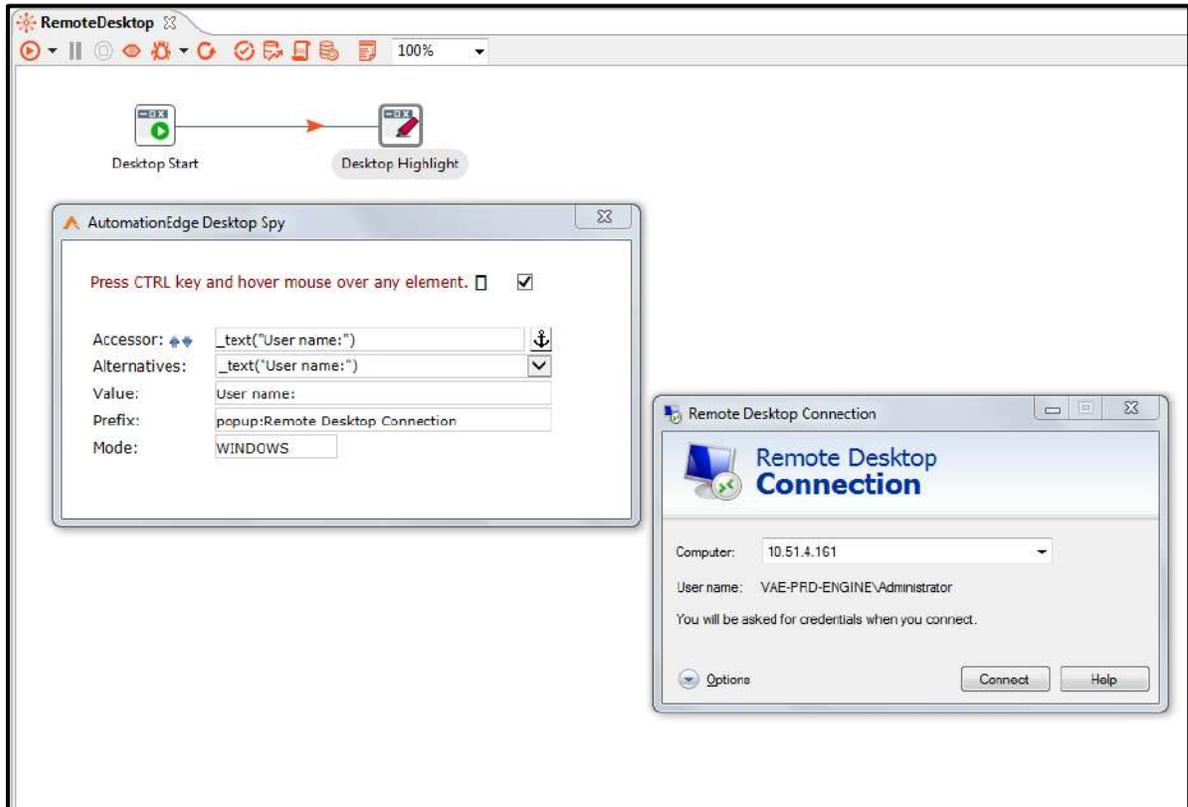
8. We need to spy the 'Remote Desktop Connection' pop-up. In the screenshot below we hover over the title area of the popup



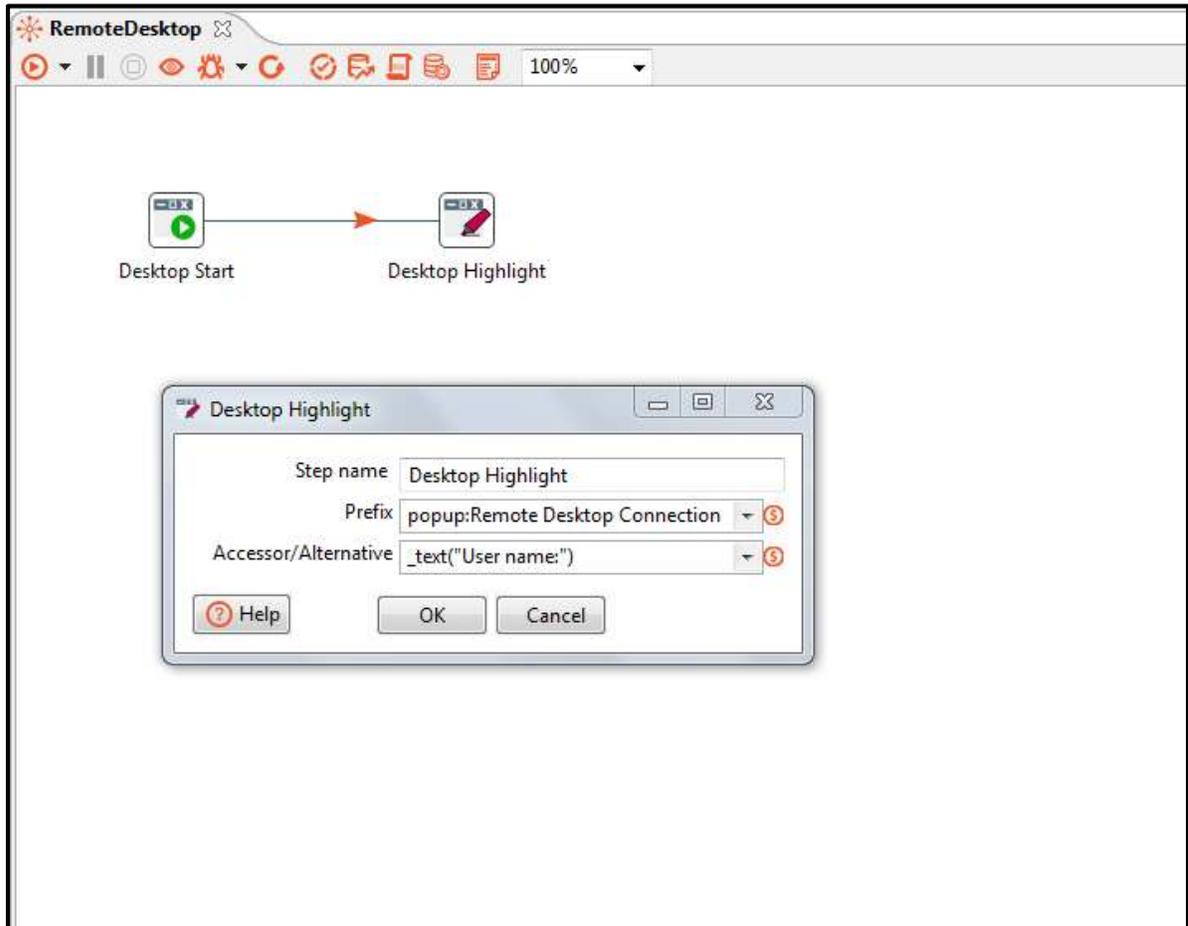
9. Add Start Browser step. Configure the step as shown below.
10. Paste the Prefix. Now you can provide a Timeout and click OK.



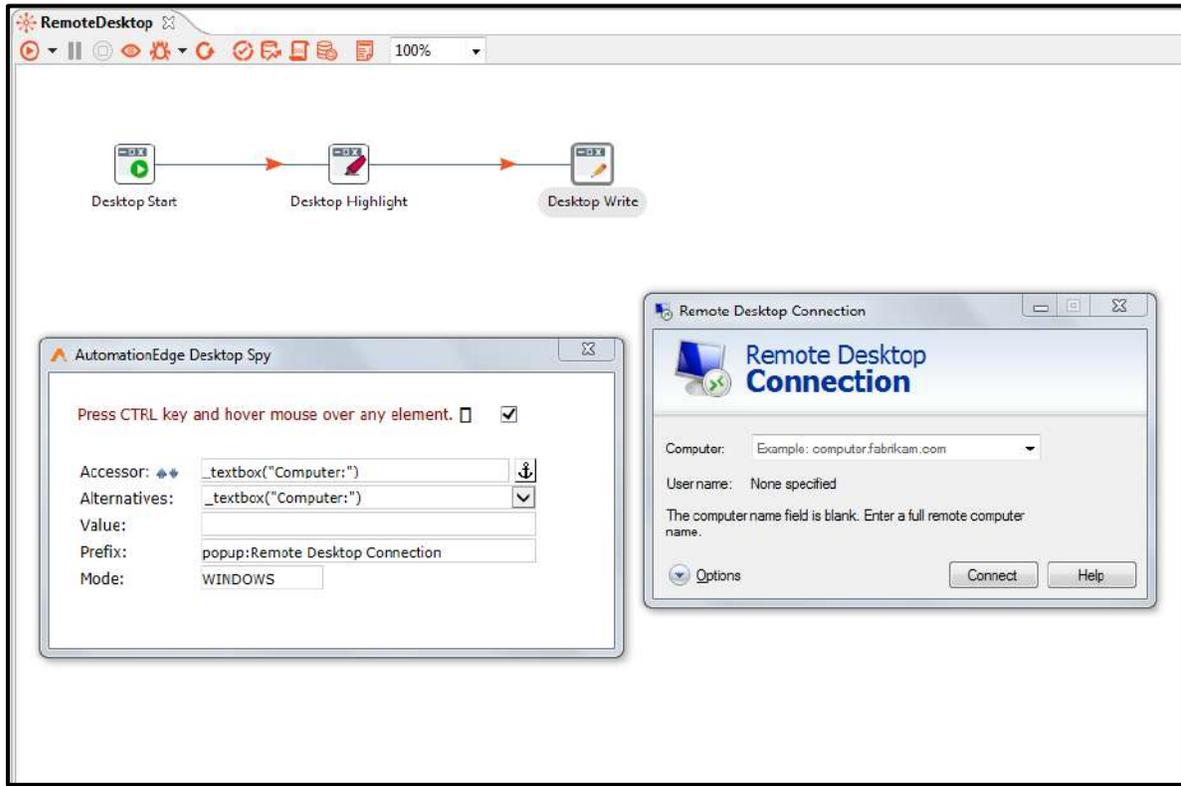
11. Next we wish to highlight User name on the 'Remote Desktop Connection' pop-up. Hover over User name and click ctrl button on the keyboard. Desktop Spy is populated as seen below.



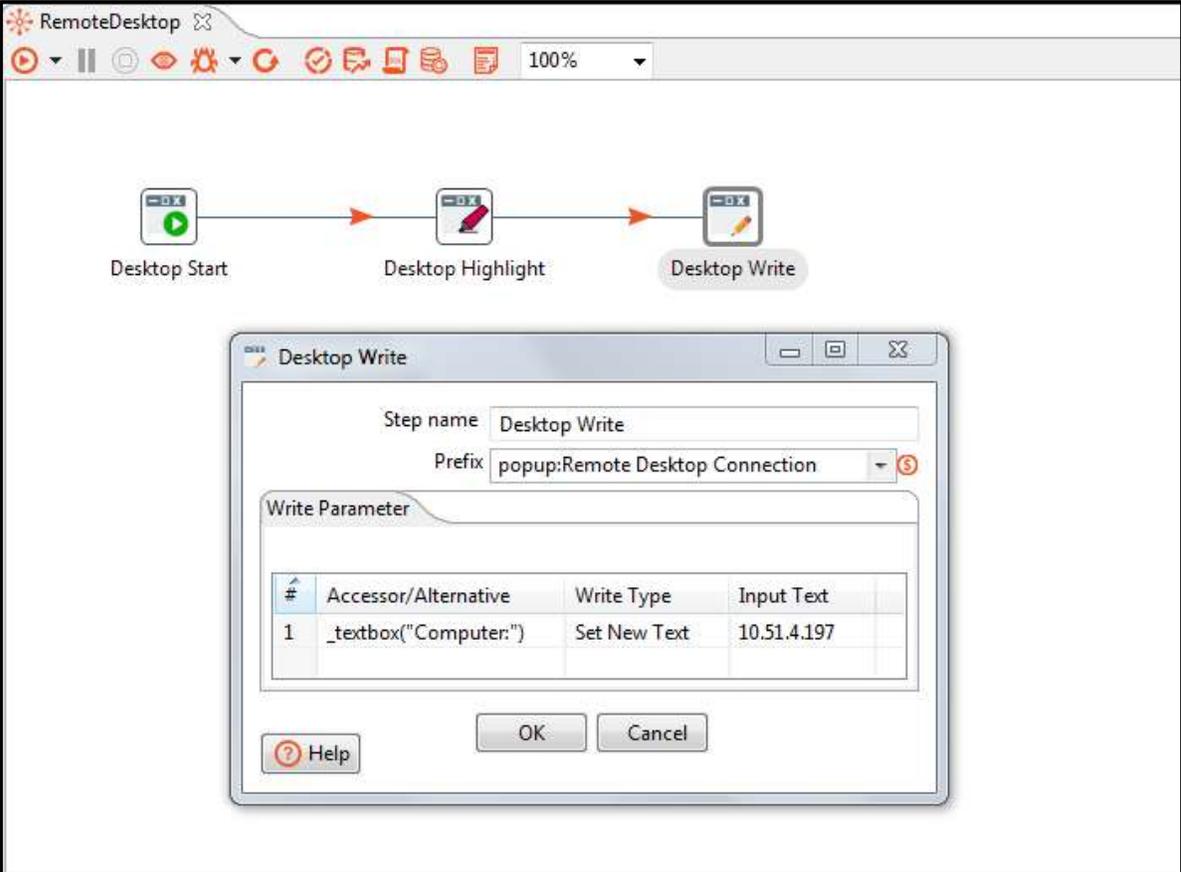
12. Copy the Prefix and Accessor/Alternative on the step configuration fields as seen below. Click OK.



13. Next we wish to write the Fully Qualified Domain Name/IP in the Computer field. Hover over the computer field and press Ctrl on the keyboard. The Desktop Spy is populated as seen below.



14. Configure the Desktop Write step with Prefix and Accessor/Alternative values as seen below. In the Write Type select Set New Text from the drop down list. Provide a value in the input Text.



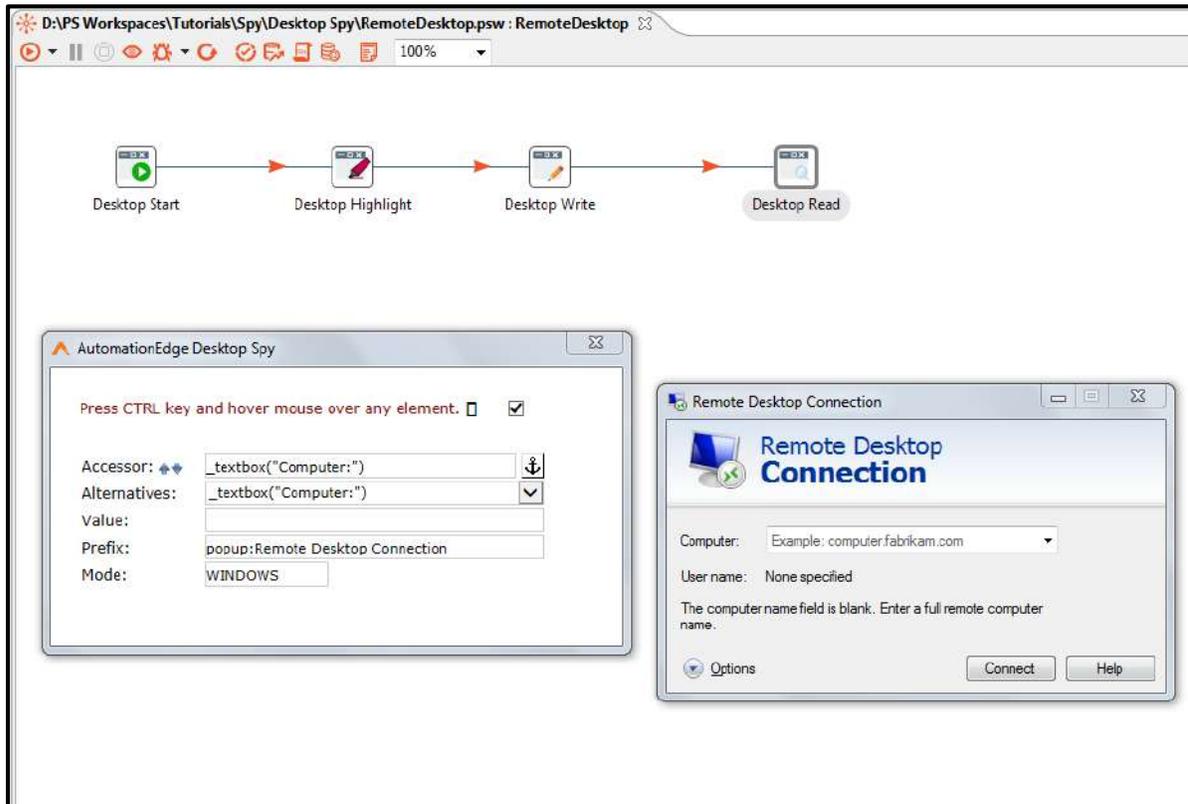
The screenshot shows the RemoteDesktop application interface with a workflow consisting of three steps: Desktop Start, Desktop Highlight, and Desktop Write. A configuration dialog for the Desktop Write step is open, displaying the following settings:

- Step name: Desktop Write
- Prefix: popup:Remote Desktop Connection
- Write Parameter table:

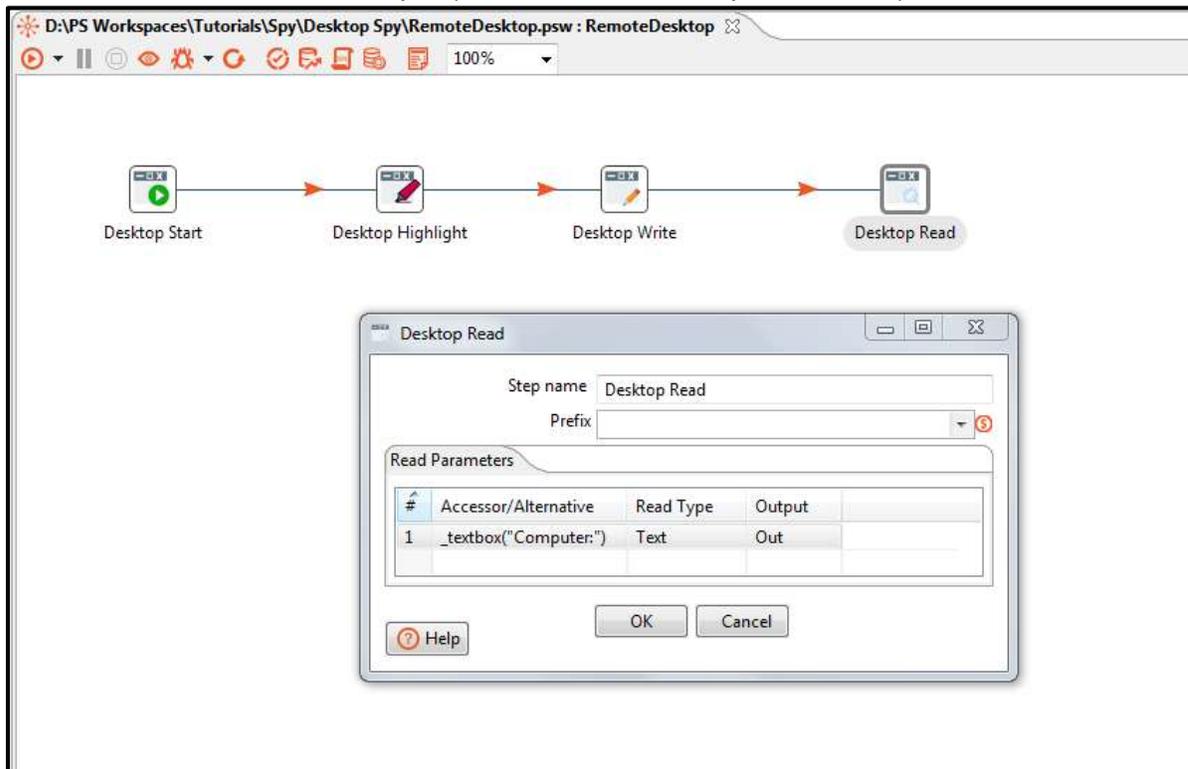
#	Accessor/Alternative	Write Type	Input Text
1	_textbox("Computer:")	Set New Text	10.51.4.197

The dialog also includes a Help button, OK button, and Cancel button.

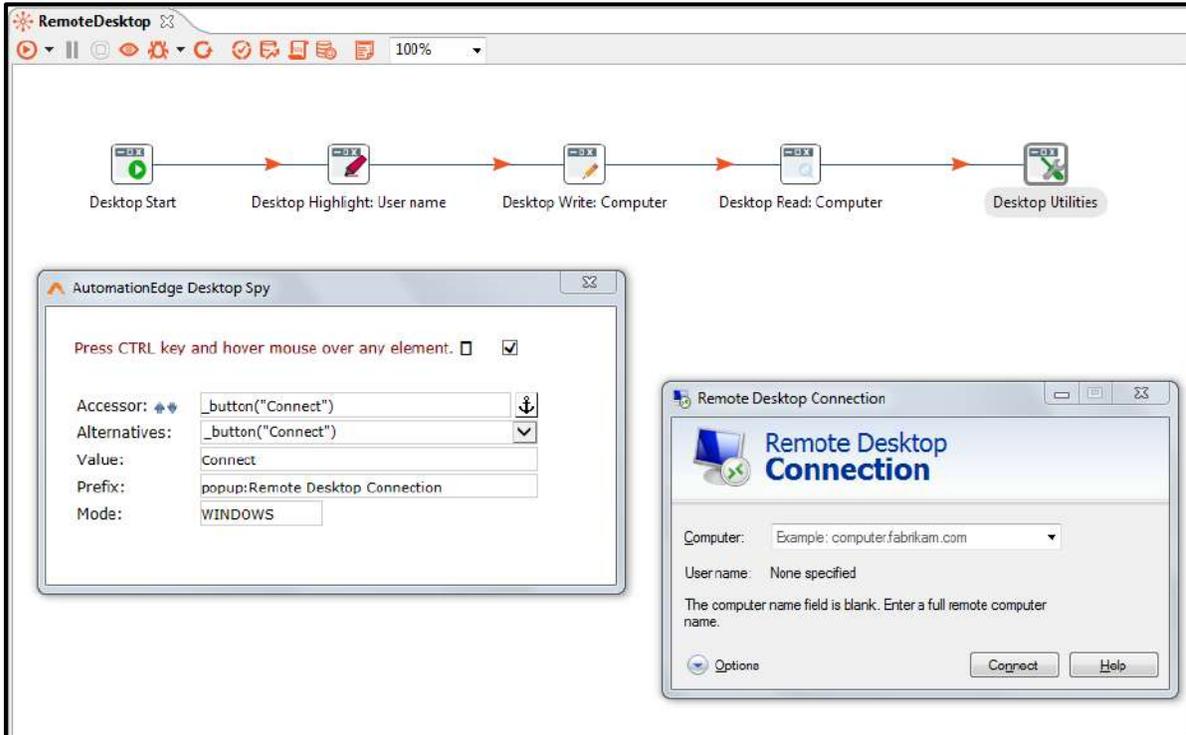
15. Next we wish to read the value that we wrote in the computer field earlier. Drag and drop a Desktop Read step. Hover over the Computer field and press Ctrl on the keyboard to get its attributes.



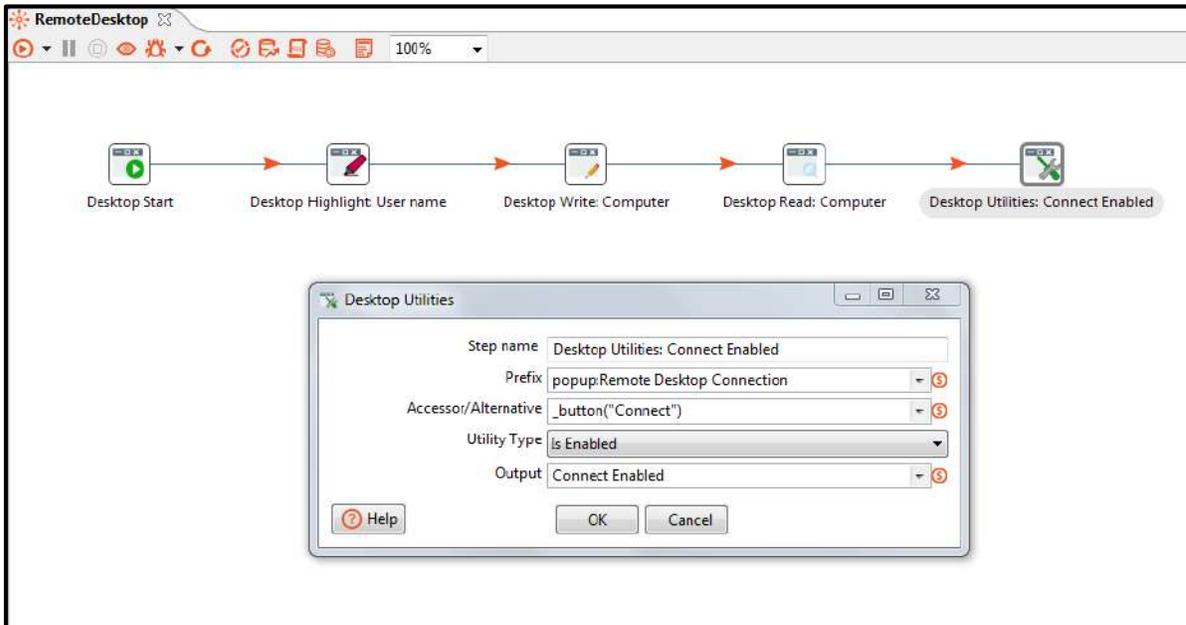
16. Configure the Desktop Read step with the attributes. Select Read Type Text from the drop down list. Provide a name for an Output (In this case we have provided Out).



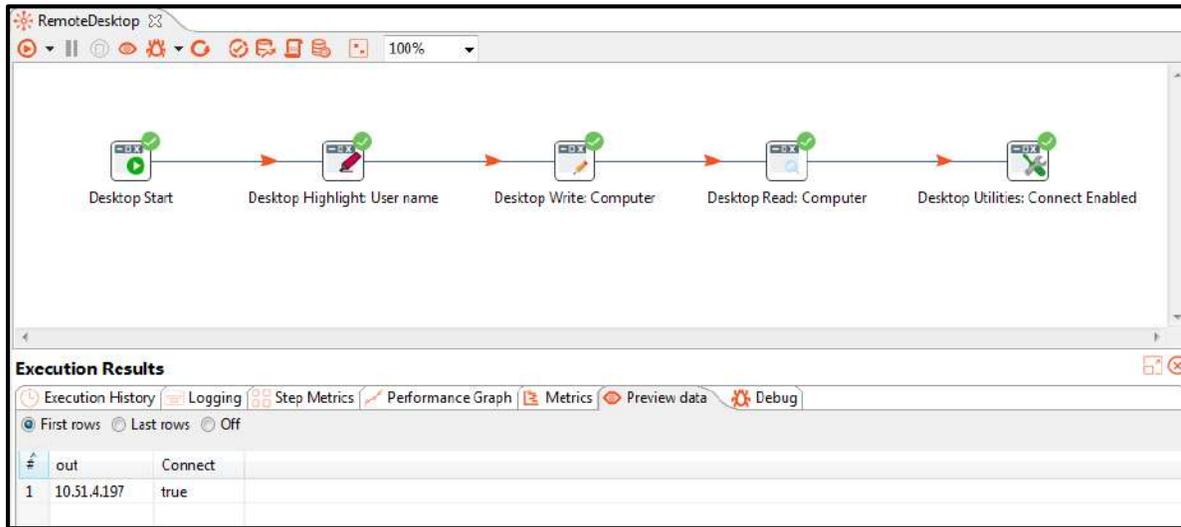
17. Next we want to check if connect button is enabled. We need a Desktop Utilities step for that.
18. Hover over the button and press Ctrl button on the keyboard to get the attributes on the Desktop Spy.



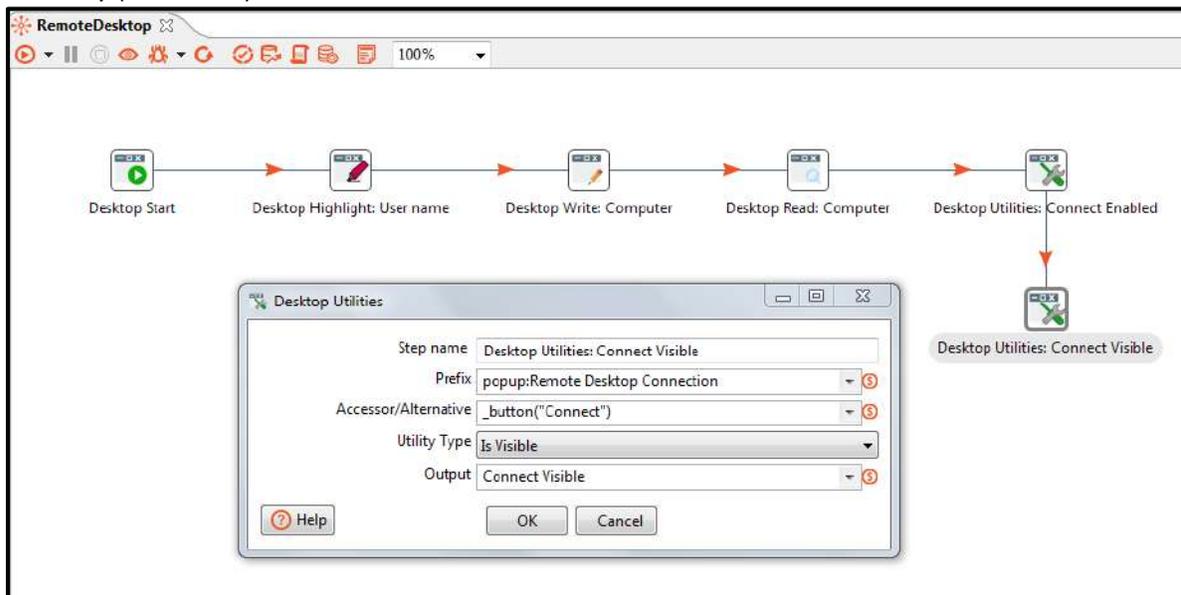
19. Configure the step as seen below. Provide a name which is self-explanatory as seen below. Select a Utility Type from the drop down list. Provide an Output field name to store the result of this step(true/false).



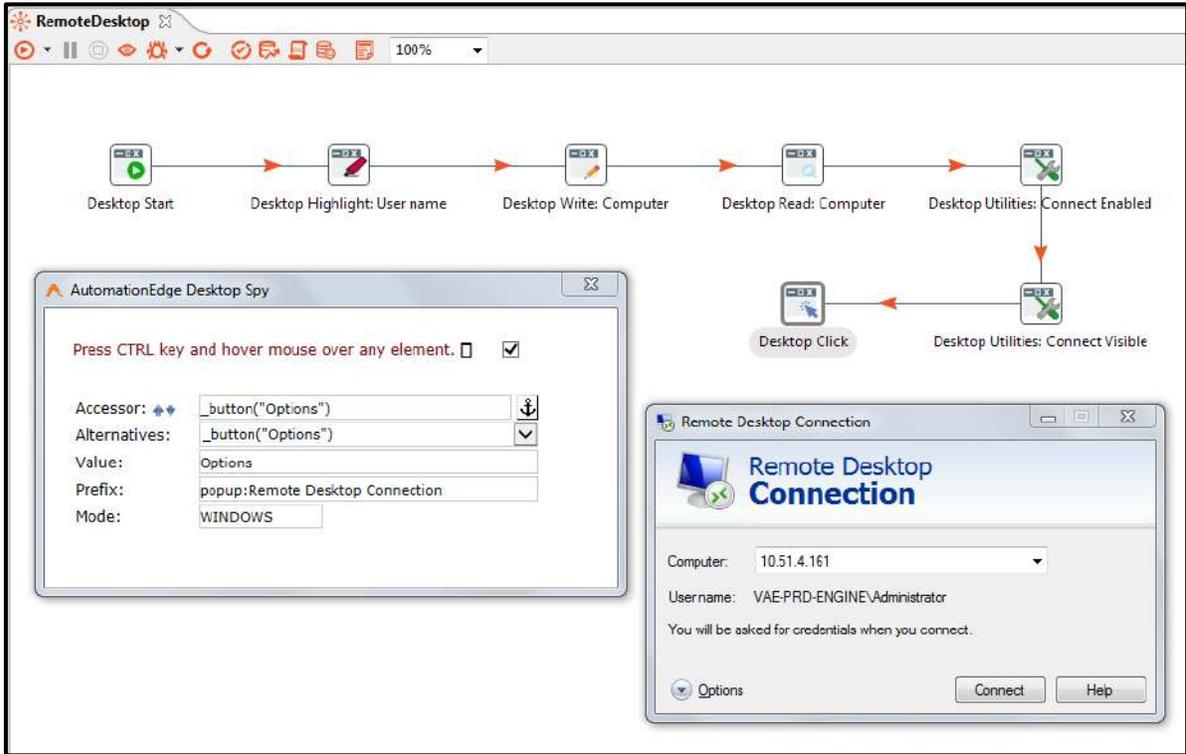
20. Next we want to check if connect button is visible. We need a Desktop Utilities step for that.
21. Hover over the button and press Ctrl button on the keyboard to get the attributes on the Desktop Spy.



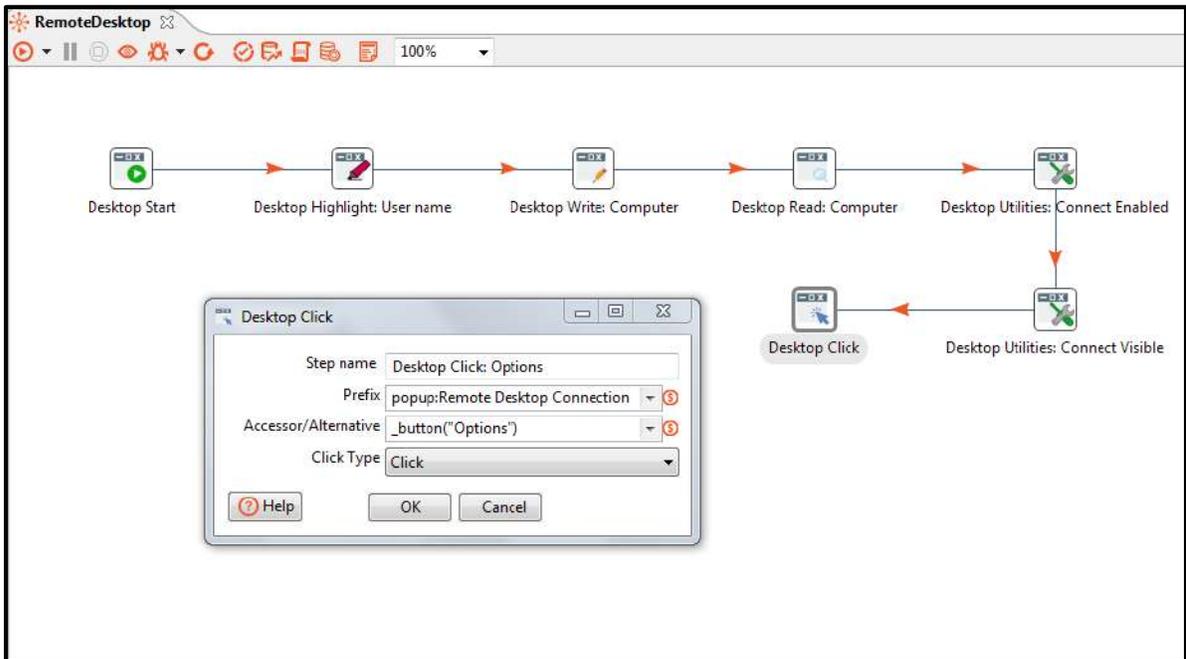
22. Configure the step as seen below. Provide a name which is self-explanatory as seen below. Select a Utility Type from the drop down list. Provide an Output field name to store the result of this step(true/false).



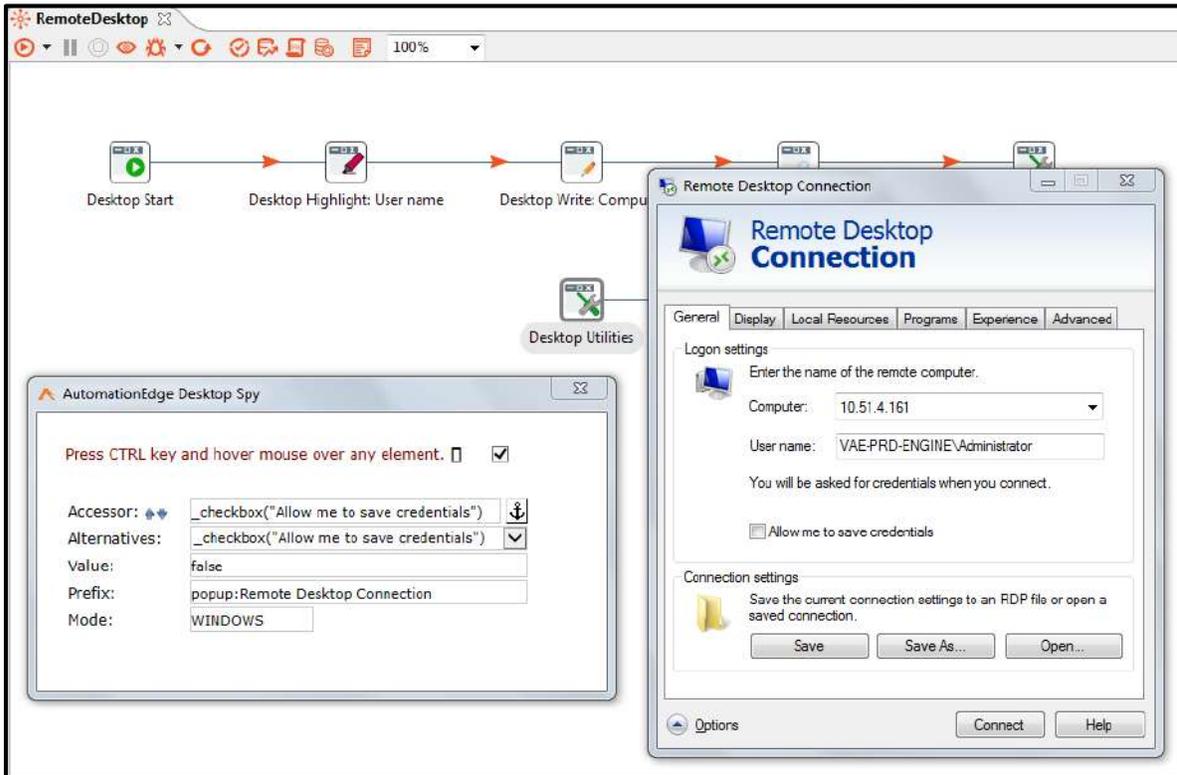
23. Next we want to check if connect button is enabled. We need a Desktop Utilities step for that.
24. Hover over the button and press Ctrl button on the keyboard to get the attributes on the Desktop Spy.



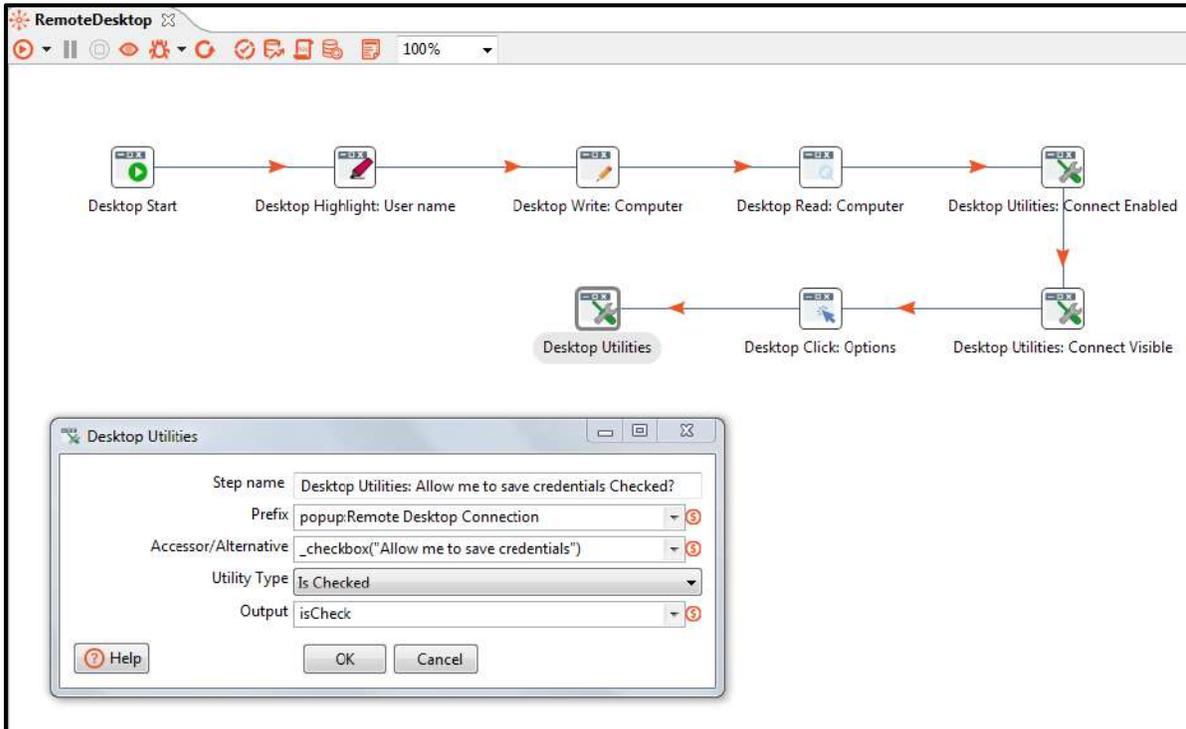
25. Copy Prefix and Accessor/Alternative from Desktop Spy to the step configurations. The Click Type can be chosen from the drop down list.



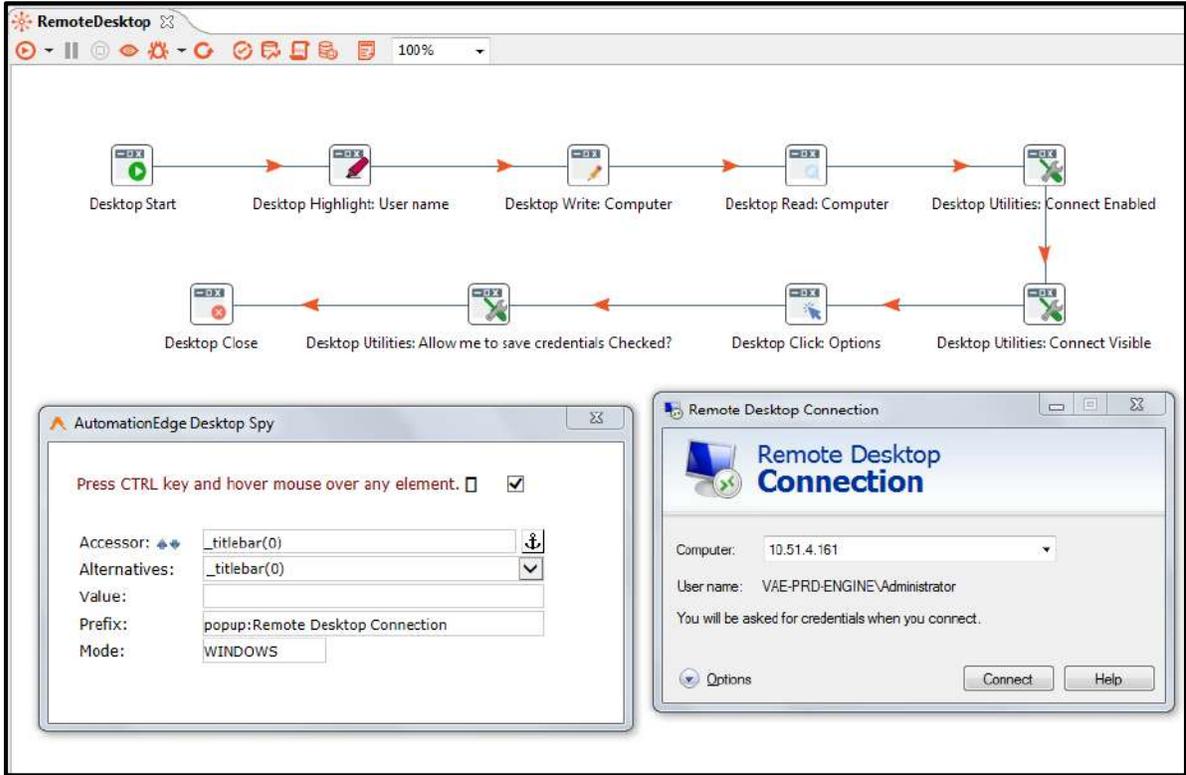
26. Next we want to check if 'Allow me to save Credentials' is checked. Spy the checkbox.



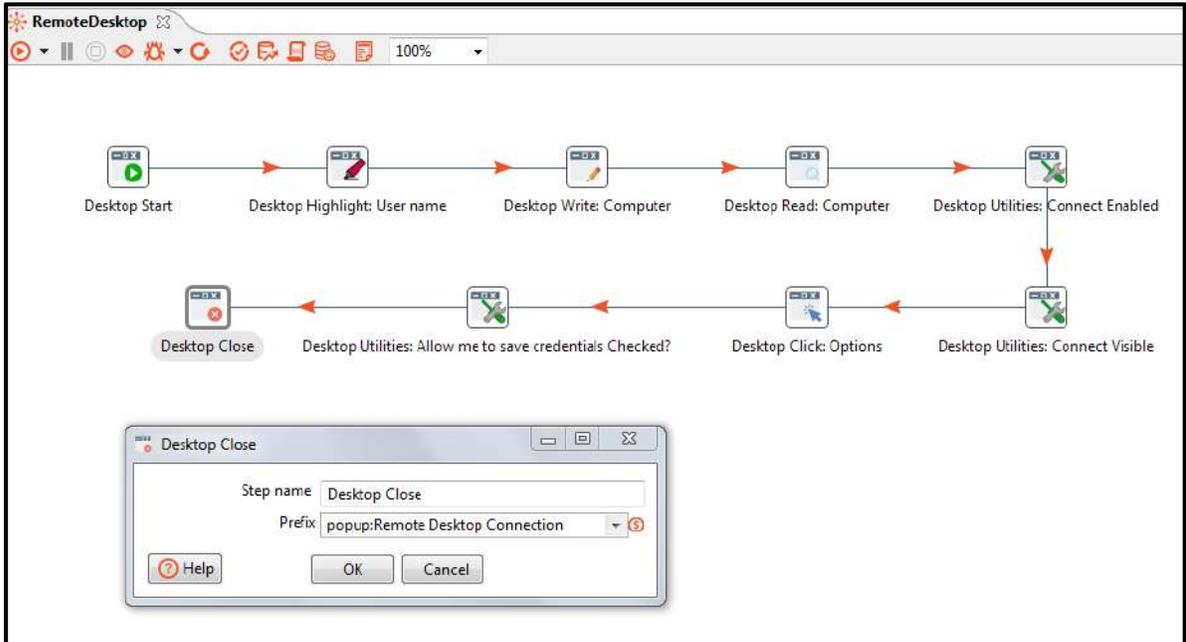
27. Copy the Spy attributes to the step configuration. Provide a suitable step name, Utility Type from the drop down and a field name to hold the step output(true/false).



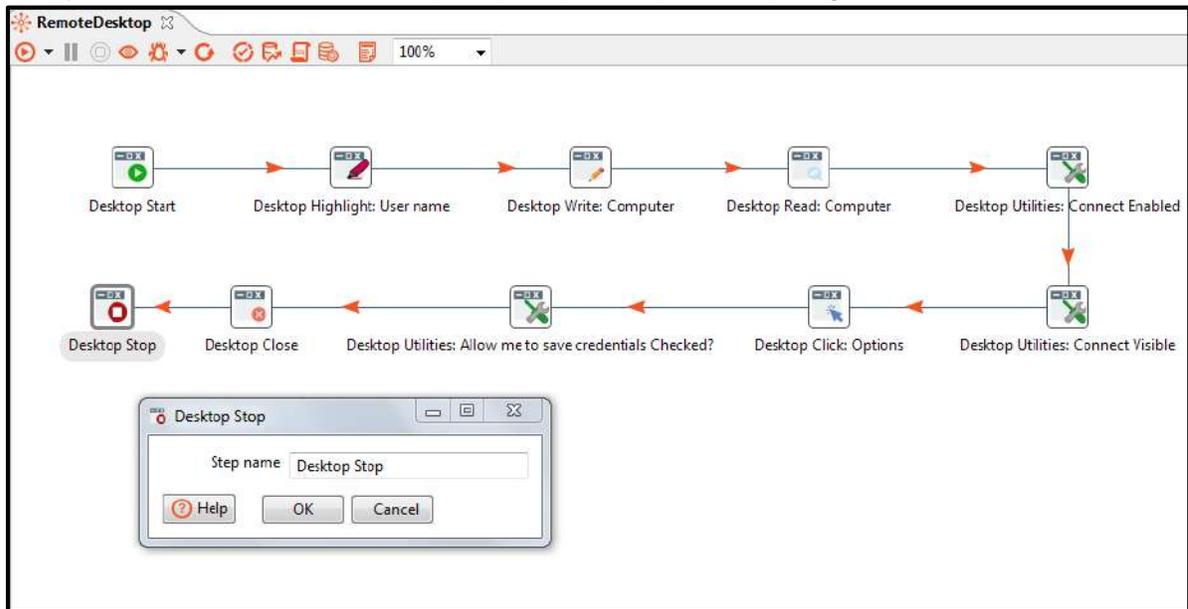
28. We can now Add the Desktop Close step to close the pop-up. Hover over the title area of the 'Remote Desktop Connection' pop-up and press Ctrl button.



29. Configure the step with the prefix for the 'Remote Desktop Connection' pop-up.



30. finally, we should add the Desktop stop step to stop Desktop engine.



31. This completes the RemoteDesktop workflow. We have used several Desktop Plugin steps and explored the configuration options. A complete description of all the Desktop Plugin steps is available in [AutomationEdge_Plugin_Reference_Guide_R2.2](#).

15 Project 8: Surface Spy

The project - Spy with folder Surface Spy is intended as a tutorial for Surface Spy.

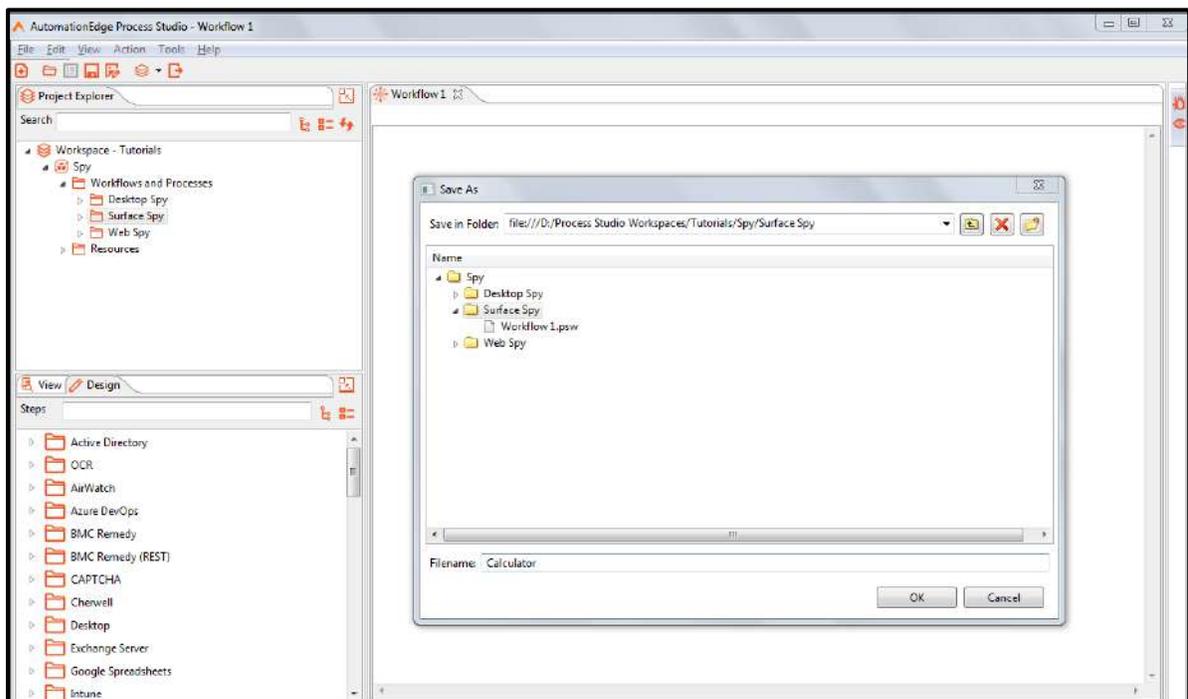
15.1 Sample Workflow with Surface Spy

15.1.1 Multiply Two Numbers using Calculator

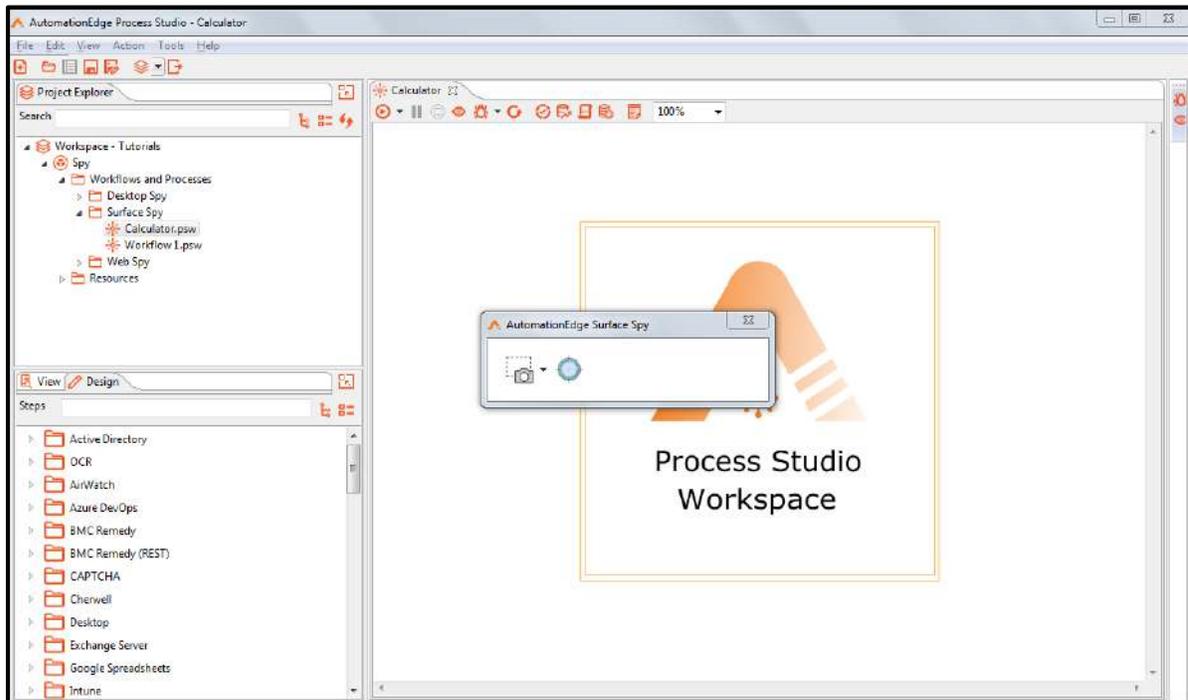
We will use Surface Plugin steps to create a workflow to multiply two numbers using Calculator. In this workflow we will use Match Pattern -**Retrieve Single closest match** as well as **Retrieve Multiple closest matches**.

Following are the steps,

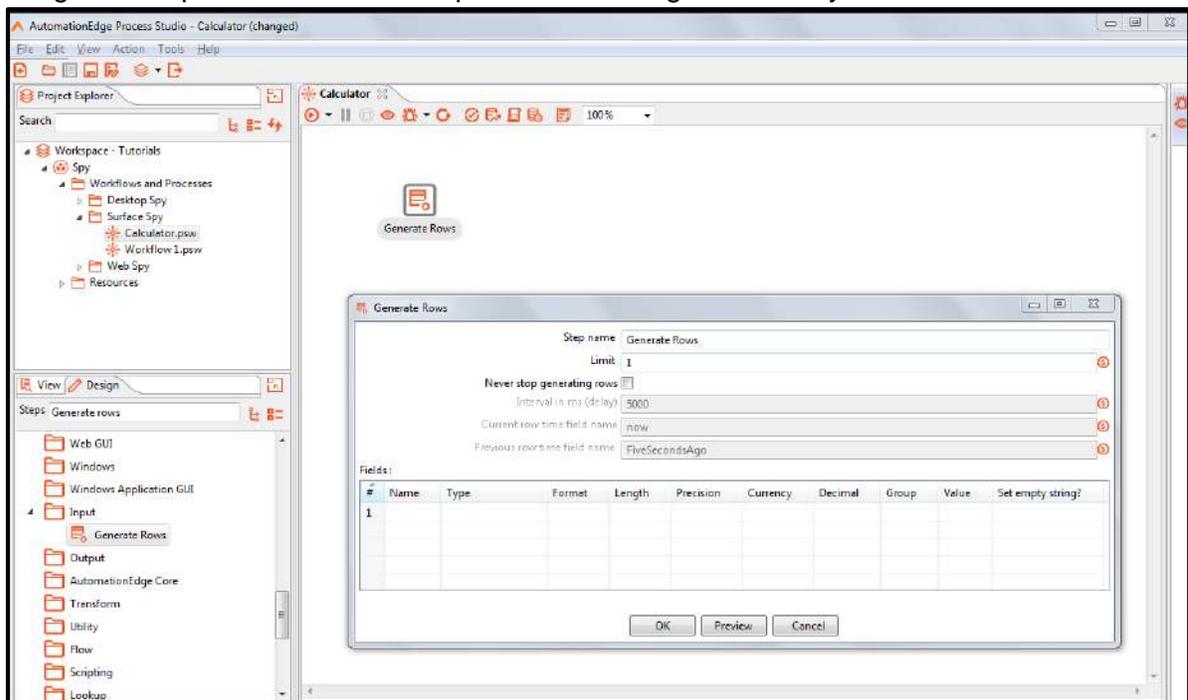
1. In Spy Project create a folder Surface Spy.
2. Add a new workflow from file menu or by clicking  icon.
3. Save the workflow with name as Calculator.



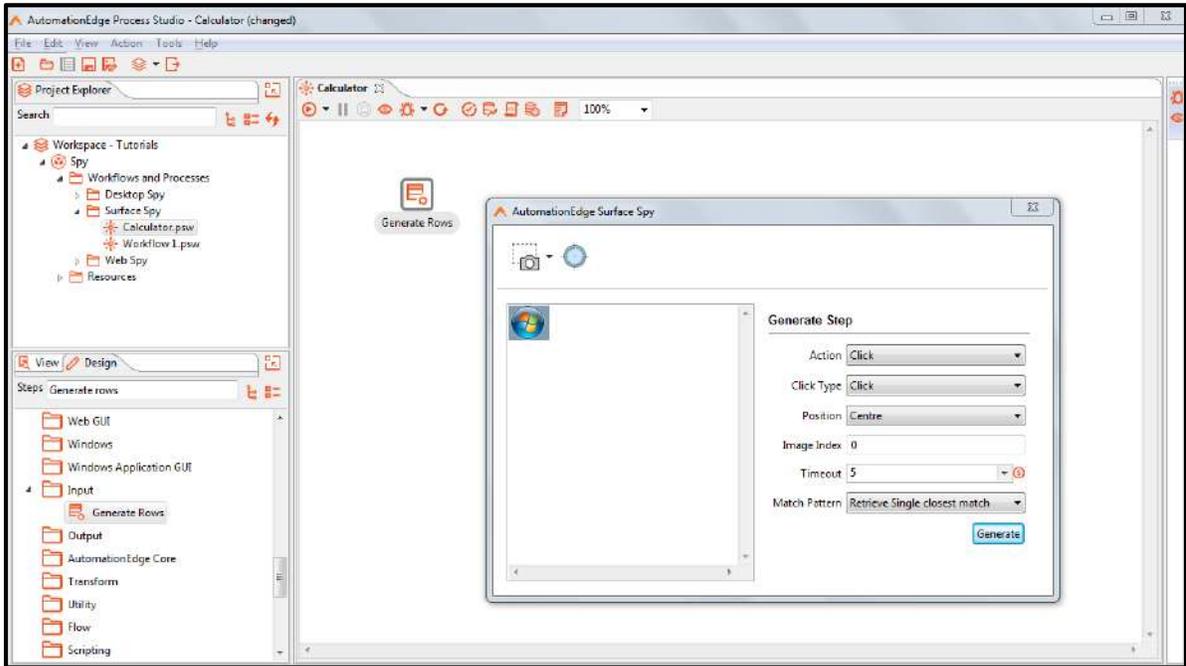
4. The Calculator workflow is now open. Open Surface Spy by navigating to Tools→Spy→Surface Spy
5. Alternately, directly navigate to Tools→Spy→Surface Spy without opening or creating a workflow. You are prompted to create a new workflow.
6. Calculator is created and AutomationEdge SurfaceSpy is opened in the foreground as seen below.



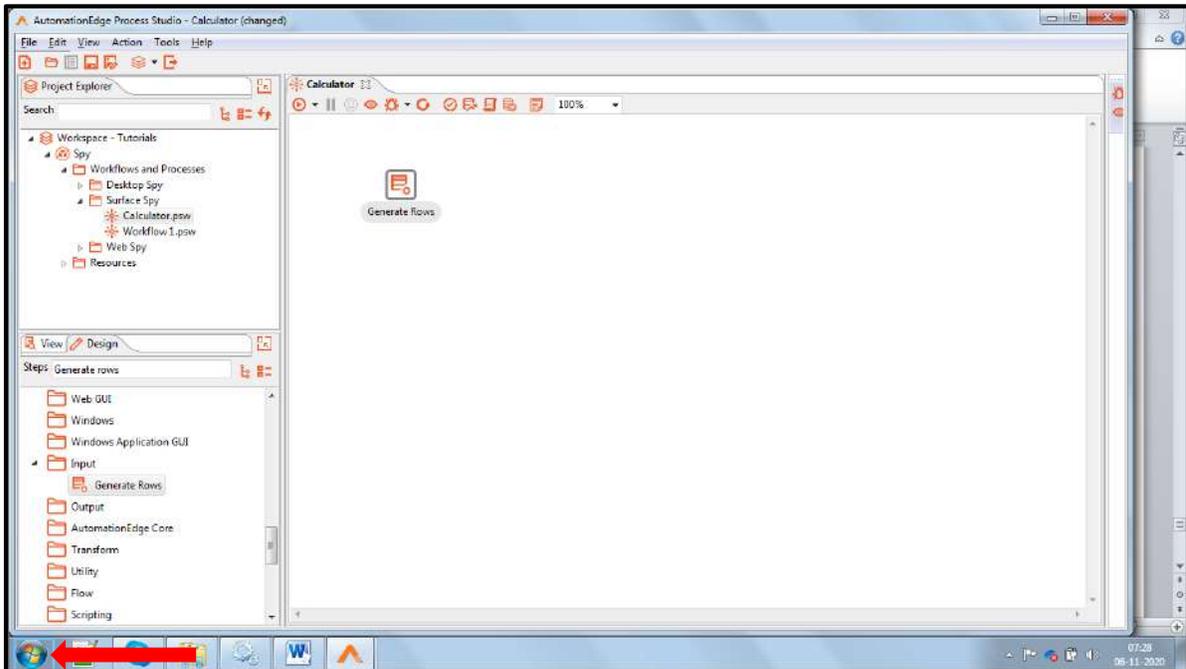
7. Drag and Drop Generate Rows step. Set Limit 1 to generate only 1 row.



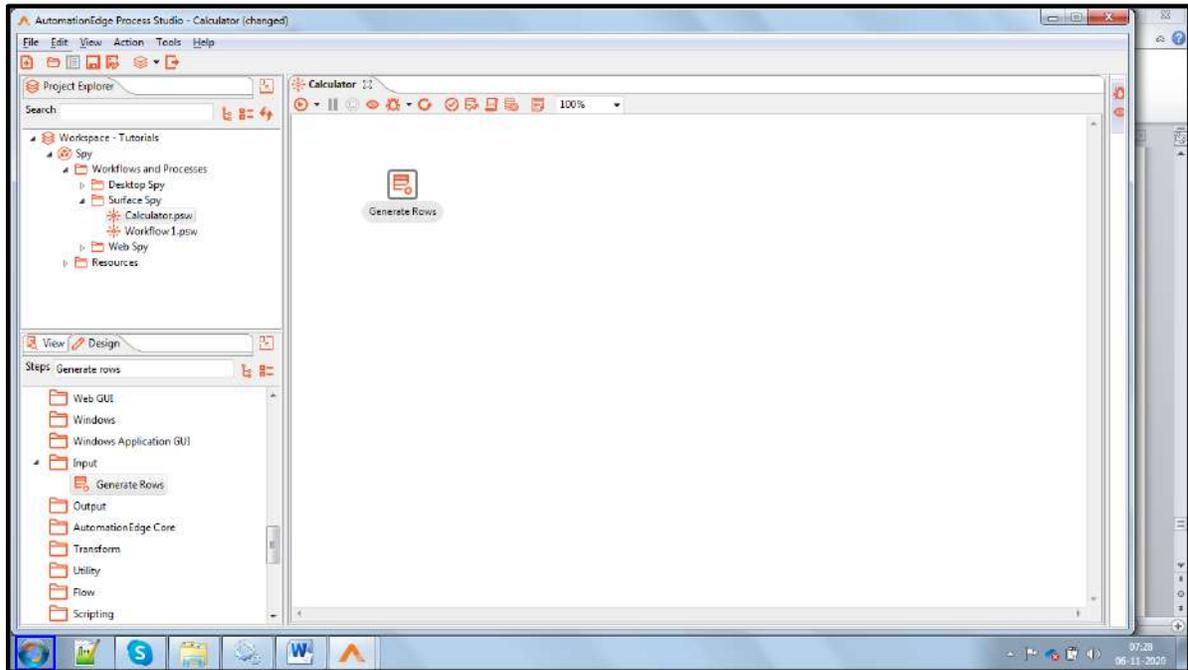
- Locate (Capture) Image Windows Start by clicking on the first icon (📷) for capture image and then drawing a square around Windows Start. The image captured can now be seen in GUI Spy.



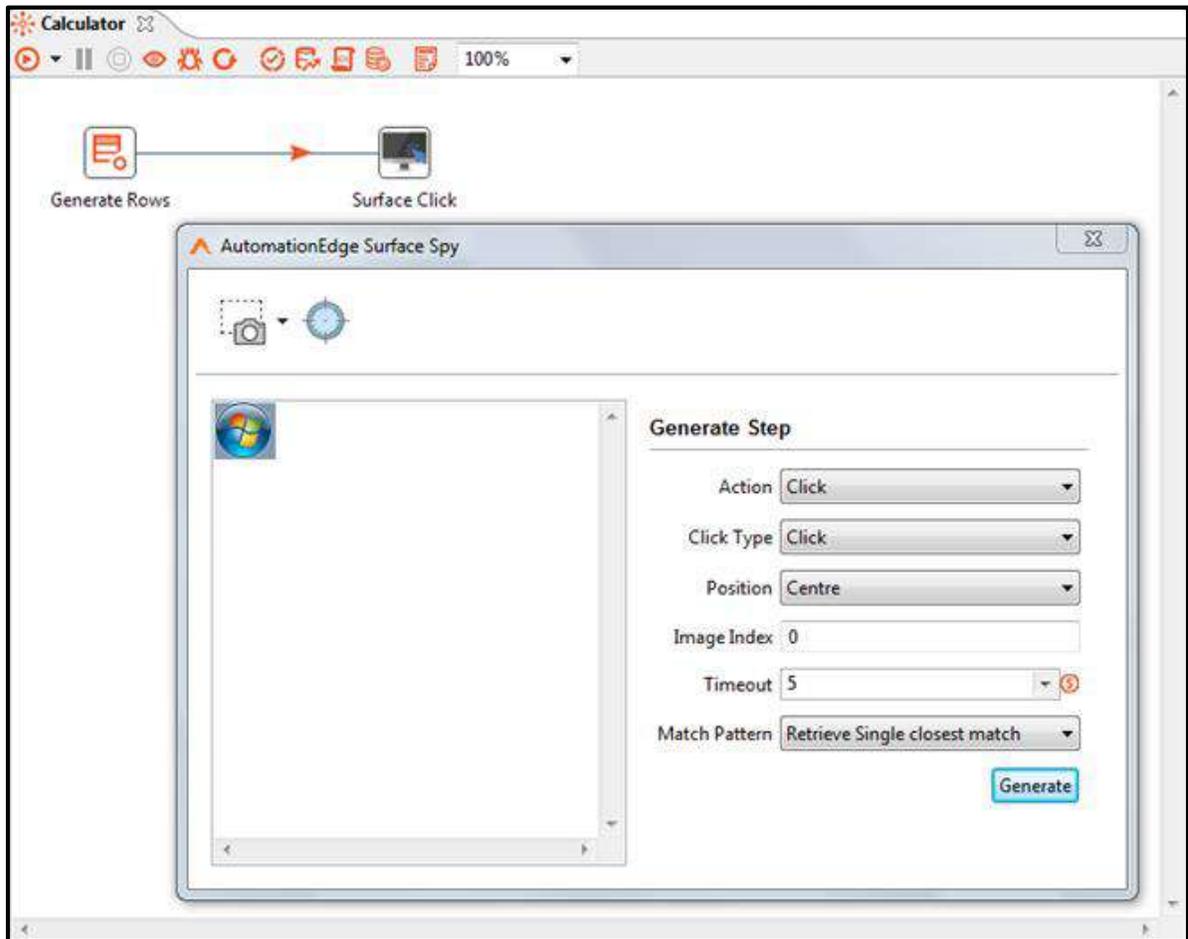
- Search for the located the image by clicking locate on Surface Spy. All matching images will be retrieved outlined with a light blue box. In the screenshot below one image is outlined with a blue box as seen below.



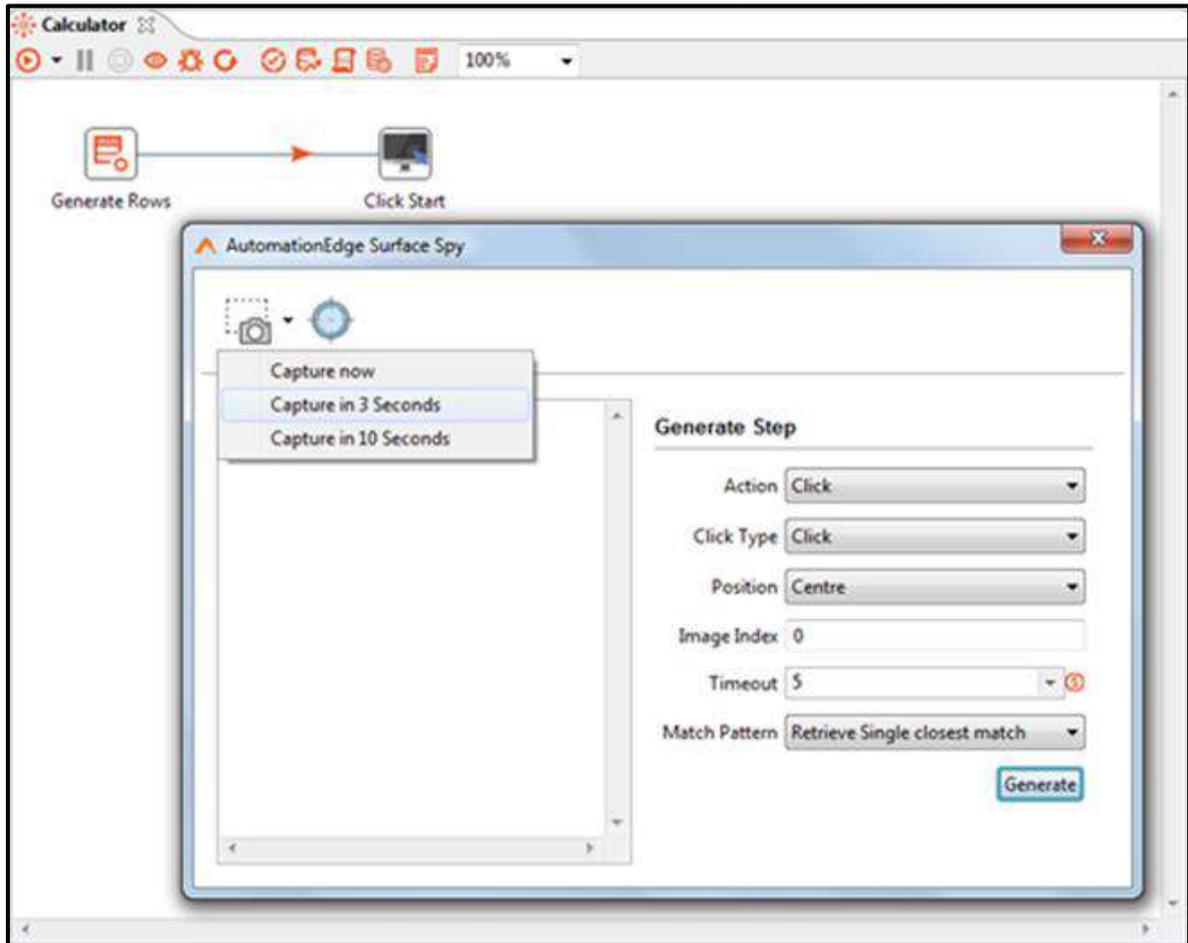
10. Click on the correct image among one or multiple images located. You can see a small blue circle on the selected image. Now click key Esc.



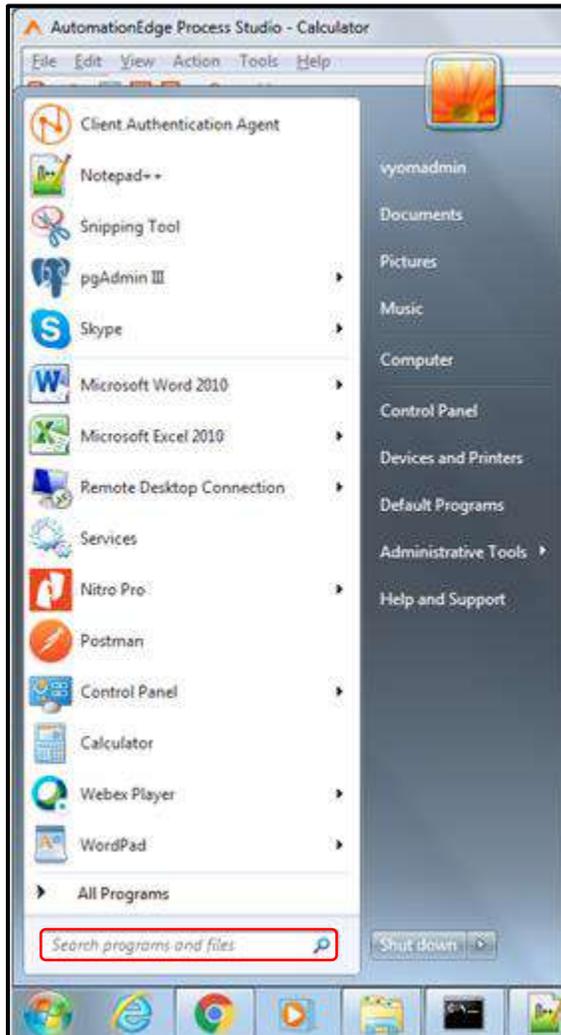
11. Click Generate. We will now be able to see a Surface Click step on the workflow as seen below.



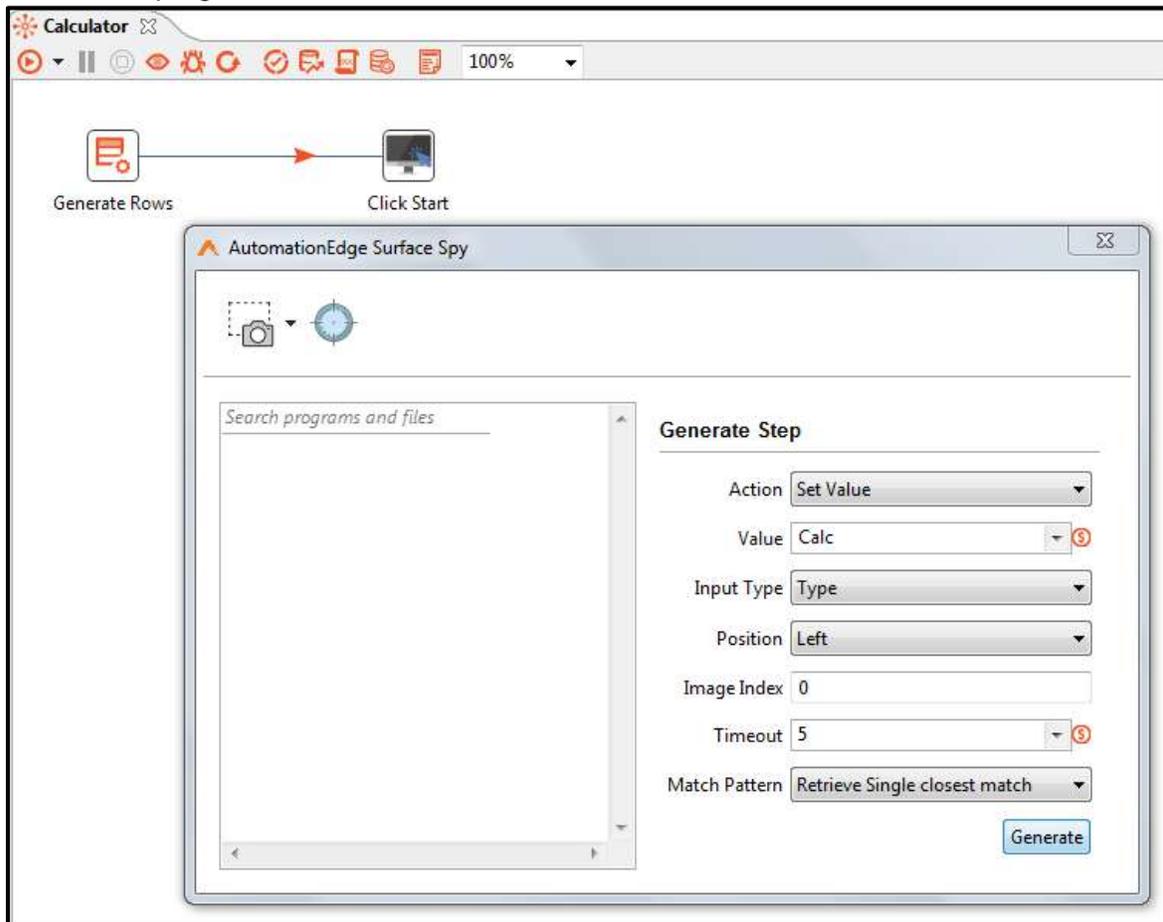
12. Rename the step Click Start. We will choose Position as Centre from the drop down.
13. Now on GUI Spy click Capture in 3 Seconds from the Capture Image drop down.



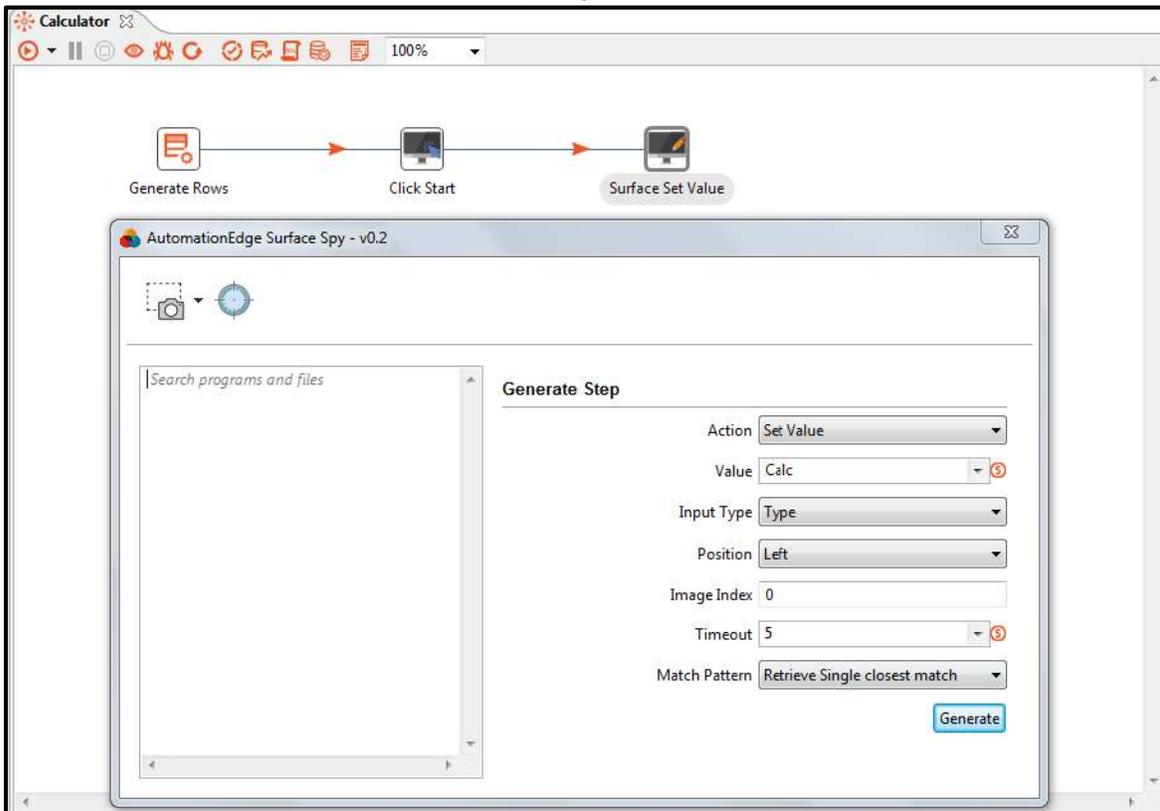
14. We wish to capture the image for Search programs and files as seen below outlined with a red box and Type Calc in the box.



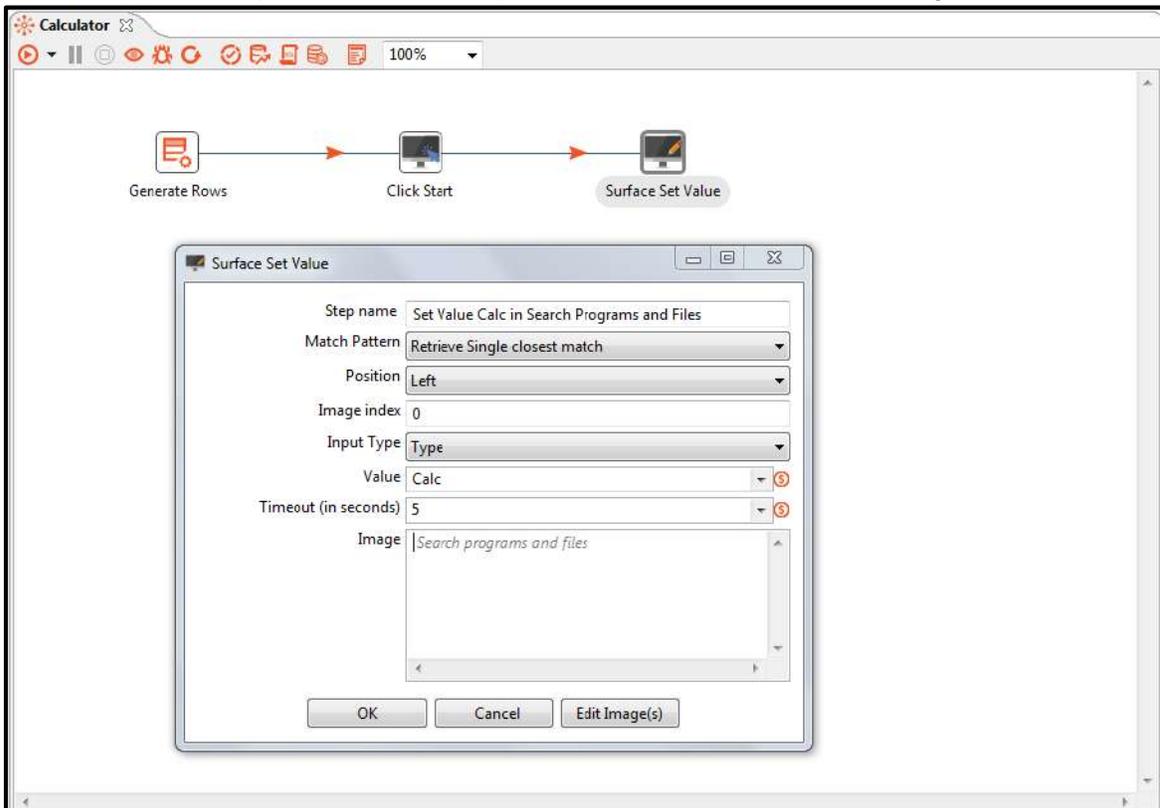
15. Now bring Surface Spy in the forefront. Change the Action to Set Value. We need to type Calc in the Search programs and files box. Set Value to Calc.



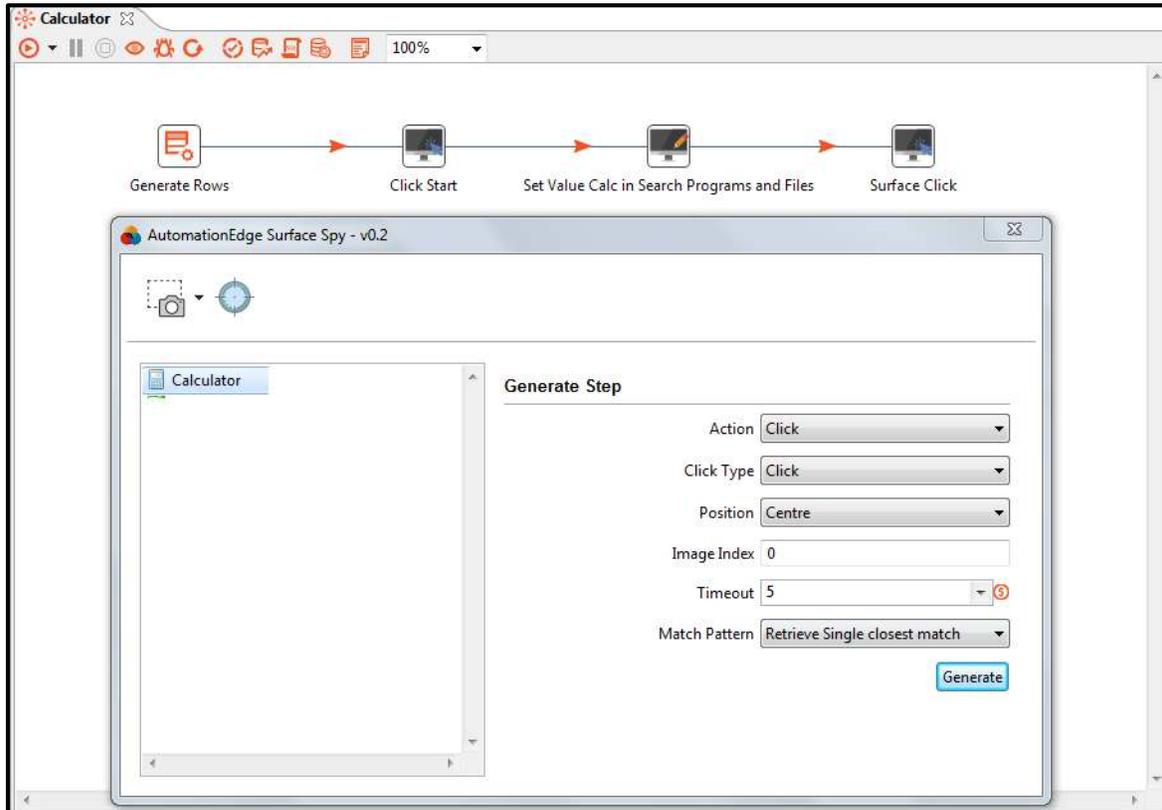
16. Click Generate. A Surface Set Value step is generated.



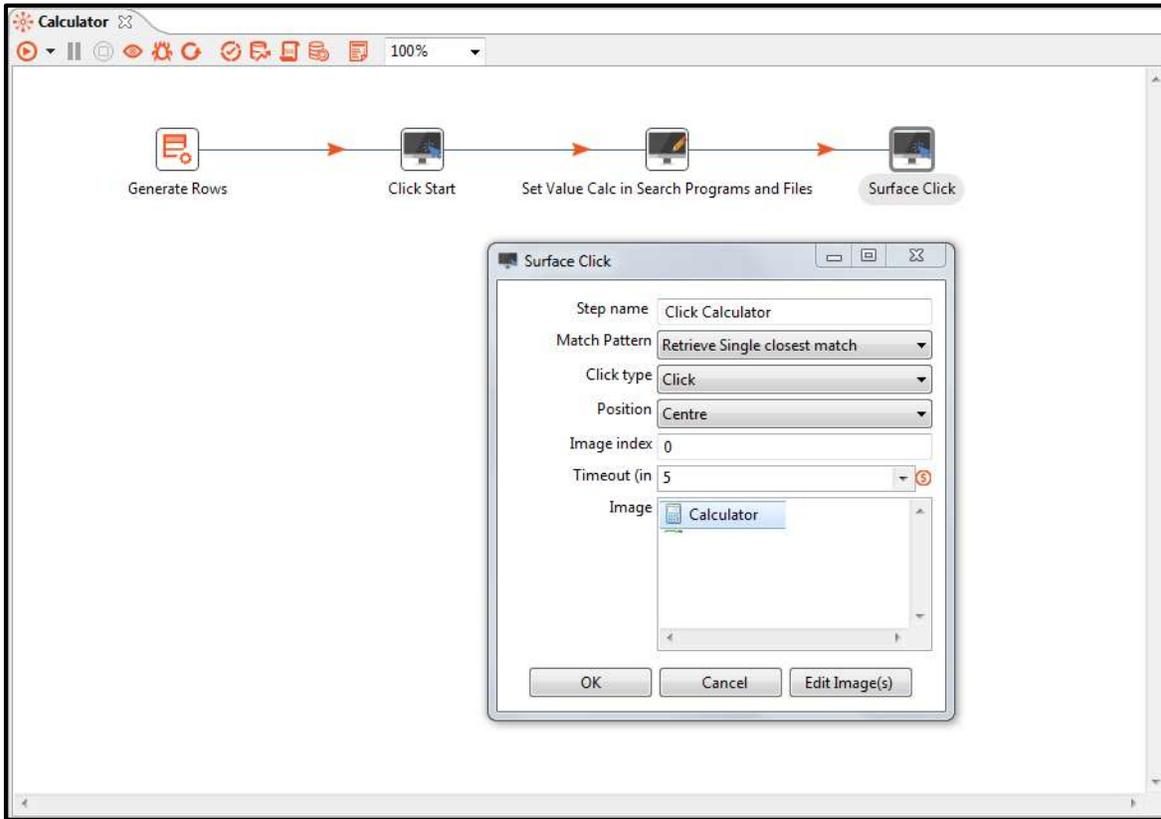
17. Double click the step. Rename the step to Set Value Calc in Search Programs and Files.



- Next locate image for Calculator as shown below.
- Click Generate. A Surface Click step is generated.

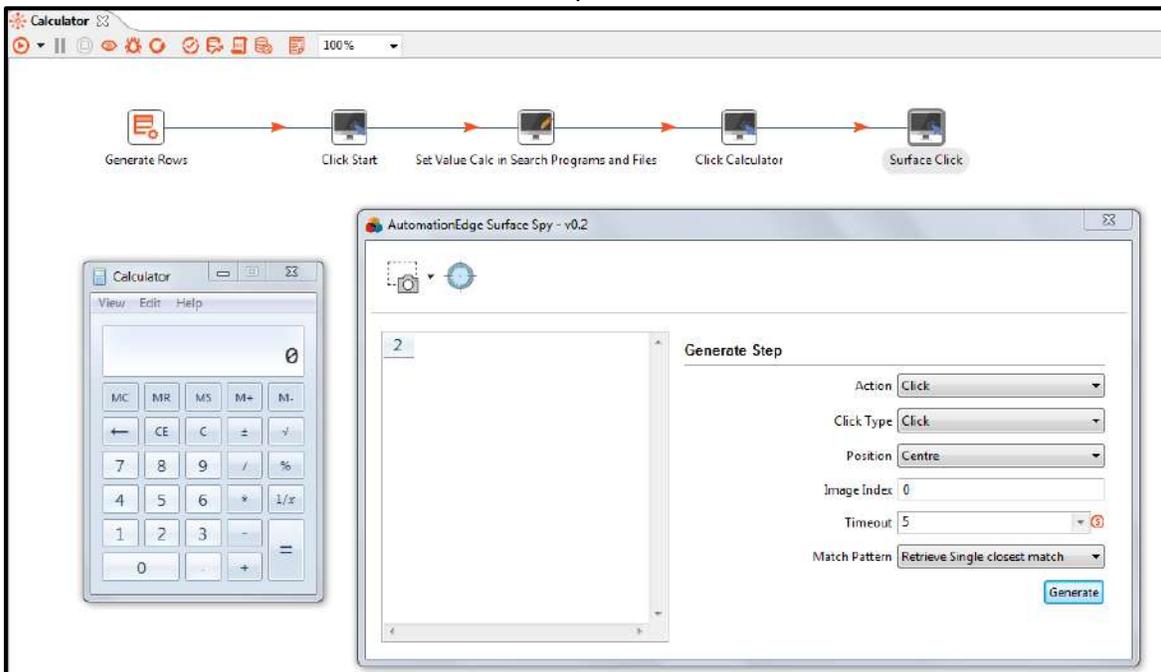


20. Rename the Surface Click step to Click Calculator.

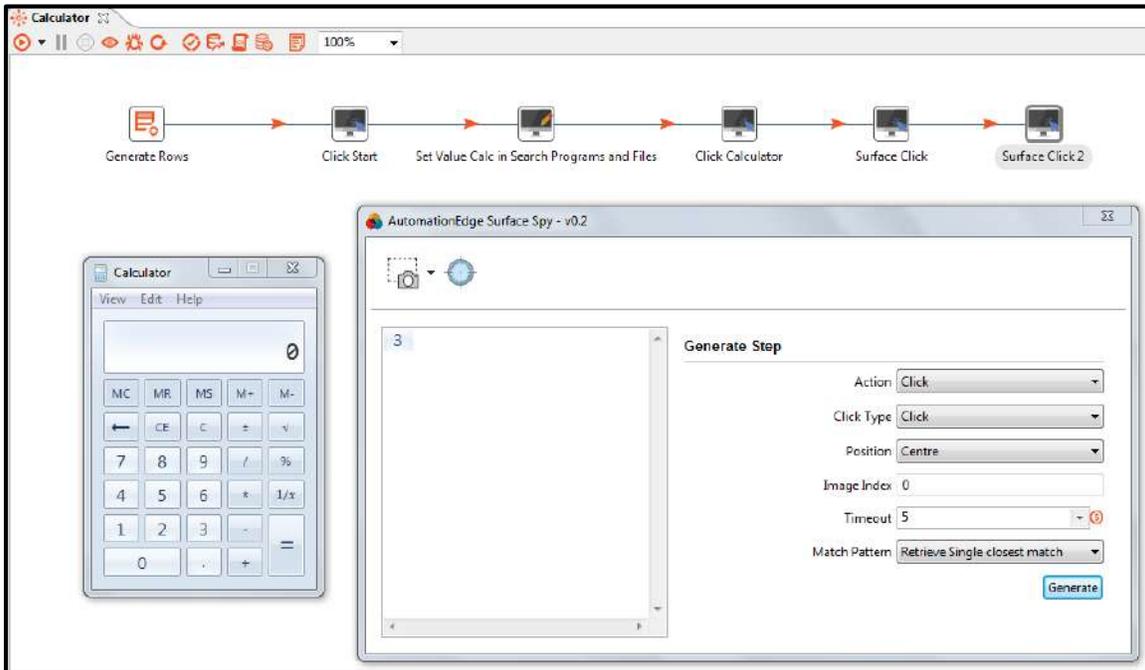


21. Capture(📷) the image for 2 on the calculator as seen below. Make sure to locate(🔍) the image and click on the exact match of the image followed by Esc key.

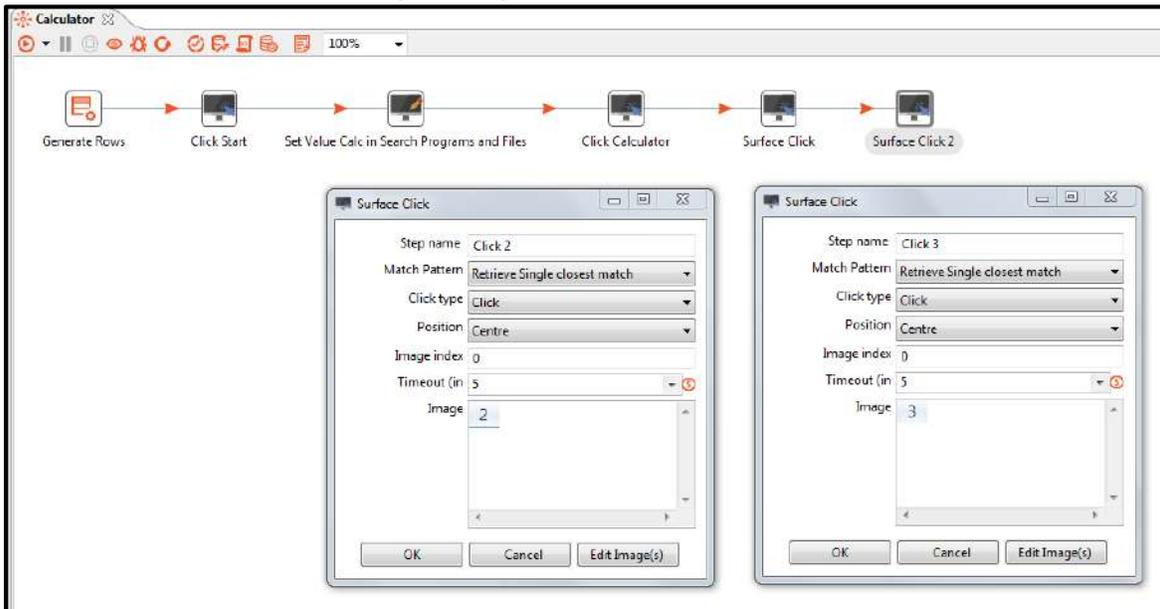
22. Click Generate to create a Surface Click Step.



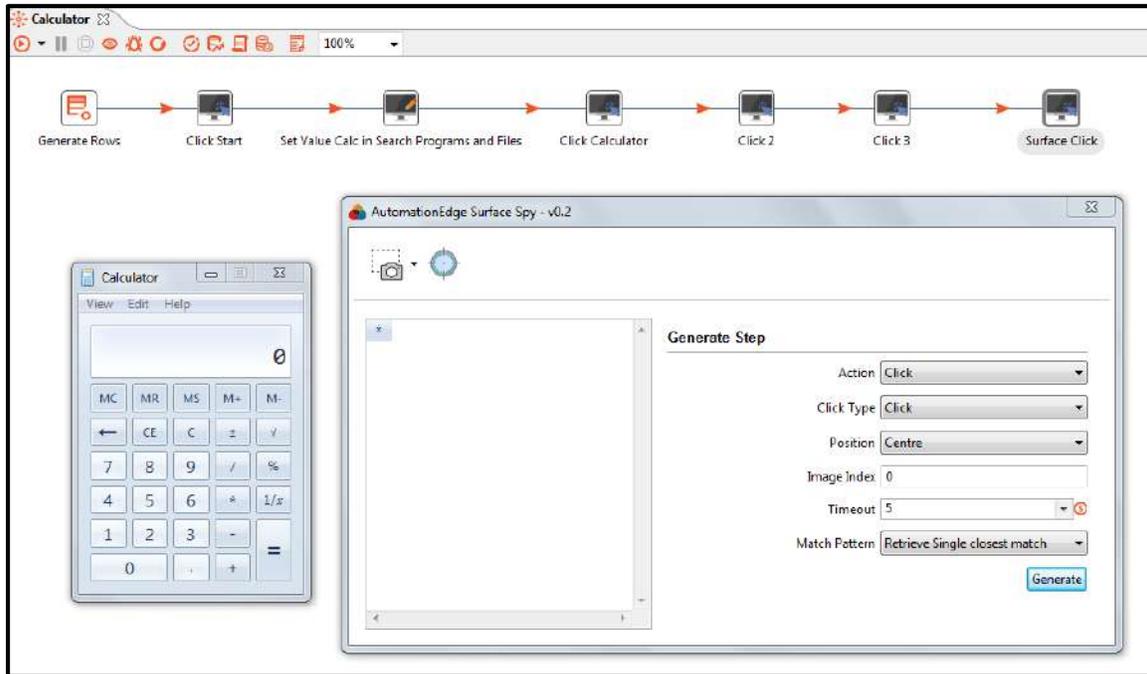
23. Capture(📷) the image for 3 on the calculator as seen below. Make sure to locate(🔍) the image and click on the exact match of the image followed by Esc key.
24. Click Generate to create a Surface Click Step.



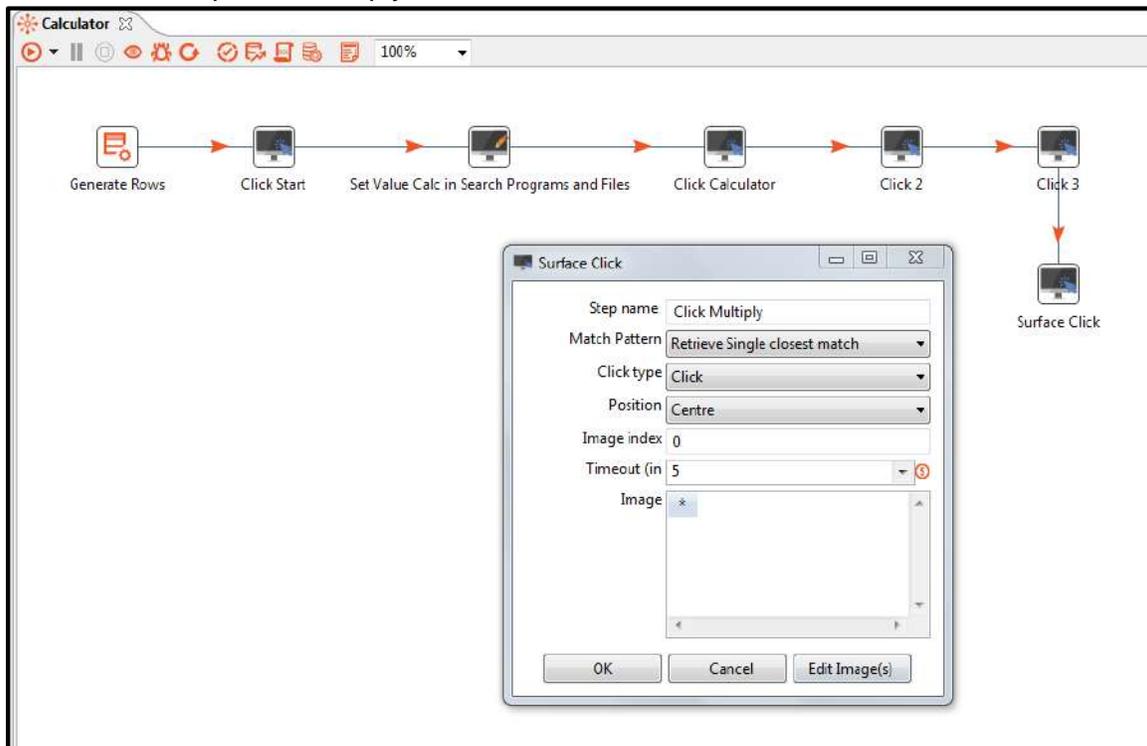
25. Double click on a step to change configuration of the step. Rename the steps as shown below.



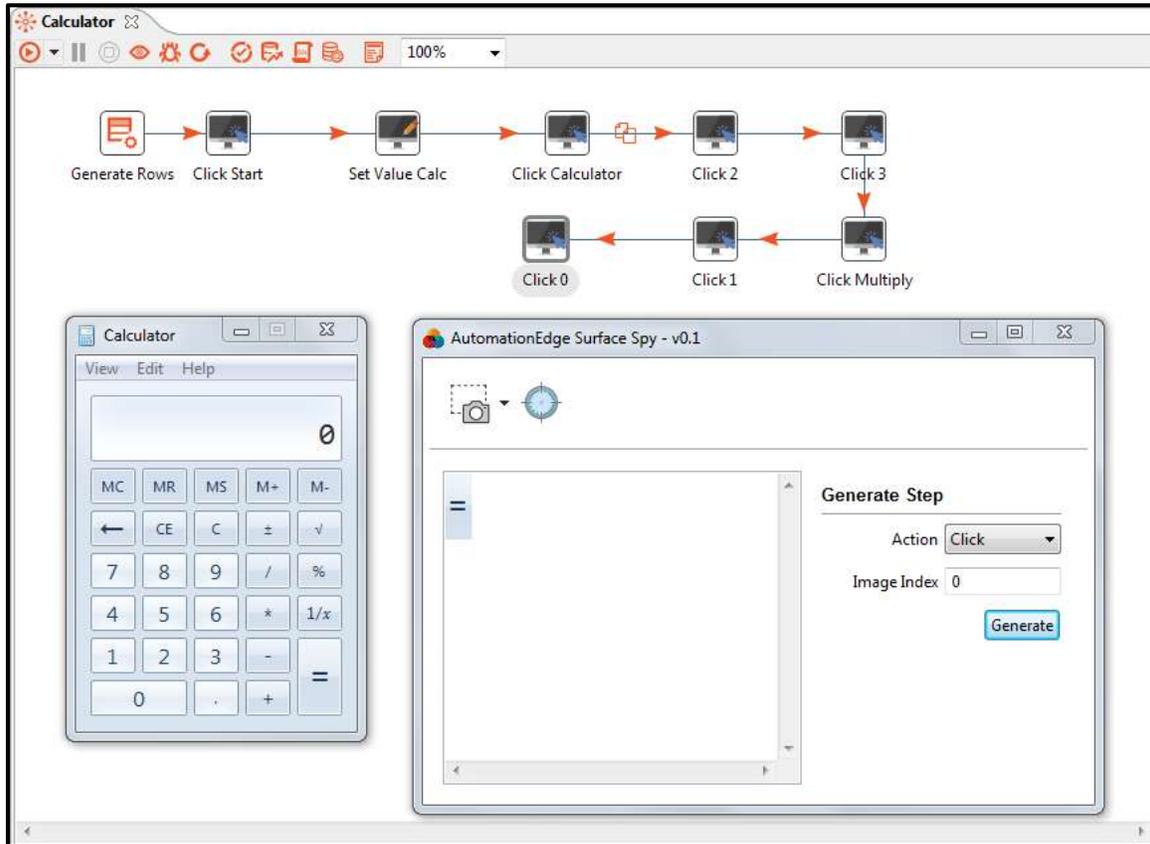
26. Capture (📷) the image for multiply(*) on the calculator as seen below. Make sure to locate (🔍) the image and click on the exact match of the image followed by Esc key.
27. Click Generate to create a Surface click step.



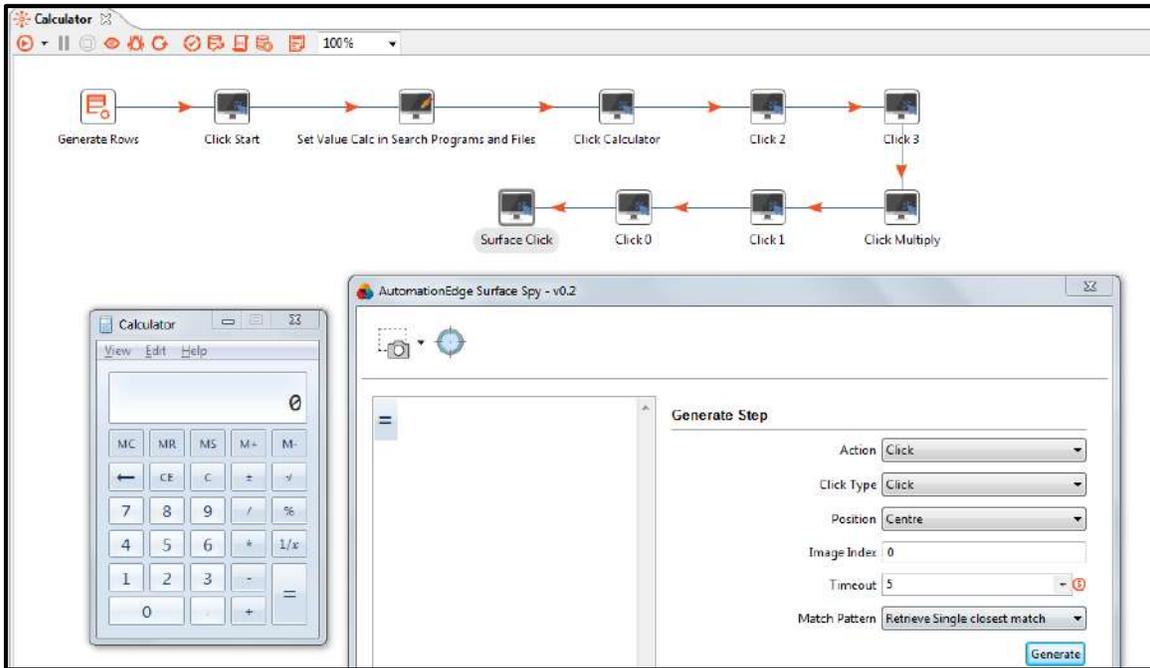
28. Rename the step Click Multiply.



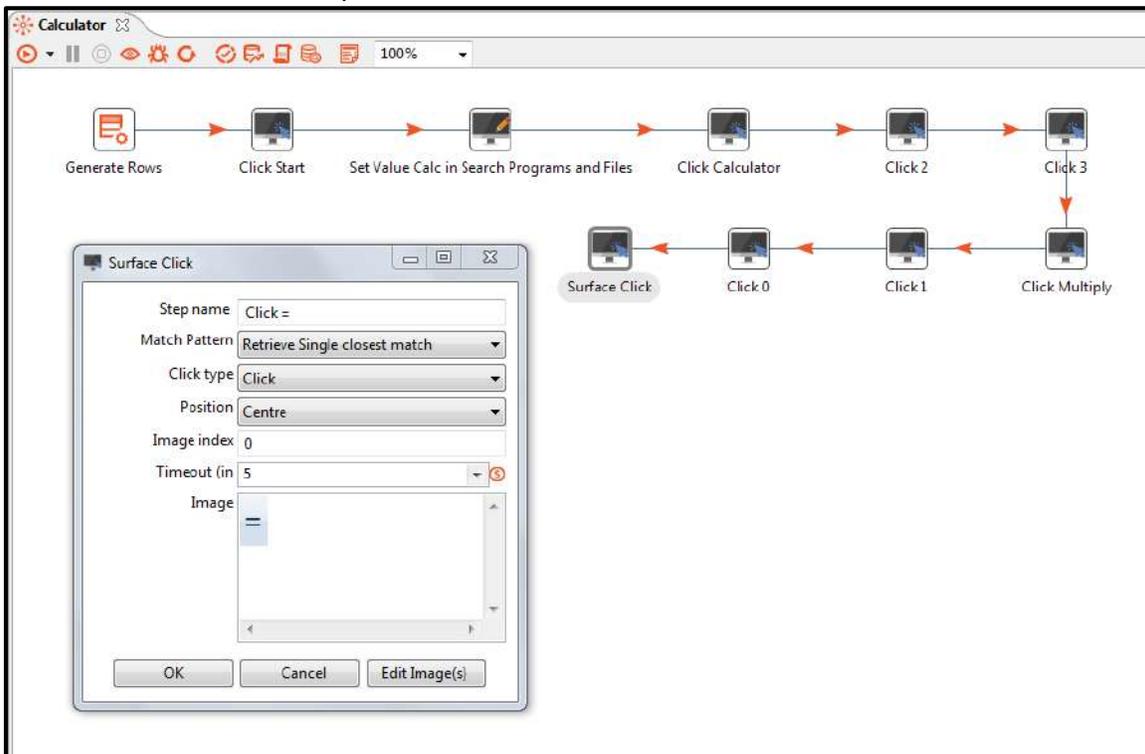
29. Next Capture(📷) the image for 1 on the calculator as seen below. In the Match Pattern select 'Retrieve multiple close matches' Make sure to locate(🔍) the image and click on the exact match of the image followed by Esc key. Generate step for clicking 1.
30. Also Capture(📷) the image for 0 on the calculator as seen below. Make sure to locate(🔍) the image and click on the exact match of the image followed by Esc key. Generate step for clicking 0.



31. Next Capture (📷) the image for = on the calculator as seen below. Make sure to locate (🔍) the image and click on the exact match of the image followed by Esc key.
32. Click Generate to create a Surface click step.



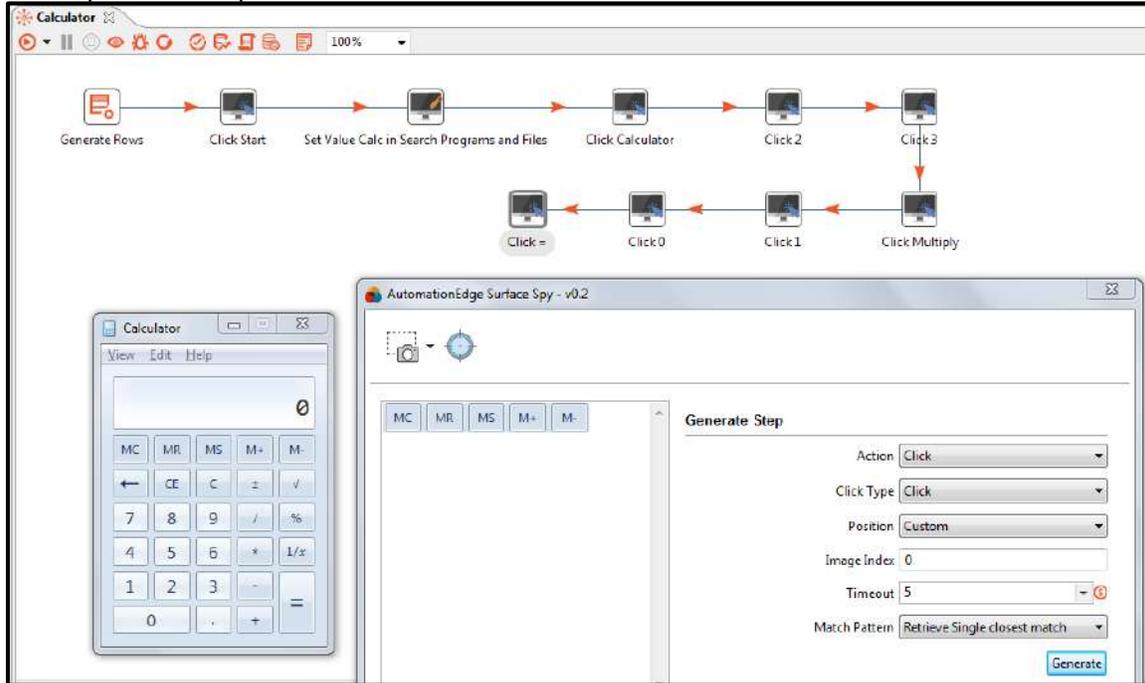
33. Rename Surface Click step to Click =



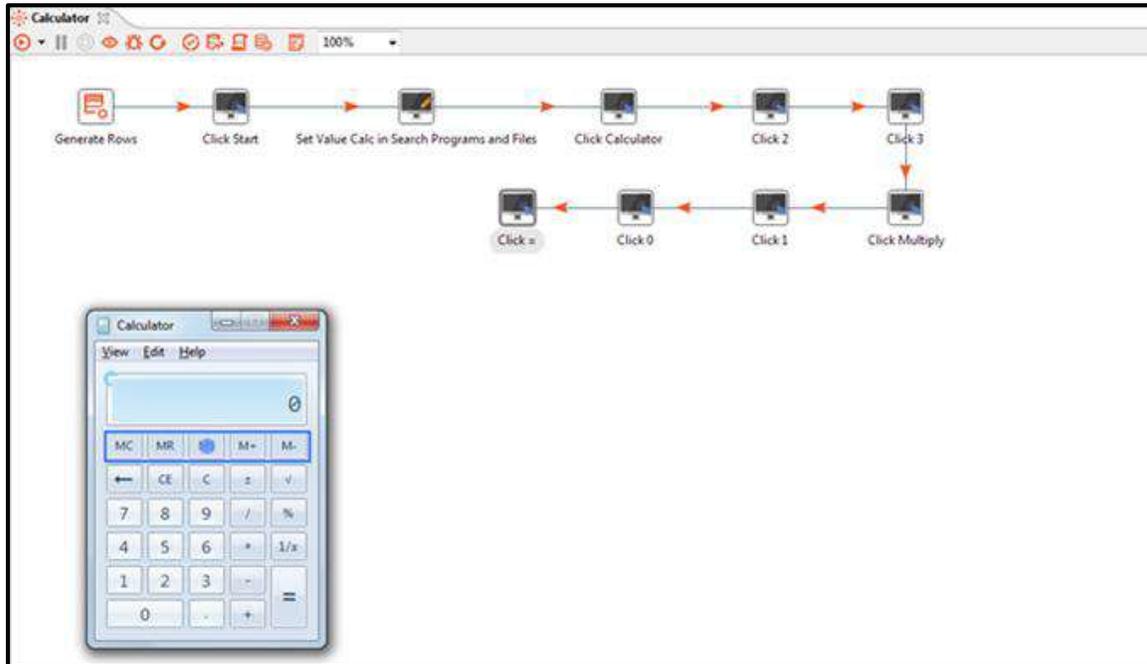
34. Next we wish to capture the calculated value of 23*10 from the Result box.

The procedure for Action Click for Positions, Left, Right and Center is the same. Now we will see how to use Custom Position.

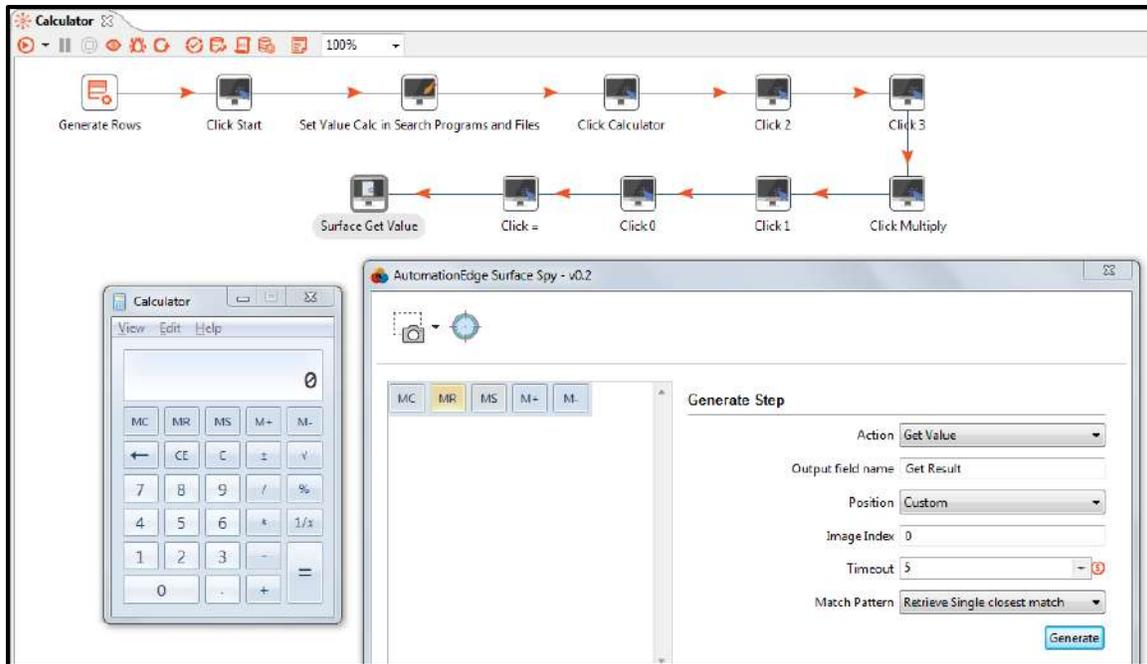
35. Let us Capture() the image for top set of Calculator buttons i.e. locate the image for the first row on the calculator containing 'MC MR MS M+ M-' as shown below. Make sure to locate() the image and click on the exact match of the image followed by Esc key. Click Generate to create a Surface click step. In the position dropdown list select Custom.



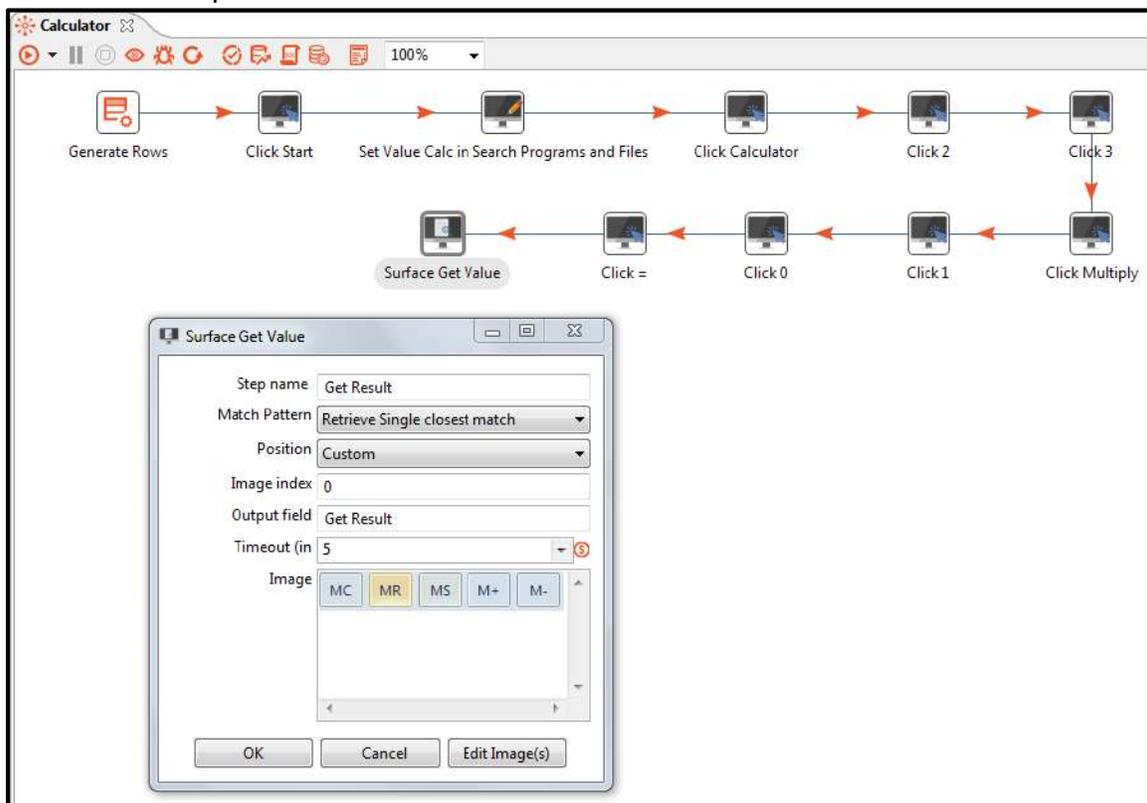
36. Click the second icon () on SurfaceSpy to locate matching images. The entire bar of top buttons on the calculator is retrieved. Click the image.
37. Next we wish to capture the calculated value of 23×10 from the Result box. First locate the image for the first row containing 'MC MR MS M+ M-' as shown below.
38. Locate the image for the first row. Click on the image. A circle appears over the first row as seen below. Click at the corner of the Result box and drag to select the box as seen below. We will Get Value from the box by identifying it as a relative position from the first row of buttons.



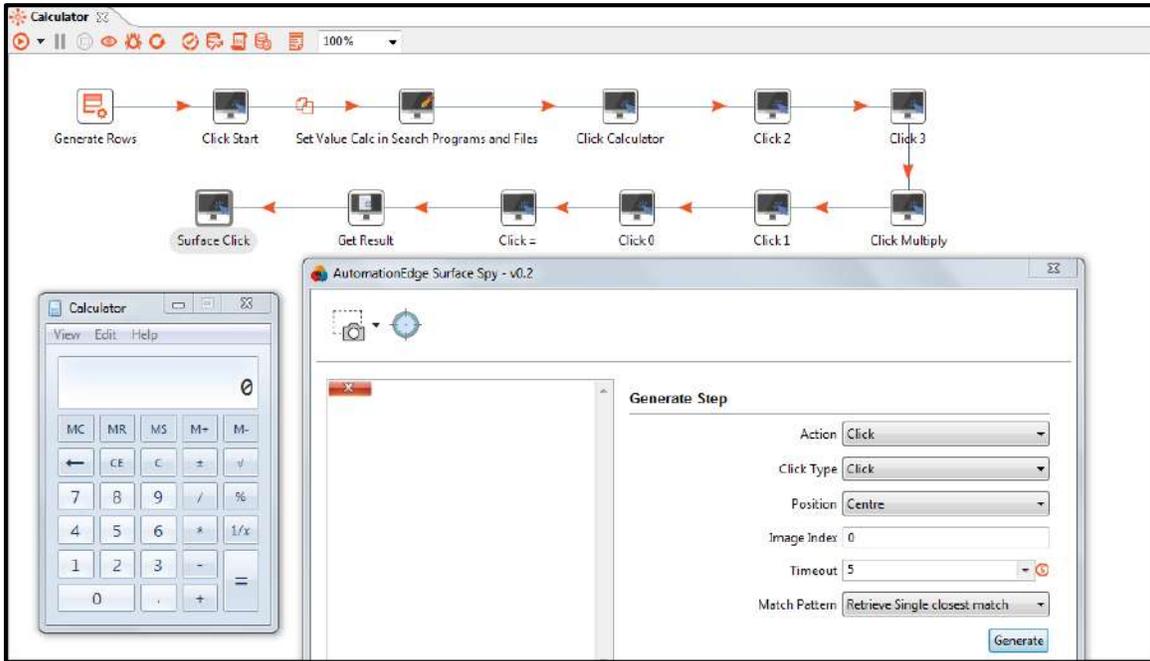
39. Bring Surface Spy to the forefront. Change the Action to Get Value. Provide an output field name say Get Result. Click Generate to create a Surface Get Value step.



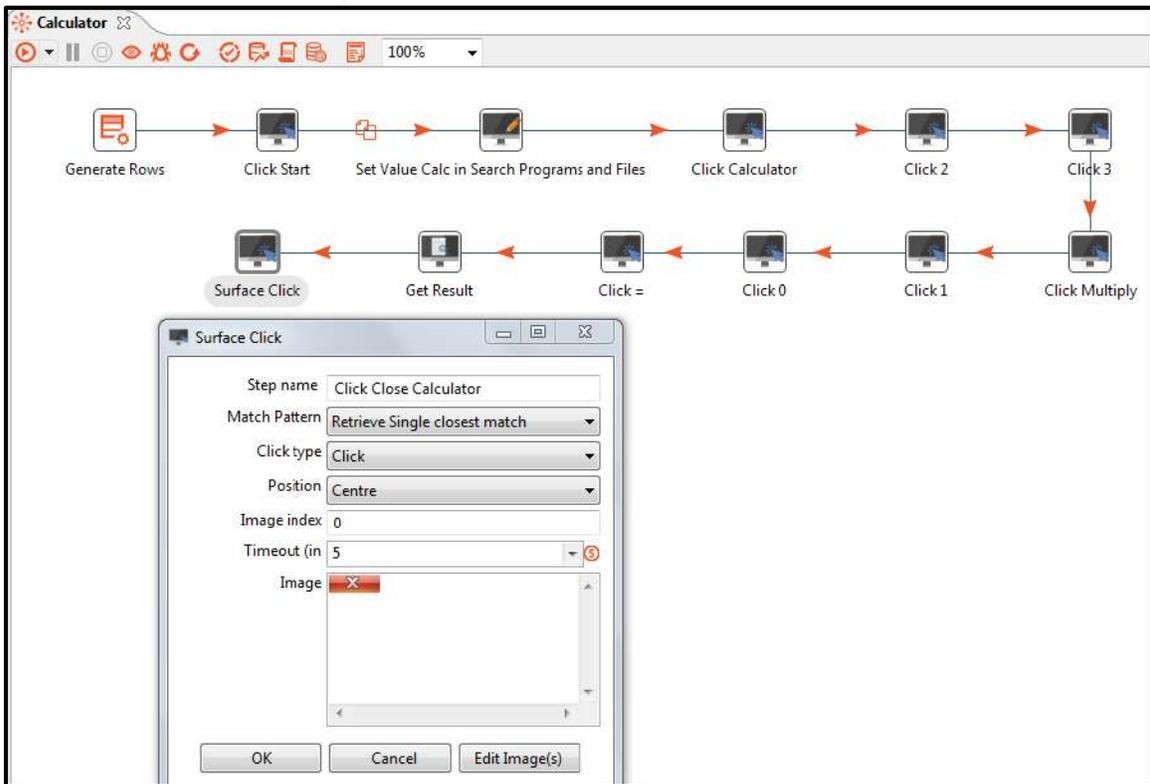
40. Rename the step to Get Result as shown below.



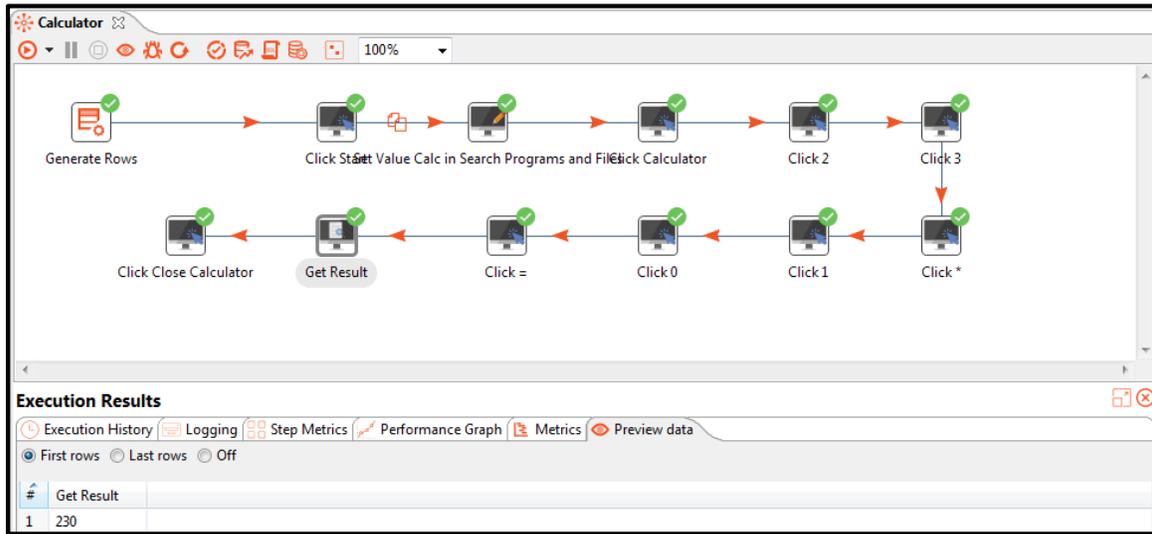
41. Now let us generate a step to close the Calculator window. Capture(📷) the image for Close (X) button on top right hand side of the calculator as shown below. Make sure to locate(🔍) the image and click on the exact match of the image followed by Esc key.



42. Rename Surface Click step to Click Close Calculator.



43. The complete workflow is shown below. It is executed to get the multiplication of two numbers as seen below.



44. This completes the process of Surface Automation to multiply two numbers.

16 Project 9: Try-Catch

Most of the steps in AutomationEdge Process Studio support error handling feature. But if a workflow contains many steps then applying error handling for each step becomes cumbersome. Thus, there is a need for functionality that can handle errors for a group of steps in a workflow. Try and Catch steps are intended for error handling in several steps.

Try and **Catch** steps are added in pairs in an AutomationEdge Process Studio workflows. Try-Catch feature is available for all the steps, except for those input steps which do not require any input to be taken from the user (e.g. Start step).

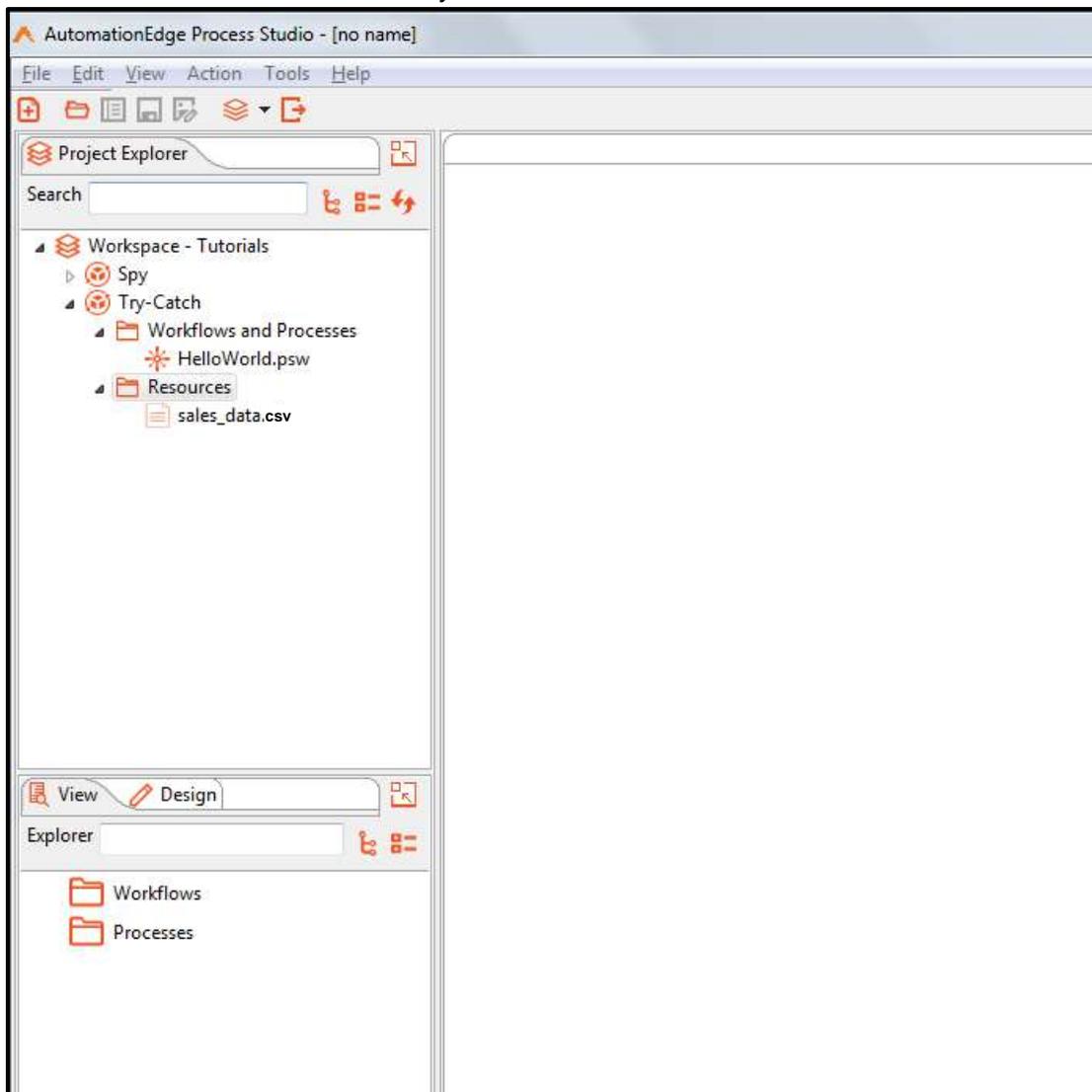
For a detailed explanations of Try-Catch steps refer [AutomationEdge_Plugin_Reference_Guide_R2.0](#).

16.1 Augmenting HelloWorld workflow with Try-Catch

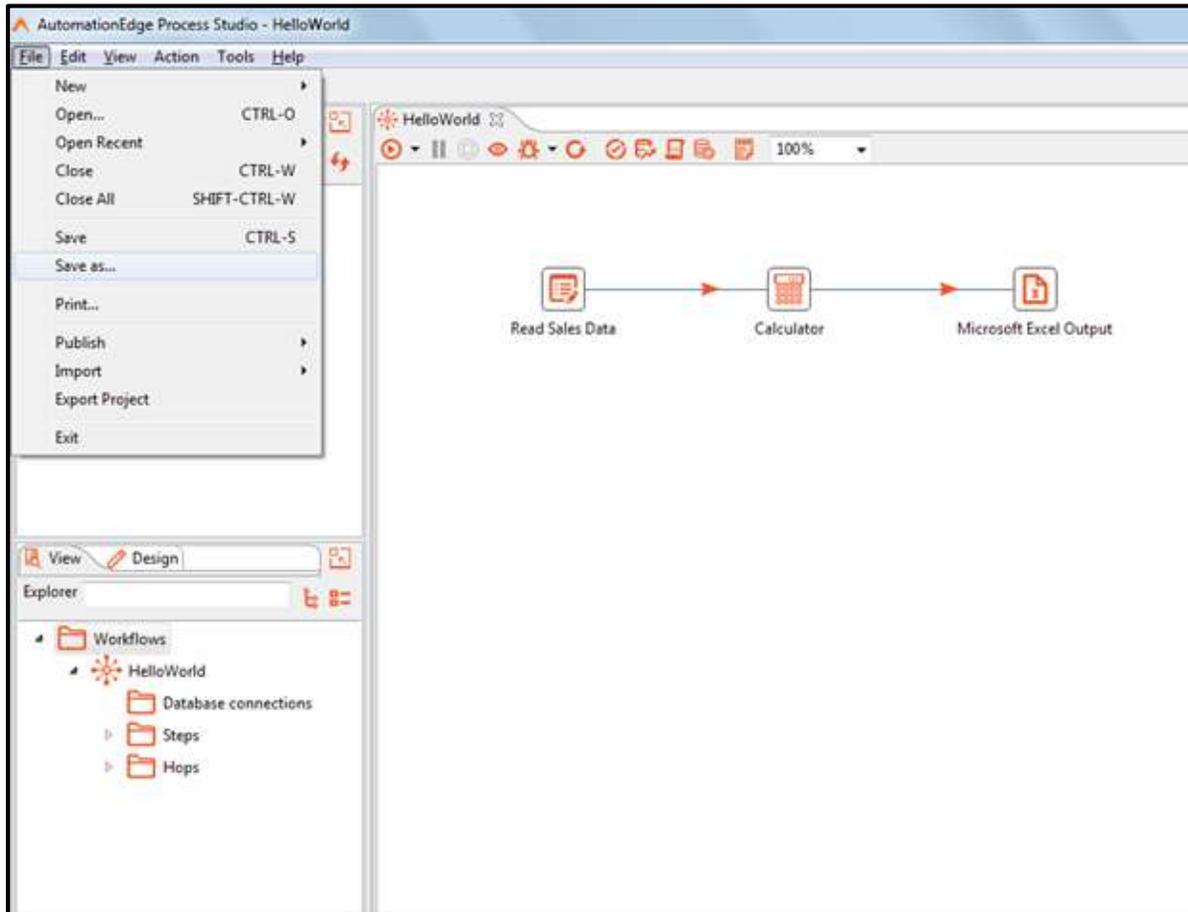
We created HelloWorld workflow in the Sales Revenue project in the Data Processing Workspace. In this section we will augment HelloWorld with Try-Catch plugin steps to handle various error scenarios. This should give a good understanding of how to implement Try-Catch. For the description of these steps refer AutomationEdge_Plugin_Reference_Guide_R2.2.

We will first copy HelloWorld from Sales Revenue project and augment it with Try-Catch steps.

1. Open Tutorials Workspace and create a project named Try-Catch as seen below.
2. Right Click on Workflow and Processes to Add File and browse for HelloWorld workflow in the **Data Processing** Workspace. Also add a Resource file sales_data.csv which is an input file for the workflow.
3. HelloWorld is now visible in the Try-Catch workflow. From the File menu select Save as..



4. Save HelloWorld as 'HelloWorld with Try-Catch' in the same project.
5. All Add the Resource file salaes_data.csv which is an input file for the workflow.

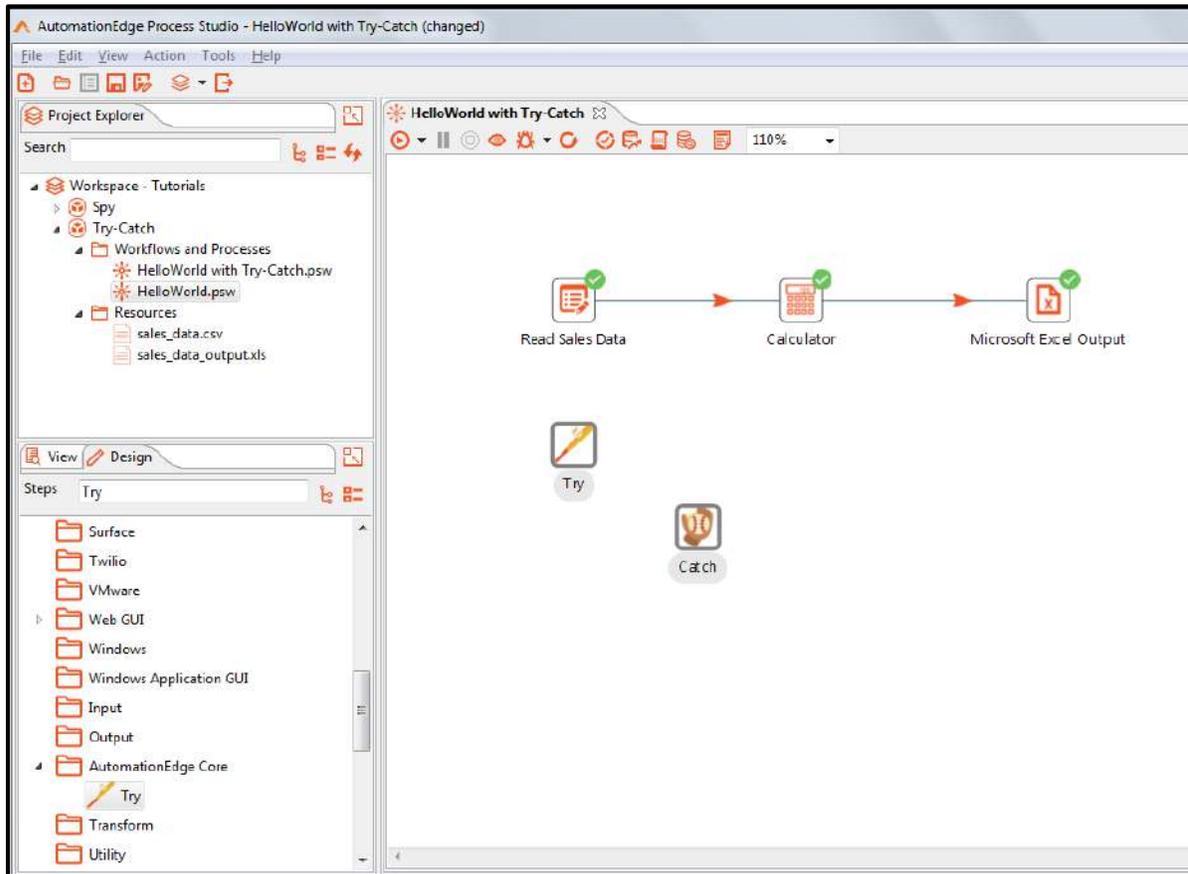


6. You may delete the HelloWorld workflow as it is no longer required. You can see 'HelloWorld with Try-Catch' below.
7. Execute the workflow to confirm its correctness. The workflow successfully executed as below.

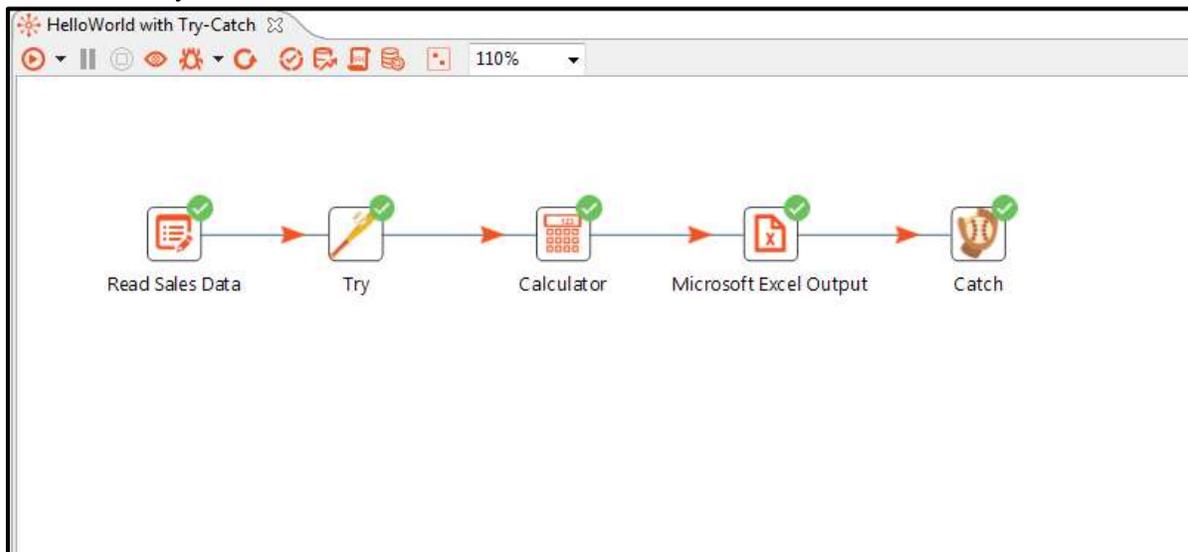
The screenshot displays the AutomationEdge Process Studio interface for a workflow named 'HelloWorld with Try-Catch'. The workflow is visualized as a sequence of three steps: 'Read Sales Data', 'Calculator', and 'Microsoft Excel Output', each marked with a green checkmark. The 'Execution Results' pane at the bottom shows a table with the following data:

#	customer	product	quantity	unitprice	totalprice
1	Thunderboltz FC	football	20	1000	20000.0
2	Google	basketball	15	900	13500.0
3	AutomationEdge	cricket bat	10	500	5000.0

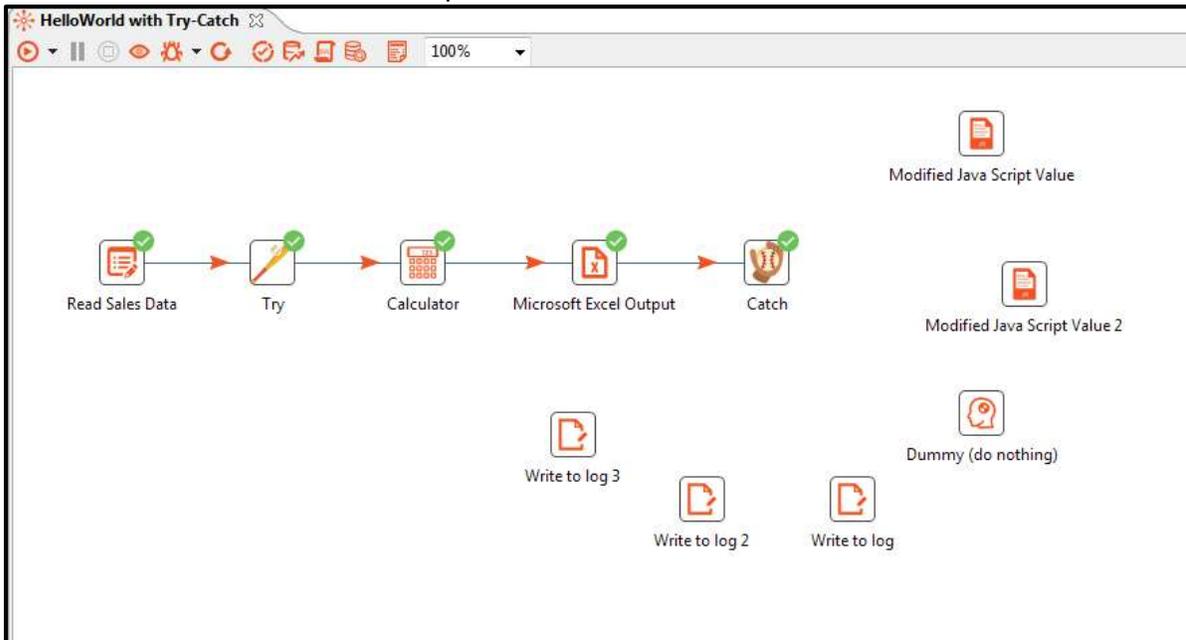
- Now type Try in the Steps field in the navigator. Locate the Try step. Drag and drop it to the canvas. Align with Try a Catch step is also dropped to the canvas automatically as Try and Catch steps are always in pairs.



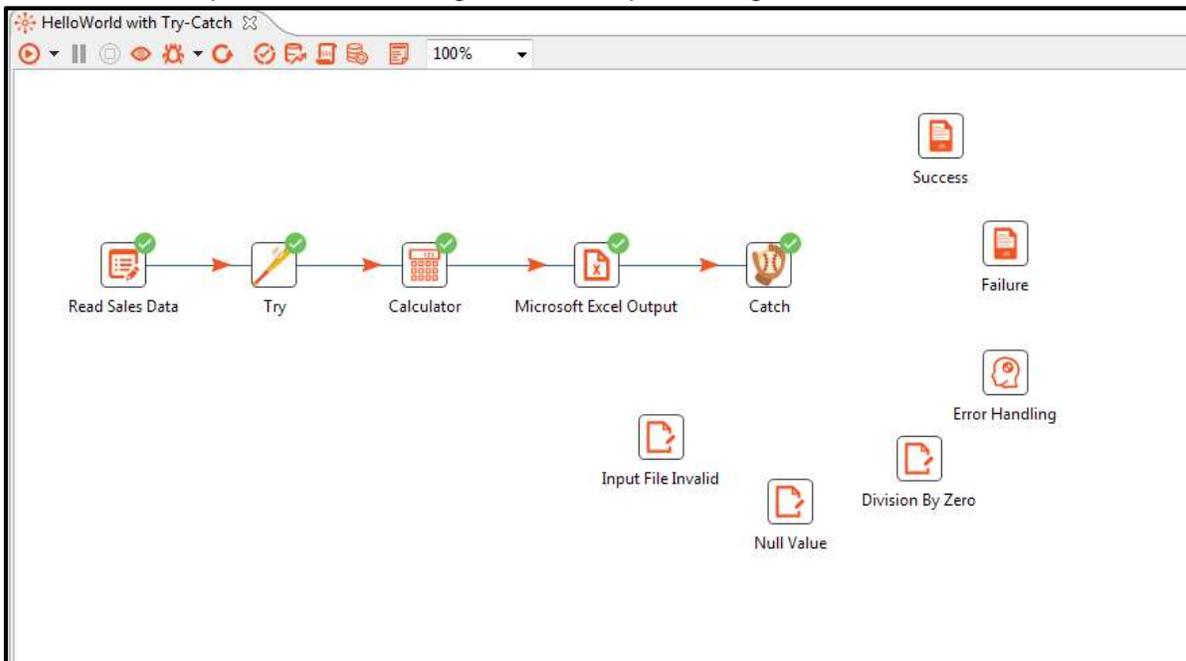
- Join the steps as shown below and execute to confirm its correctness. The Try step can also be the first step in the workflow followed by the csv input step, if you wish to have csv input step also in the Try-Catch block.



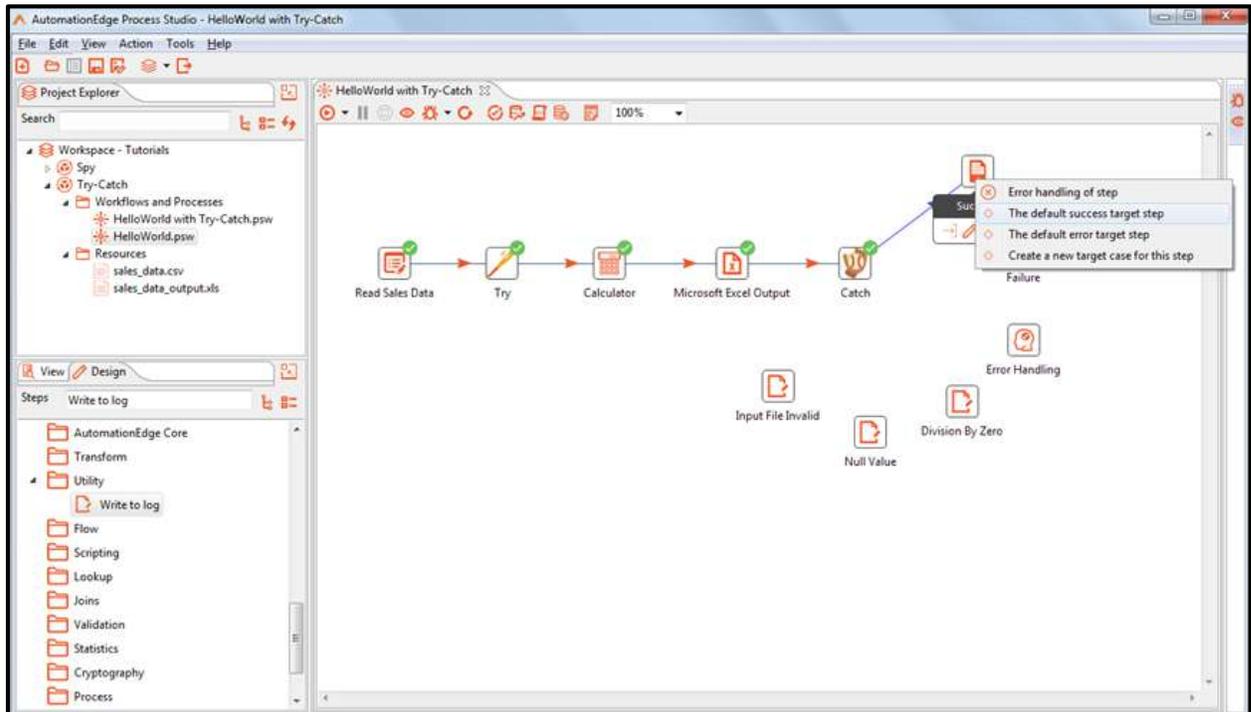
10. The screenshot shows several step added to handle different error scenarios.



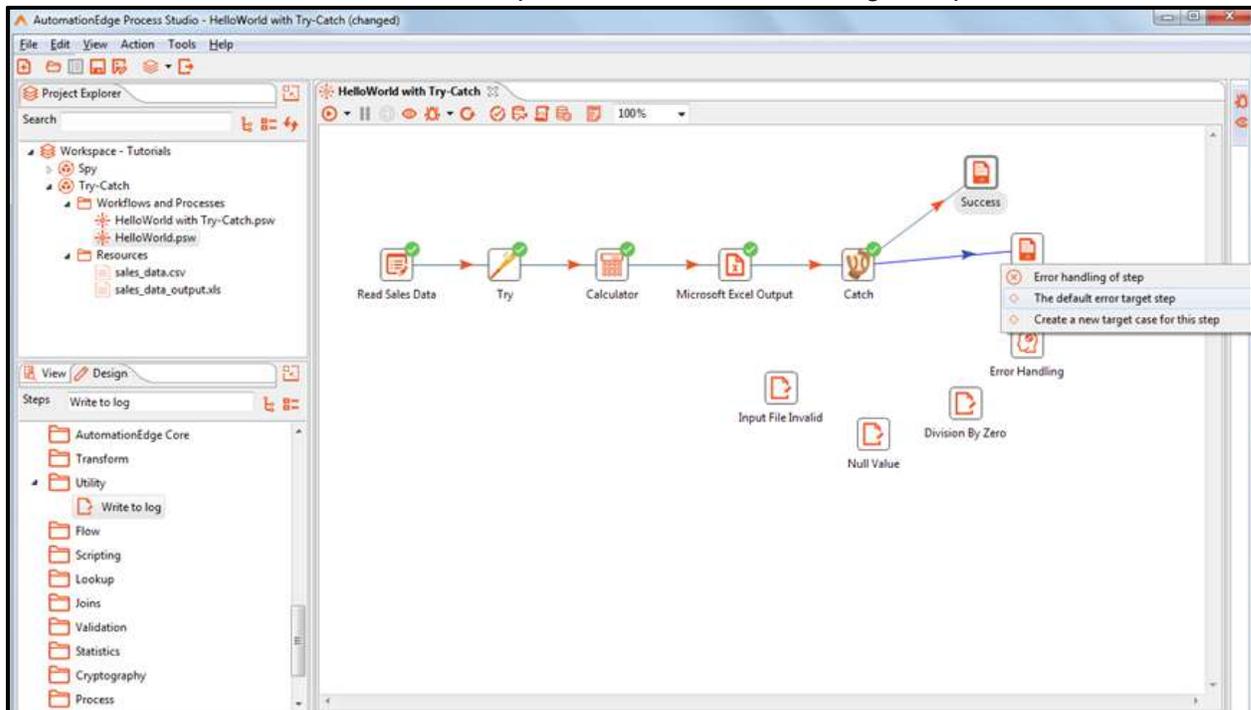
11. Rename the steps to more meaningful names representing the error as seen below.



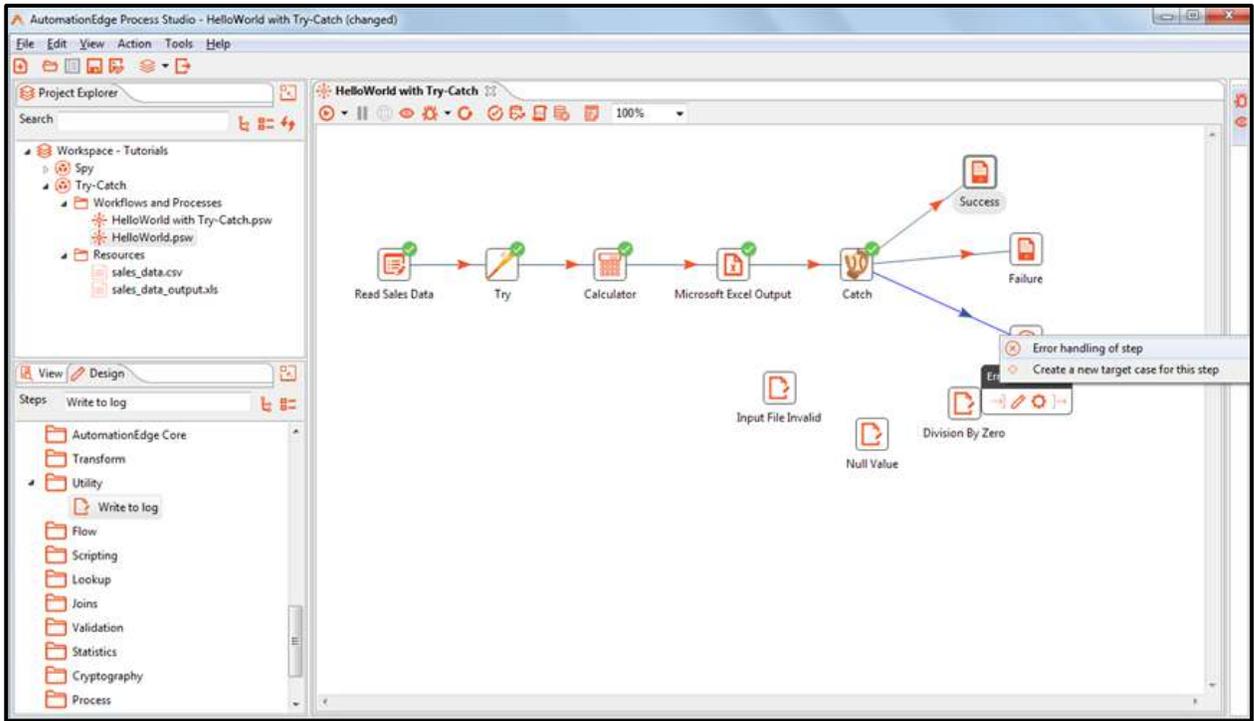
12. Draw a hop from Catch to the Success step.
13. By default, there are four types of hops from Catch step. Choose 'The default success target step' for this hop.



14. Connect Catch to the second failure step with 'The default error target step'.

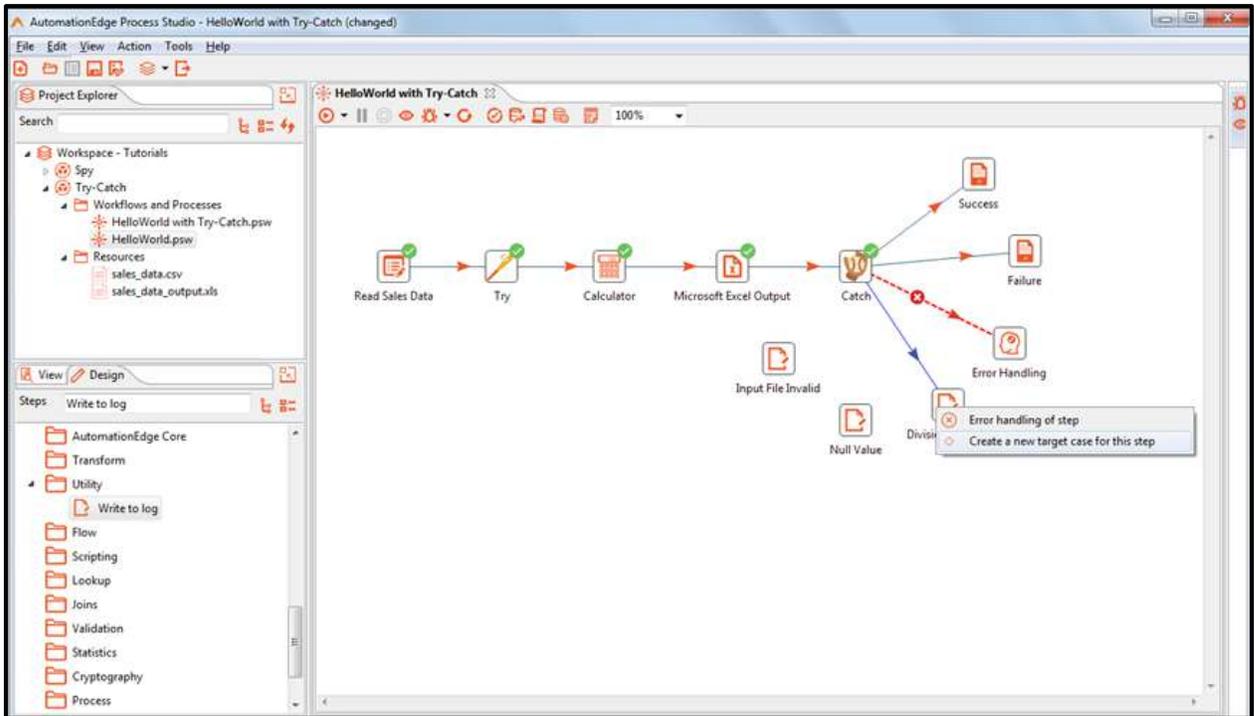


15. Connect Catch with Dummy step with 'Error handling of step'.

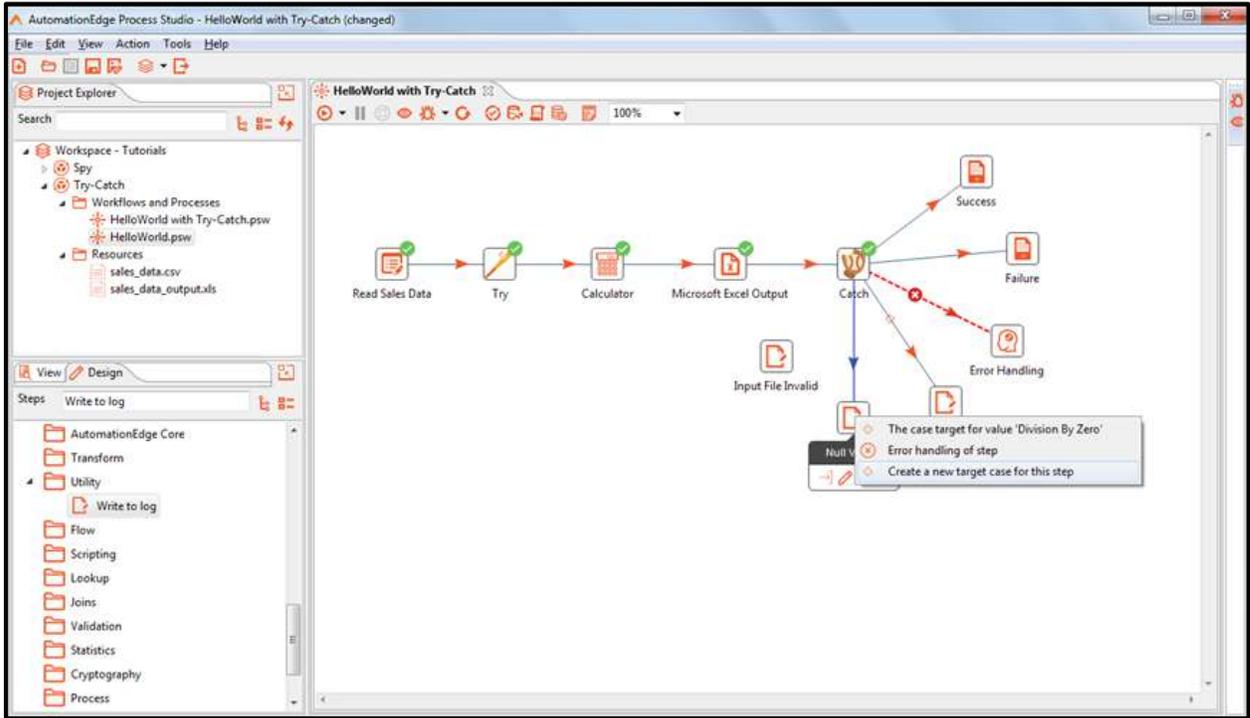


16. Connect Catch to Division by zero step with 'Create a new target case for this step'.

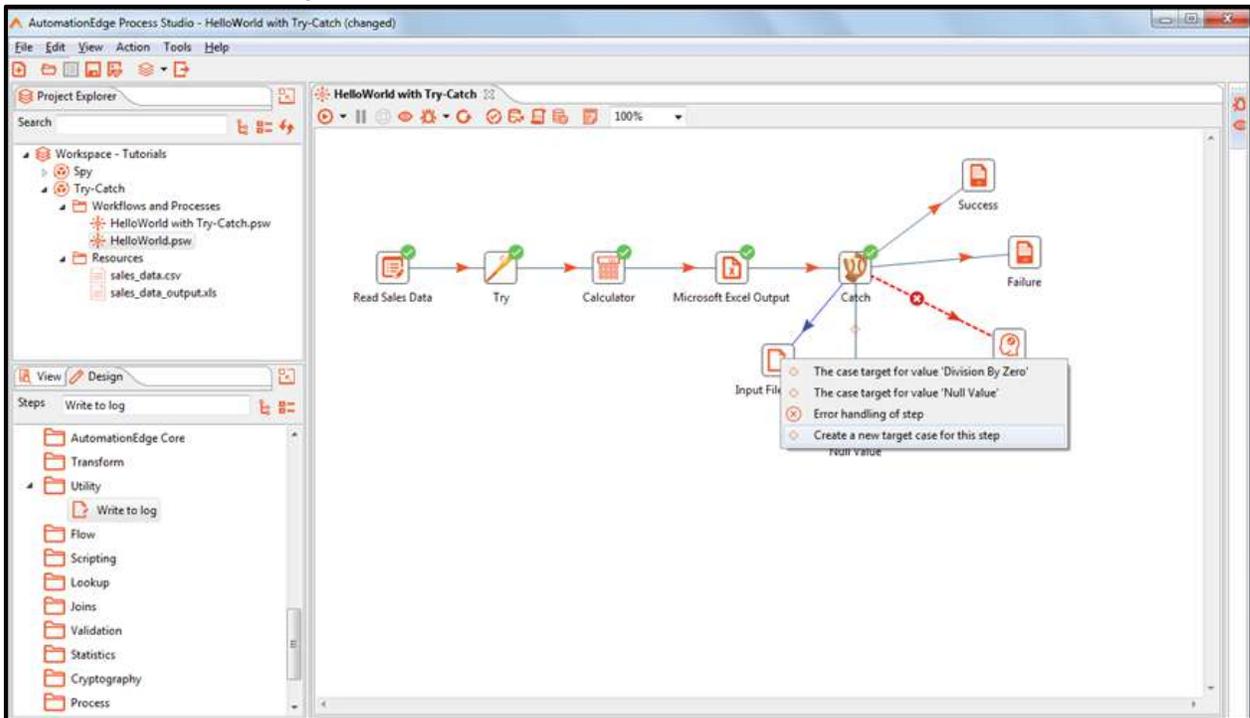
17. This creates a new Target step in the Catch step with the same description a step name.



18. Connect Catch to Null Value step with 'Create a new target case for this step'.
19. This creates a new Target step in the Catch step with the same description a step name.

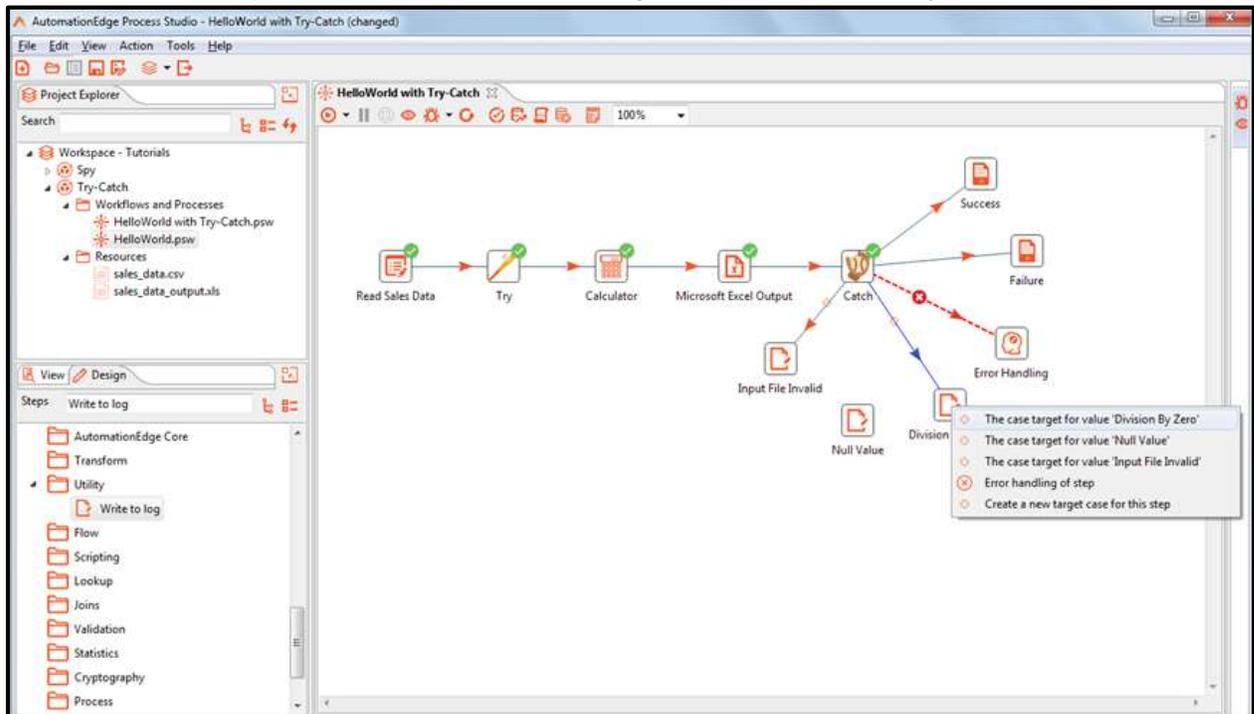


20. Connect Catch to Invalid File Input step with 'Create a new target case for this step'.
21. This creates a new Target step in the Catch step with the same description a step name.

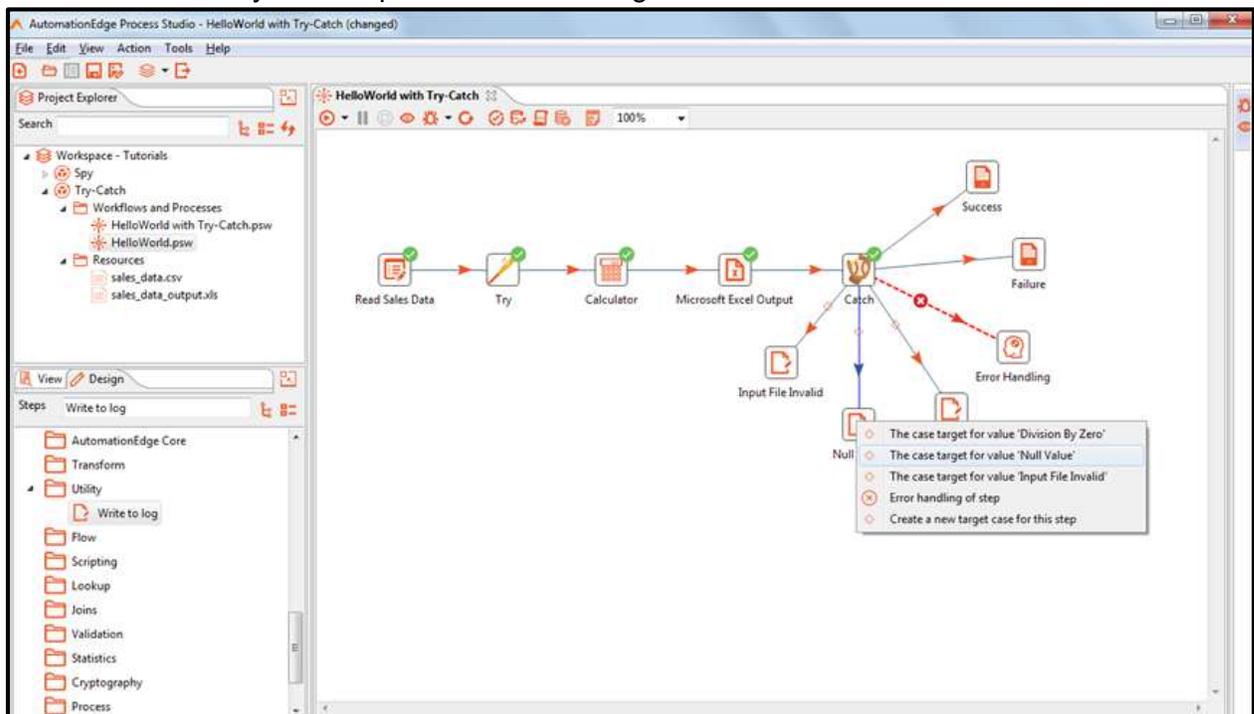


22. Now connect the missing hops with the corresponding cases.

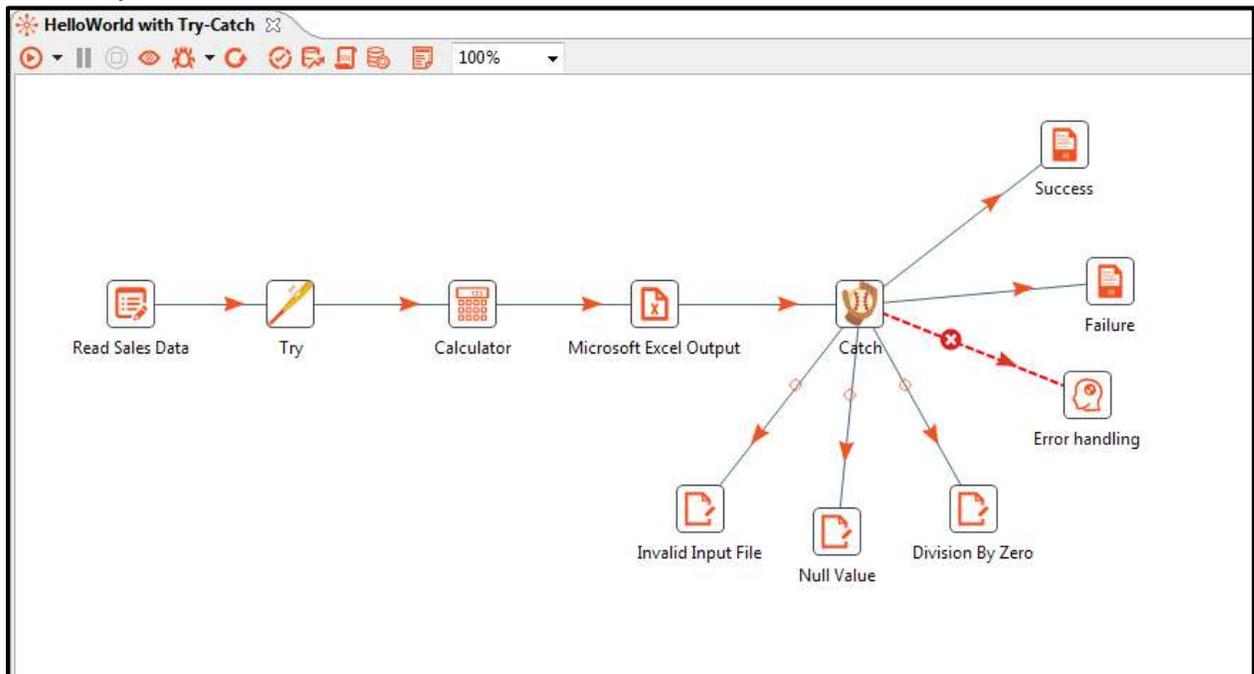
23. Connect Division By zero step with 'The case target for value Division by zero'.



24. Connect Division By zero step with 'The case target for value Null Value'.



25. The below snapshot shows the completed workflow. We have augmented HelloWorld workflow with a Try-Catch block to handle various error scenarios.



26. Double click to open Catch step properties. Check that configurations are correctly set as below.

#	Error_Description	Error_Code	Target_Step
1	Division By Zero		Division By Zero
2	Null Value		Null Value
3	Invalid Input File		Invalid Input File



Note: Default Success Target Step field is optional. Default Error Target Step field is also optional. In case the 'Default Error Target Step' field is empty, the workflow terminates at the 'Catch' step with the error which received by the 'Catch' step.

27. Go to the fields tab. Give appropriate field names in the Error Handling Fields.
28. Click Get Fields button to get fields from the input stream.

The screenshot displays the 'Catch' step configuration dialog in Process Studio. The dialog is titled 'Catch' and has a 'Step name' field set to 'Catch'. It is divided into two main sections: 'Configuration' and 'Fields'.

Configuration: This section contains 'Error Handling Fields' with the following fields:

- Number of Errors: Number of Errors
- Error Description: Error Description
- Error field: Error Fields
- Error Code: Error Code
- Error Step Name: Error Step Name

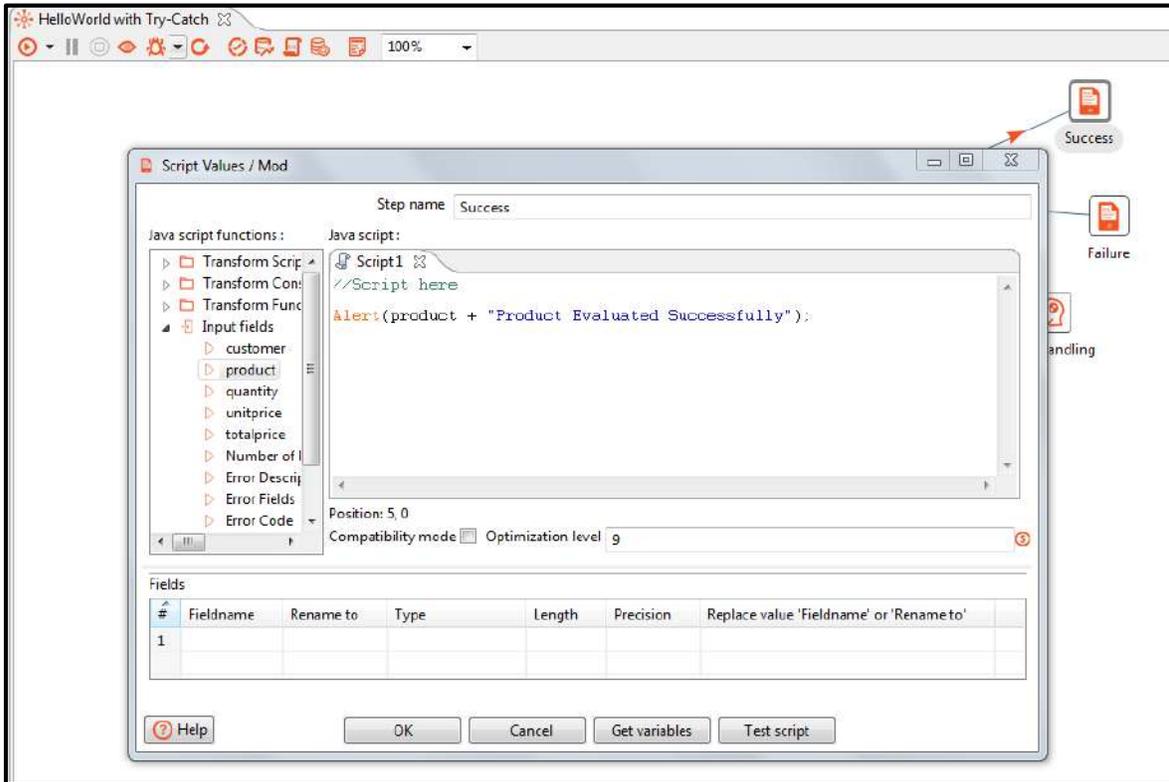
Fields: This section contains a table with the following data:

#	Field Name	Default Values
1	customer	
2	product	
3	quantity	
4	unitprice	
5	totalprice	

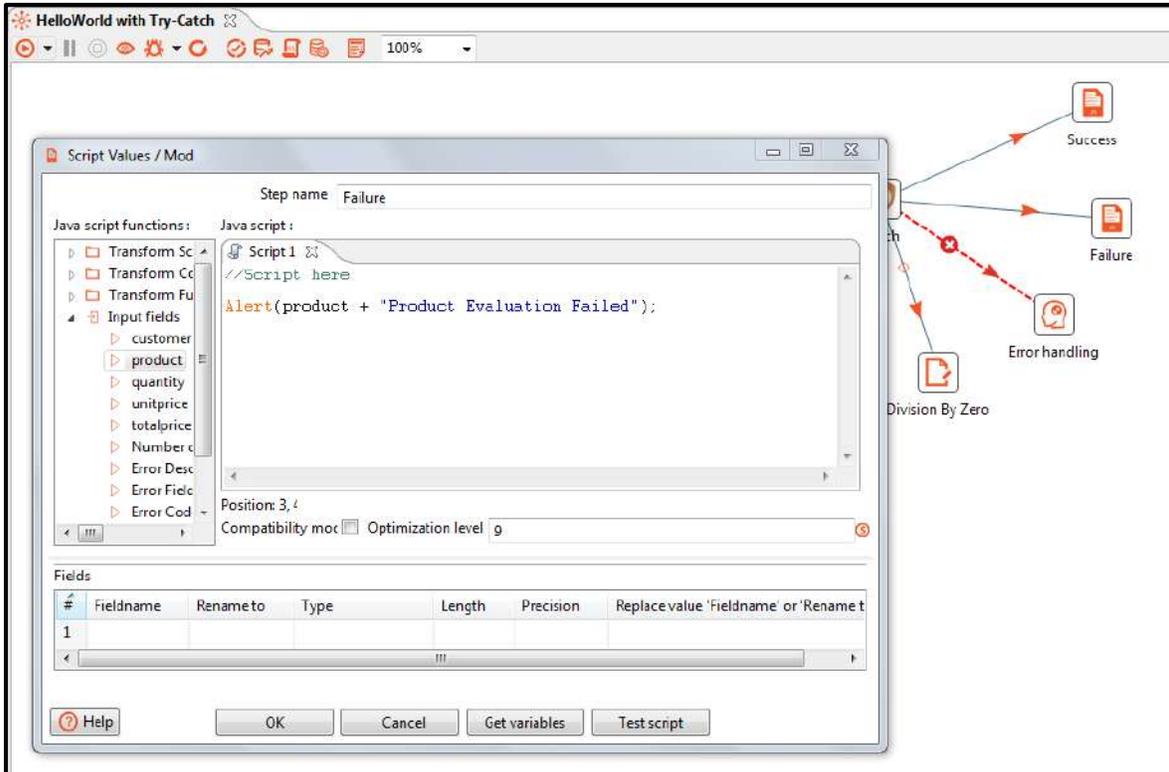
At the bottom of the dialog, there are buttons for 'Get fields', 'OK', and 'Cancel'. A 'Help' button is also present in the bottom left corner.

The background of the screenshot shows a workflow diagram. A central 'Catch' step is connected to several target steps: 'Success', 'Failure', 'Error handling', 'Null Value', and 'Division By Zero'. A red dashed arrow with a red 'X' indicates a failed connection from the 'Catch' step to the 'Error handling' step.

29. Configure the Success step to Alert the Product name for which the calculation succeeded.



30. Configure the Failure step to Alert the Product name for which the calculation failed.



31. The below screenshot shows that all the rows went to the Success step which shows all the rows data.

The screenshot displays a workflow in Process Studio titled "HelloWorld with Try-Catch". The workflow consists of the following steps: Read Sales Data, Try, Calculator, Microsoft Excel Output, and Catch. The Catch step is configured with three error handlers: Invalid Input File, Null Value, and Division By Zero. The workflow also includes a Success step, a Failure step, and an Error handling step. The execution results table below shows that all three rows of data were successfully processed and sent to the Success step.

#	customer	product	quantity	unitprice	totalprice	Number of Errors	Error Description	Error Fields	Error Code	Error Step Name
1	Thunderboltz FC	football	20	1000	20000.0	<null>	<null>	<null>	<null>	<null>
2	Google	basketball	15	900	13500.0	<null>	<null>	<null>	<null>	<null>
3	AutomationEdge	cricket bat	10	500	5000.0	<null>	<null>	<null>	<null>	<null>

32. This completes the process of configuring Try-Catch steps.

In the following sections we shall see more scenarios of Try-Catch steps with various Plugin Steps and workflow designs.

16.2 Try-Catch with Loop Steps

1. During the execution of the workflow, if an error occurs in any of the intermediate steps which reside in-between Try-Catch steps pair, then the error goes to the 'Catch' step.
2. Likewise, exception can occur in any loop (including start loop, break loop or continue loop steps) and handled in catch. However, Try - Catch steps can be used in combination with Start Loop - Continue Loop steps only in the following ways,
 - Start Loop - Continue Loop steps should be completely designed in between Try - Catch steps.
 - Try - Catch steps should be completely designed in between Start Loop - Continue Loop steps.
 - All the hops from Catch in a loop (in true path of loop) should be connected to Continue (while maintaining the loop validations)
 - ✓ Backward hop from Catch is not allowed.

16.2.1 Try-Catch with Simple Loop

1. The following screenshot depicts a simple loop in a Try-Catch block. As expected Continue Loop is in the true path of Start Loop and Catch is in the false path.

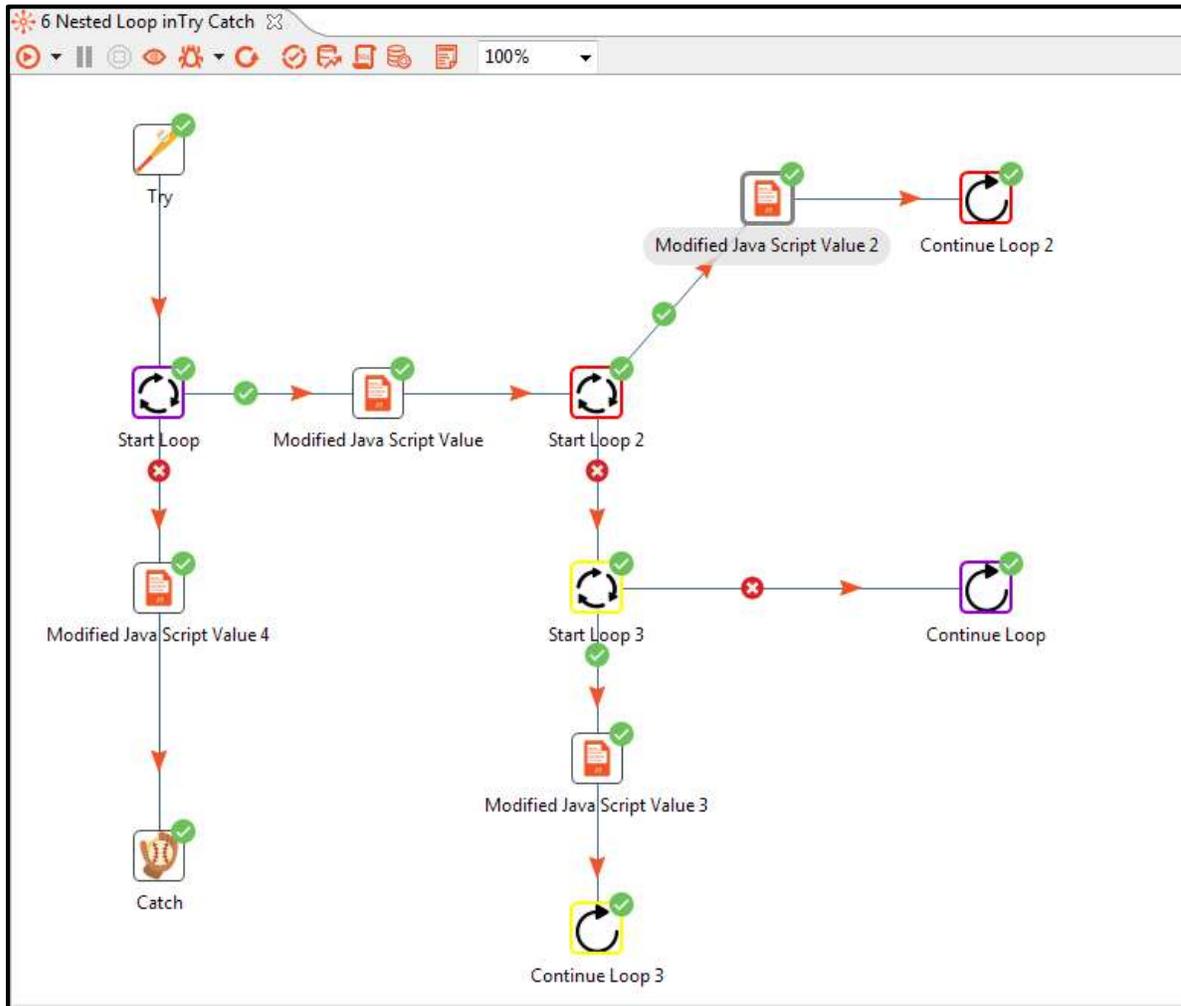
The screenshot displays a workflow diagram titled "2 Simple Loop in Try-Catch". The workflow starts with a "Generate Rows" step, followed by a "Try" step. Inside the Try block, there are three steps: "Modified Java Script Value", "Start Loop", and "Catch". The "Start Loop" step is connected to a "Continue Loop" step, which loops back to the "Start Loop" step. The "Catch" step is connected to two "Write to log" steps. The execution results show the following sequence of events:

```

2021/07/03 16:40:43 - Write to log 2.0 - =====
2021/07/03 16:40:44 - Try.0 - Finished processing (I=0, O=0, R=1, W=1, U=0, E=0)
2021/07/03 16:40:44 - Modified Java Script Value.0 - Finished processing (I=0, O=0, R=1, W=1, U=0, E=0)
2021/07/03 16:40:44 - Start Loop.0 - Finished processing (I=0, O=0, R=3, W=2, U=0, E=1)
2021/07/03 16:40:44 - Modified Java Script Value 2.0 - Finished processing (I=0, O=0, R=2, W=1, U=0, E=1)
2021/07/03 16:40:44 - Catch.0 - Finished processing (I=0, O=0, R=1, W=1, U=0, E=0)
2021/07/03 16:40:44 - Continue Loop.0 - Finished processing (I=0, O=0, R=2, W=2, U=0, E=0)
2021/07/03 16:40:45 - Write to log 2.0 - Finished processing (I=0, O=0, R=1, W=1, U=0, E=0)
2021/07/03 16:40:45 - 2 Simple Loop in Try-Catch - Workflow Result [ 2 Simple Loop in Try-Catch ] : Time (in millisecs) : 13 , execution units : 0 , max number of executions : 3
2021/07/03 16:40:45 - AutomationEdge Process Studio - The workflow has finished!!
  
```

16.2.2 Try-Catch with Nested Loop

1. In the following screenshot a Nested Loop is within a Try-Catch block. The Nested Loop is contained within the Try-Catch block.



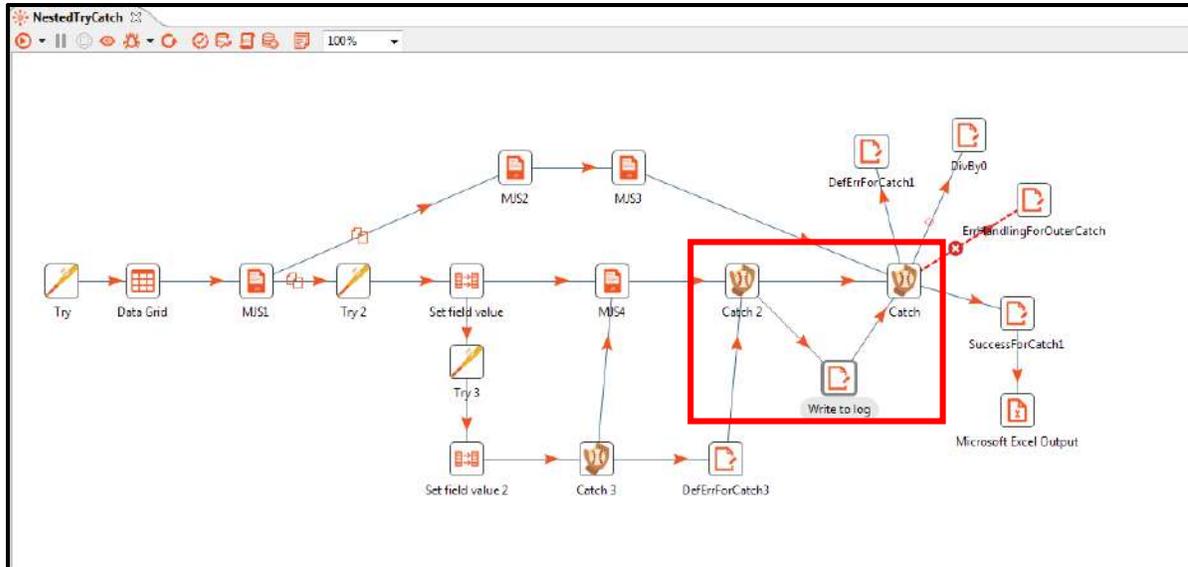
16.3 Nested Try Catch

1. In the snapshot below there are two Try-Catch blocks within a loop.
2. In the snapshot below as expected Continue Loop is in the true path of Start Loop and both Catch steps are also in the true path followed by Continue loop.
3. During execution of the workflow, if any error occurs in the inner Catch step then it is sent to the outer Catch step

The screenshot displays a workflow diagram titled "7 Nested Try-Catch in Loop". The workflow starts with "Generate Rows", followed by a "Start Loop" step. Inside the loop, there are two nested try-catch blocks. The first try block contains "Try 2 Modified Java Script Value 3" and "Try". The second try block contains "Modified Java Script Value" and "Catch". The "Catch" step is connected to "Catch 2". The loop ends with a "Continue Loop" step. The execution results table below shows the following data:

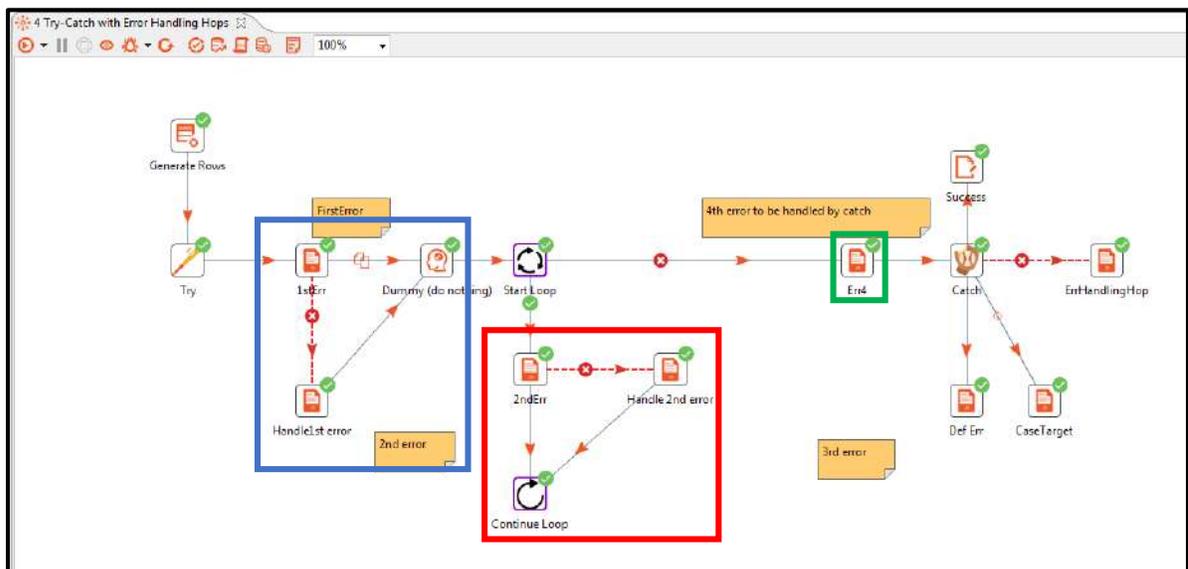
#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
1	Generate Rows	0	0	11	0	0	0	0	0	Finished	0.0s	1,833	-
2	Start Loop	0	22	22	0	0	0	0	0	Finished	2.8s	8	-
3	Modified Java Script Value 4	0	11	11	0	0	0	0	0	Finished	3.0s	4	-
4	Try 2	0	11	11	0	0	0	0	0	Finished	3.1s	4	-
5	Modified Java Script Value 3	0	11	11	0	0	0	0	0	Finished	3.3s	3	-
6	Try	0	6	6	0	0	0	0	0	Finished	3.5s	2	-
7	Modified Java Script Value	0	6	6	0	0	0	0	0	Finished	3.7s	2	-
8	Modified Java Script Value 7	0	5	5	0	0	0	0	0	Finished	3.5s	1	-
9	Modified Java Script Value 2	0	3	3	0	0	0	0	0	Finished	3.9s	1	-
1.	Catch	0	6	6	0	0	0	0	0	Finished	4.5s	1	-
1.	Catch 2	0	11	11	0	0	0	0	0	Finished	4.8s	2	-

- Any split hop from inner catch needs to be joined to the parent catch or any step leading to the Catch. As seen in the snapshot below and highlighted in red, a split hop from Catch 2 (i.e. Write to log) is connected to the parent Catch step; or it could also have been connected to a step before Catch.



16.4 Error Handling Hop

- Intermediate steps between Try-Catch cannot be connected by error handling hops. A workaround for this is you can have a split hop to an error handling step.
 - However, the error handling step should connect back to next step in the Try-Catch block. This is seen in the section highlighted in **blue** in the screenshot below.
 - In case of Loops in the Try-Catch block the error handling step should connect to the Continue step. This is seen in the section highlighted in **red** in the screenshot below.
- Steps in a Try-catch block can have error handling steps to handle the errors. During execution of the workflow, if any error occurs in an intermediate step and is handled by an error handling step; the error is sent to the Catch step
- For the steps that do not have error handling steps the error is handled by the Catch step. In the snapshot below the step Err4 highlighted in **green** does not have error handling, hence errors are handled in the Catch step.



In this section we have seen how to use Try-Catch steps. We have also seen several workflow design scenarios and how to design Try-Catch steps in these scenarios.

17 Project 10: Debugging

All the artefacts of the Debugging feature in Process Studio were discussed in the section [Error! Reference source not found.](#)

In this section we will create a workflow with debugging features, provide a step-by-step demonstration of creating breakpoints in a workflow and also demonstrate workflow execution in all the Debugging modes.

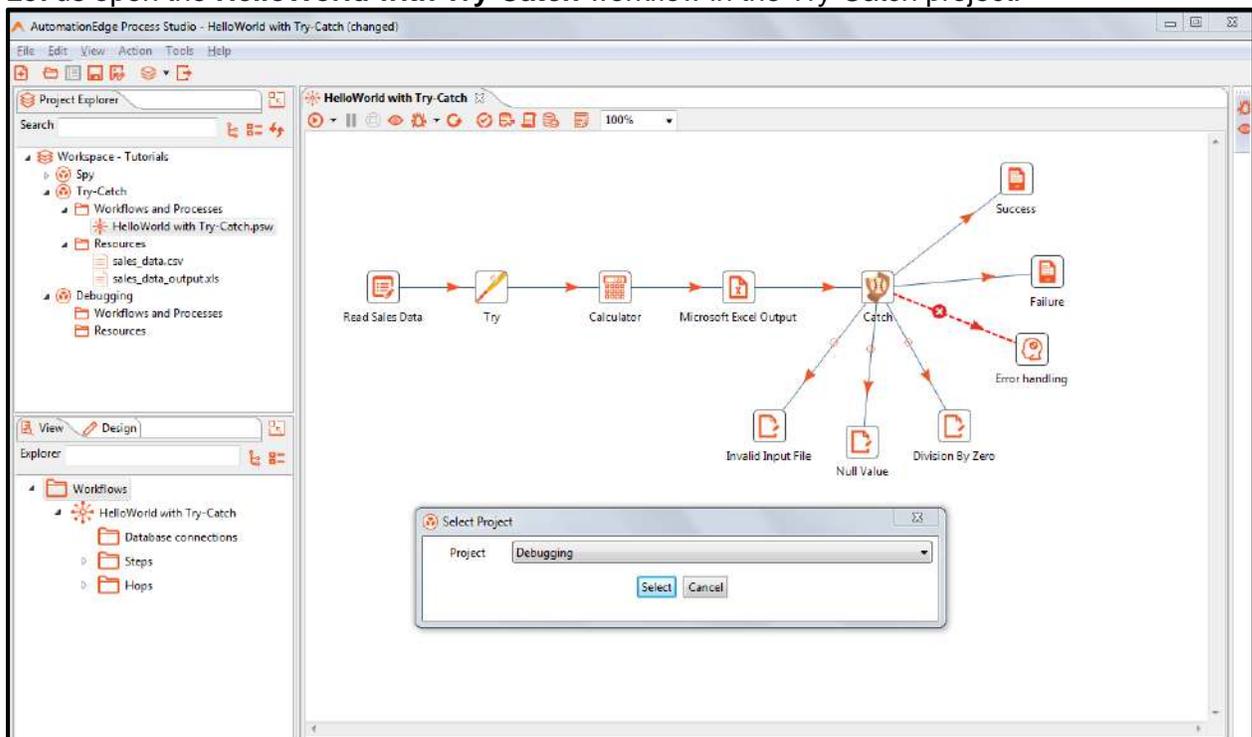
Let us begin by creating a project for our debugging workflow.

1. Launch Process Studio and open a workspace (in this e.g. Tutorials).
2. Right click on Workspace and click New Project. In the Create Project window specify Debugging as the Project Name.
3. Debugging project will now be available in the workspace.

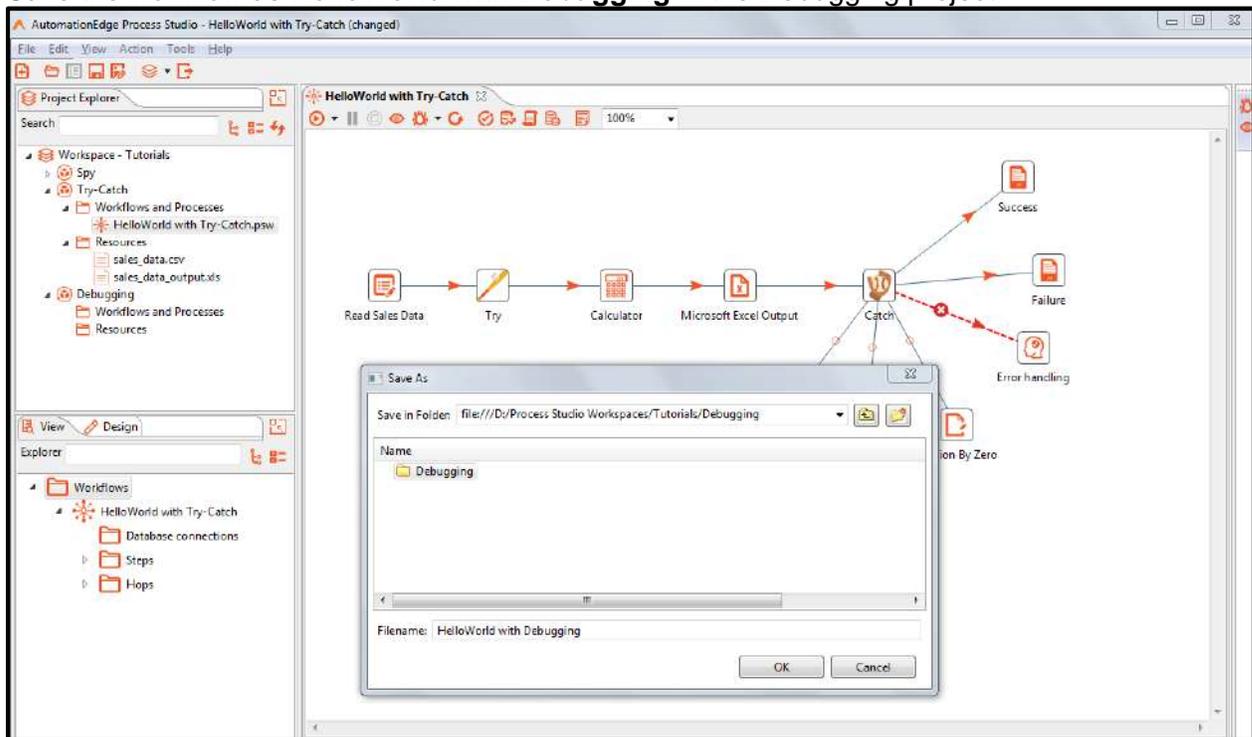
17.1 Create a workflow and setup Debugging features

In this section we will save the HelloWorld with Try-Catch workflow we created in the Try-Catch Project and save as HelloWorld with Debugging, add Breakpoints and execute the workflow with Debugging.

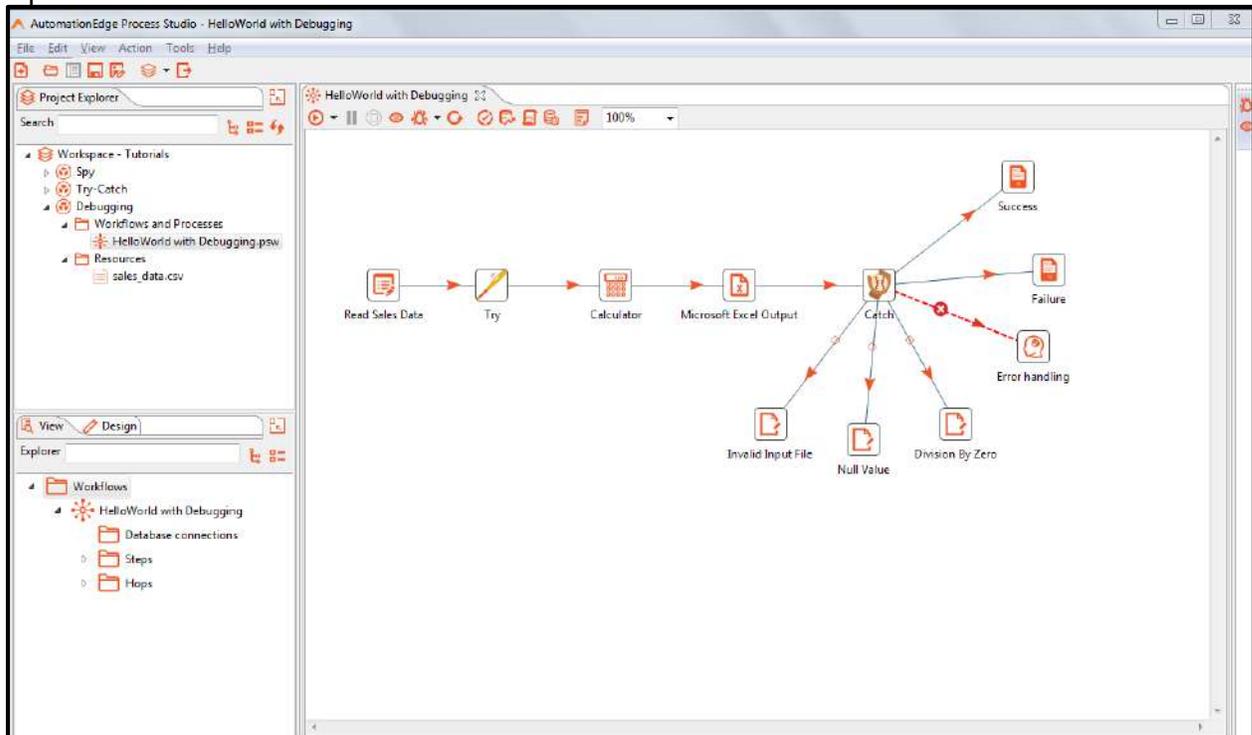
1. Let us open the **HelloWorld with Try-Catch** workflow in the Try-Catch project.



2. Save the workflow as **HelloWorld with Debugging** in the Debugging project.



3. **HelloWorld with Debugging** workflow is now visible in the Debugging Project. Also add an input file to the workflow as a Resource file.



4. Run the workflow to confirm its validity.

The screenshot displays the Process Studio interface for a workflow named 'HelloWorld_with_Debugging'. The workflow consists of the following steps: Read Sales Data, Try, Calculator, Microsoft Excel Output, Catch, and Success. The 'Catch' step is configured with three error-handling paths: 'Invalid Input File', 'Null Value', and 'Division By Zero'. A 'Failure' path is also shown, leading to an 'Error handling' step. The 'Execution Results' section at the bottom shows a table with columns for customer, product, quantity, unitprice, totalprice, Number of Errors, Error Description, Error Fields, Error Code, and Error Step Name. The table contains three rows of data, all with null values for error-related fields.

#	customer	product	quantity	unitprice	totalprice	Number of Errors	Error Description	Error Fields	Error Code	Error Step Name
1	Thunderboltz FC	football	20	1000	20000.0	<null>	<null>	<null>	<null>	<null>
2	Google	basketball	15	900	13500.0	<null>	<null>	<null>	<null>	<null>
3	AutomationEdge	cricket bat	10	500	5000.0	<null>	<null>	<null>	<null>	<null>

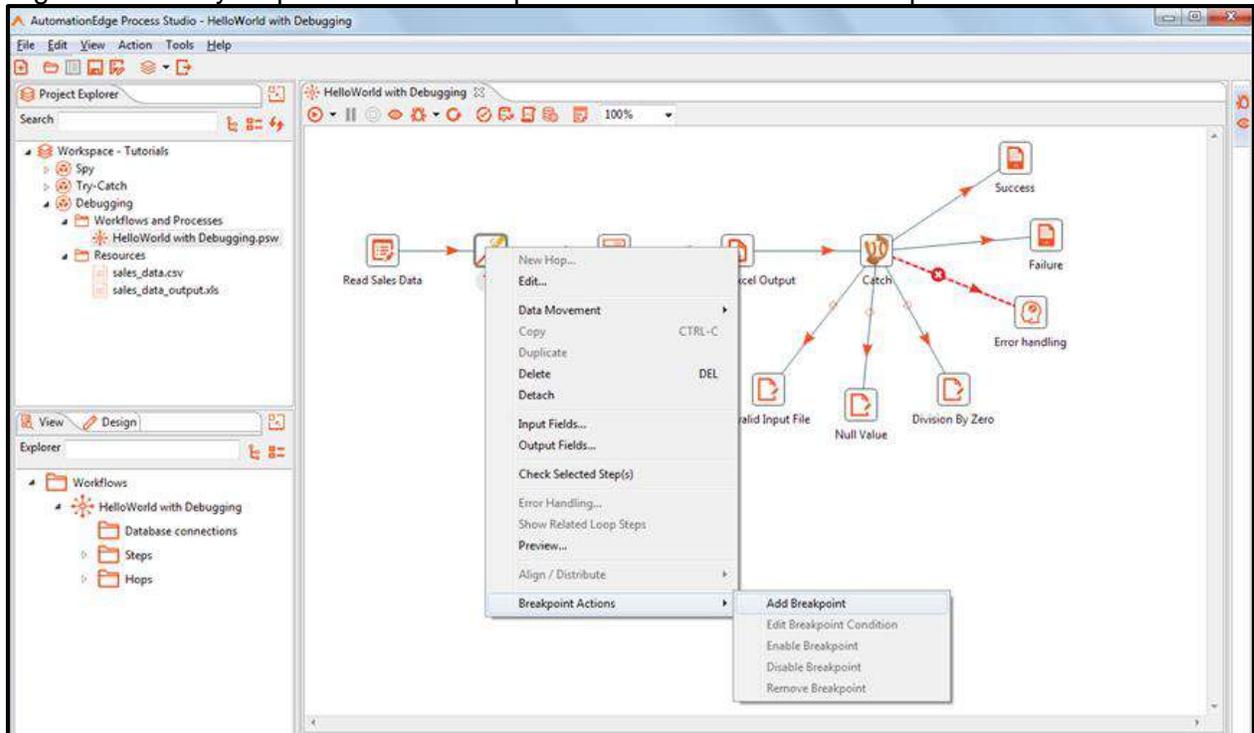
5. The contents of the file are as below.

A	B	C	D
customer	product	quantity	unitprice
Thunderboltz FC	football	20	20000
Google	basketball	15	1500
AutomationEdge	cricket bat	10	1000

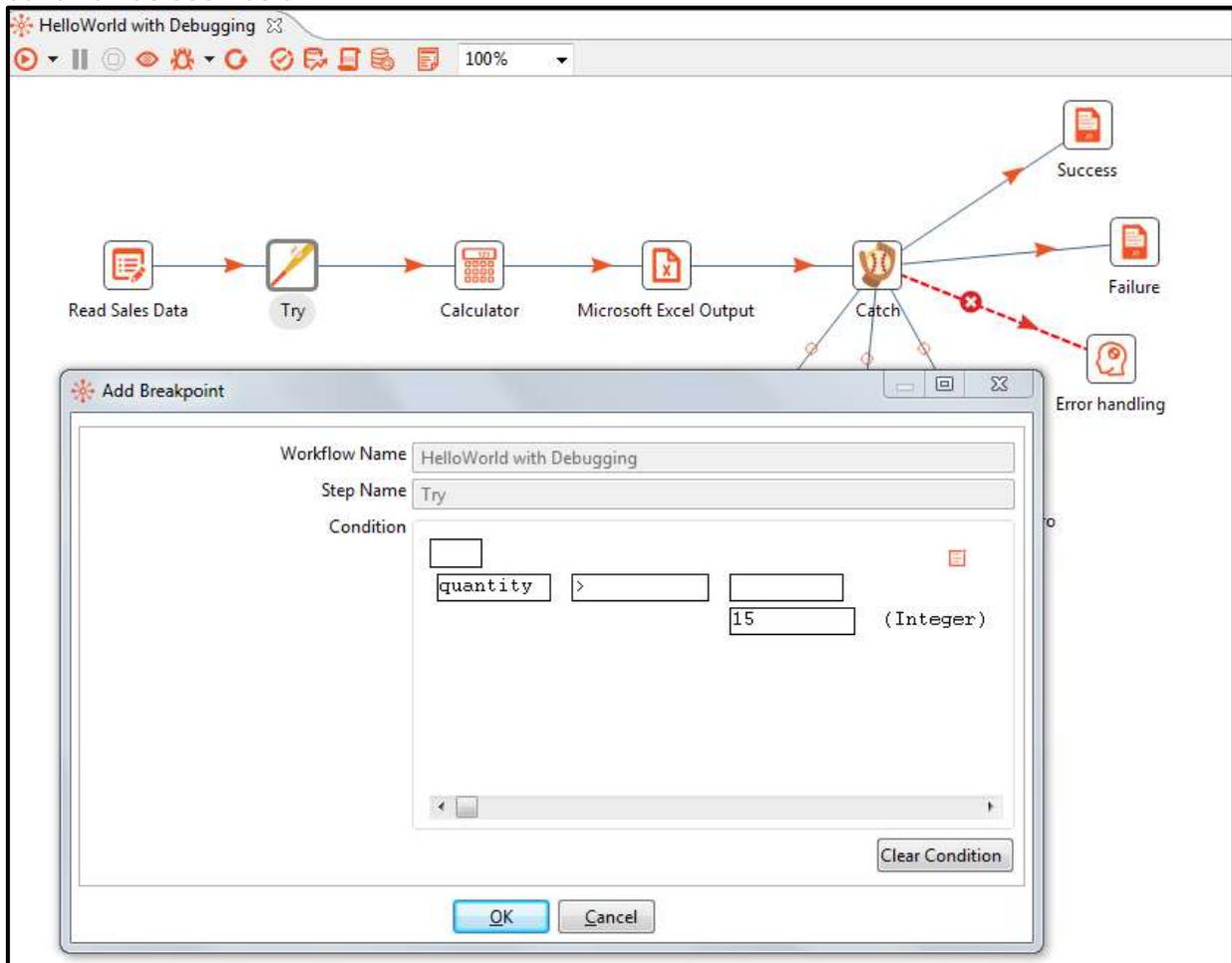
17.2 Set Breakpoints

Let us add a few Breakpoints in this section to our workflow.

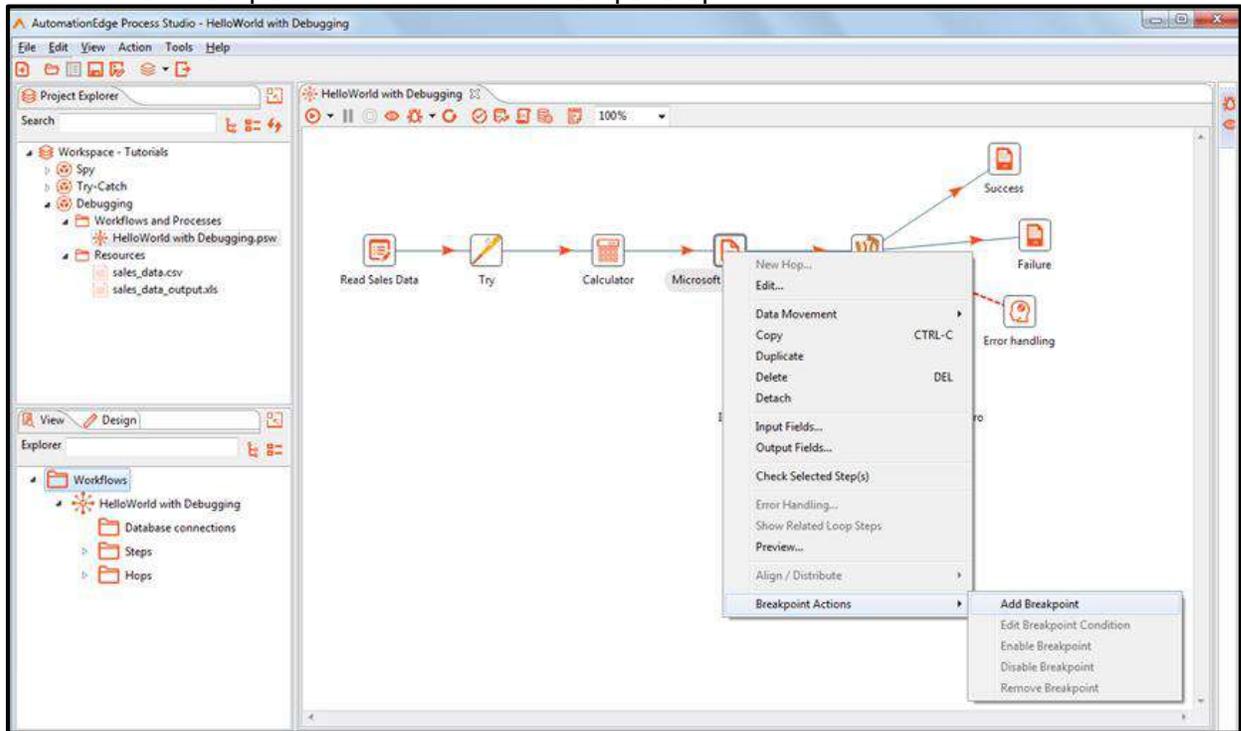
1. Right click on Try step and in the Breakpoint Actions select Add Breakpoint.



- Click on the condition fields to select a field from the steps in the stream or enter values. Add a condition as seen below.



3. Add another Breakpoint on Microsoft Excel Output step.



4. Add a condition as seen below.

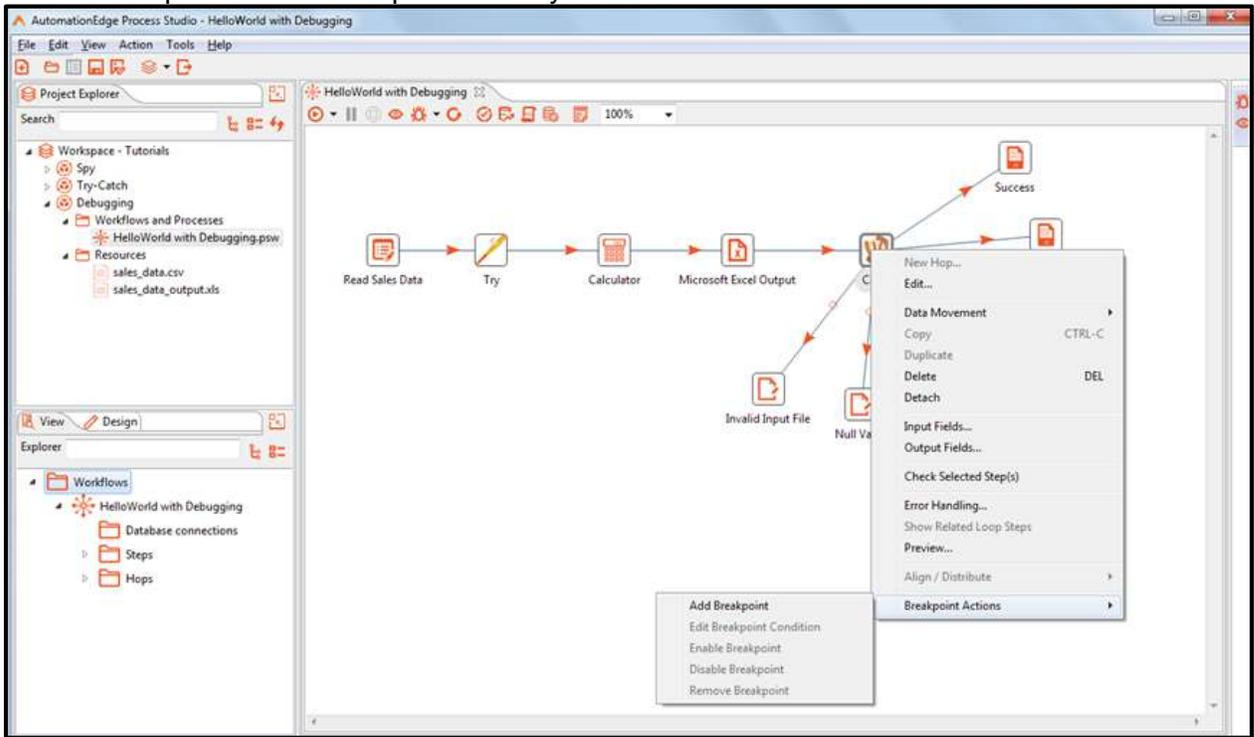
The screenshot displays the Process Studio interface for a workflow named "HelloWorld with Debugging". The workflow consists of the following steps: Read Sales Data, Try, Calculator, Microsoft Excel Output, and Catch. The Catch step is connected to three final outcomes: Success, Failure, and Error handling. A red dashed line with a plus sign indicates a breakpoint is being added to the Microsoft Excel Output step.

The "Add Breakpoint" dialog box is open, showing the following configuration:

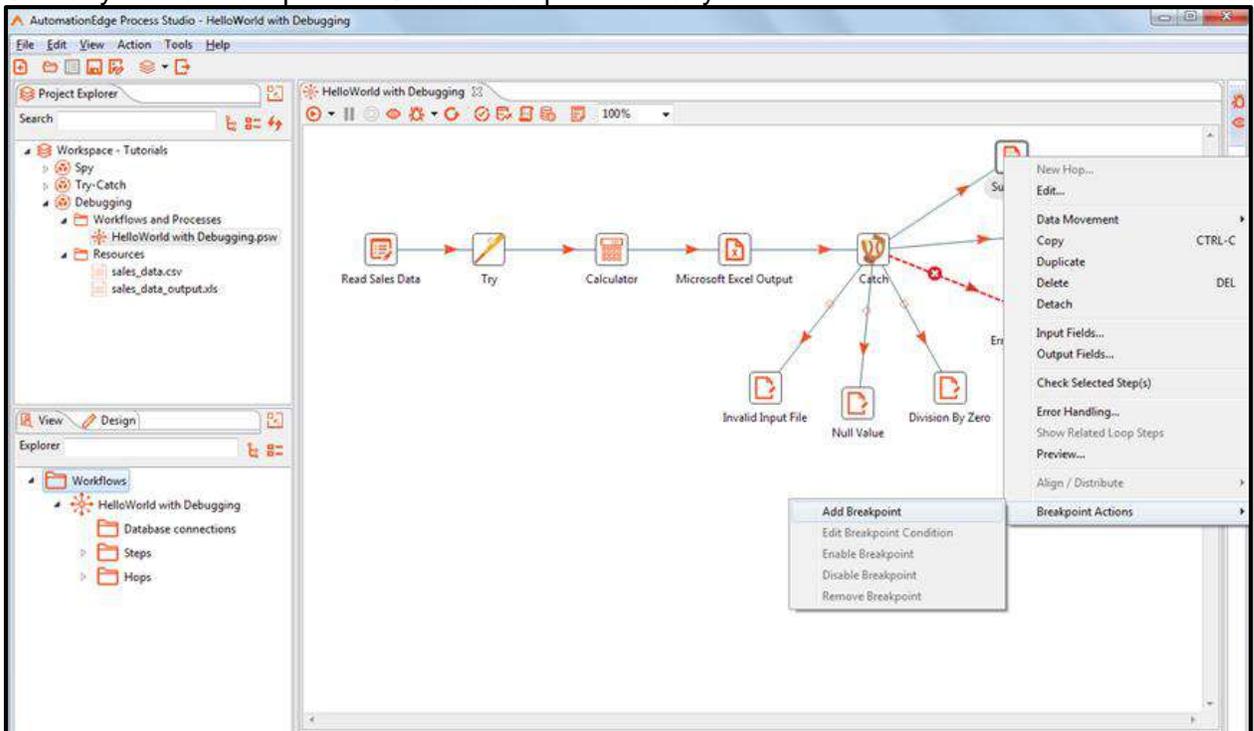
- Workflow Name: HelloWorld with Debugging
- Step Name: Microsoft Excel Output
- Condition: `totalprice > 1000` (Number)

The dialog box includes a "Clear Condition" button and "OK" and "Cancel" buttons at the bottom.

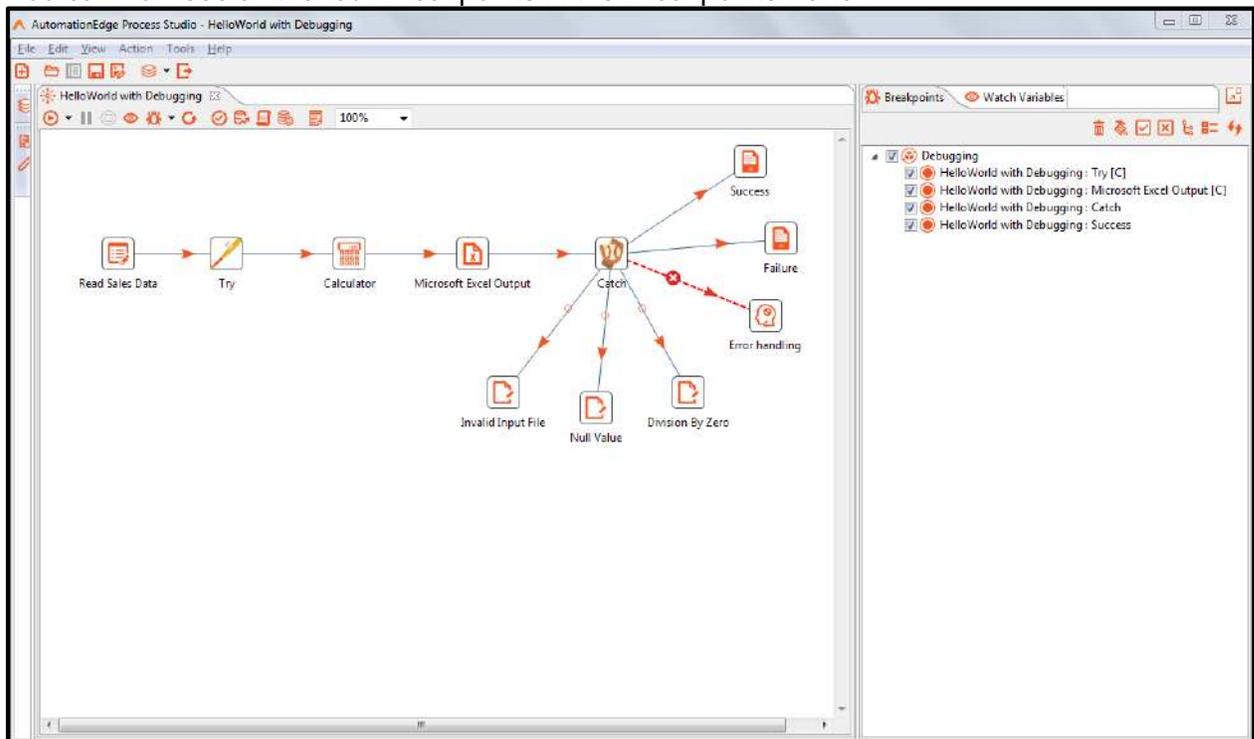
5. Add a Breakpoint to Catch step without any condition.



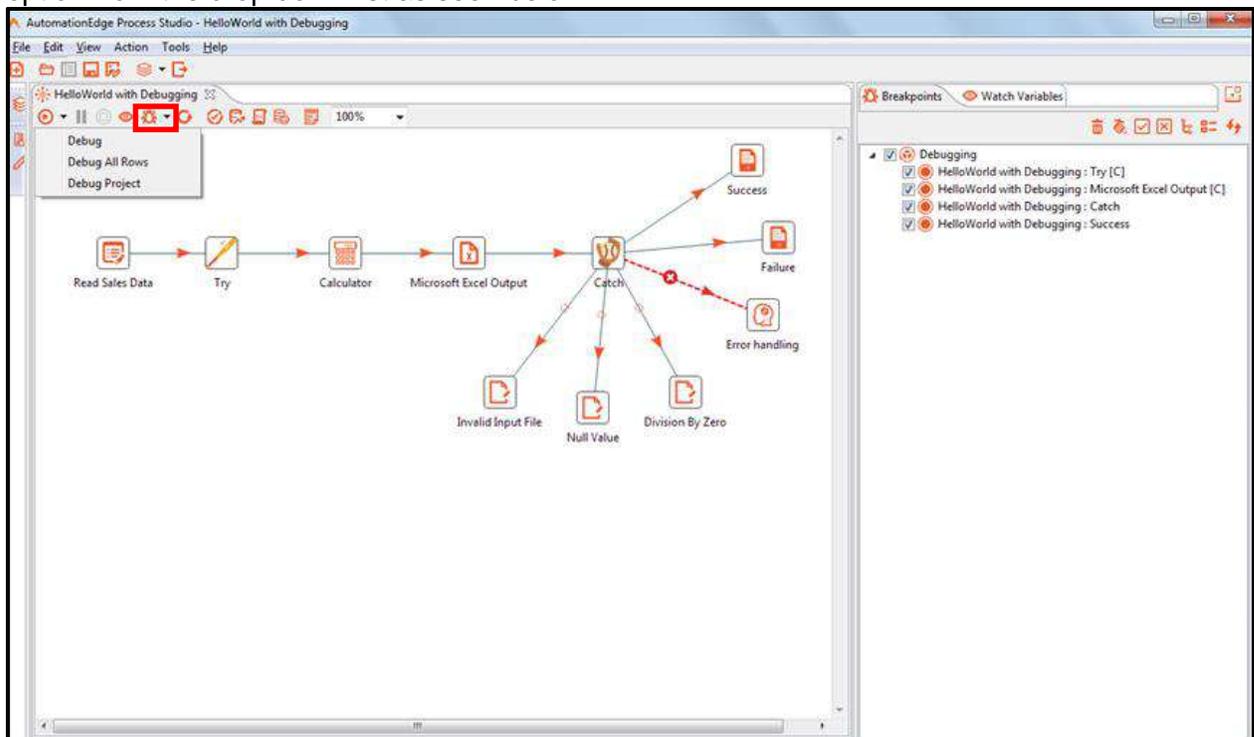
6. Similarly Add a Breakpoint to Success step without any condition.



7. You can now see all the four Breakpoints in the Breakpoints Panel.



8. You can execute the workflow in debug mode by clicking the Debug icon or by selecting an option from the drop down list as seen below.



9. We can start workflow execution in any of the three execution modes,
 - Debug (The first option is the same as clicking the Debug icon)
 - Debug All rows or
 - Debug Project

In the following sections we will run the workflow in all the three modes,

Note the contents of the input file as follows,

customer	product	quantity	unitprice
Thunderboltz FC	football	20	1000
Google	basketball	15	900
AutomationEdge	cricket bat	10	500

The workflow has two breakpoints with conditions as below

- quantity >15 on Try step and
- totalprice>1000 on Microsoft Excel output step.

And two breakpoints without conditions as below

- on Catch step
- on Success step

17.3 Debug

In the first case we will run the workflow in Debug mode by directly clicking the Debug button or selecting the first option Debug in the list.

In Debug mode the workflow will stop at steps with 'Breakpoints with condition' if condition is met and it stops at all steps with 'Breakpoints without condition'; for all the rows.

1. Execute the workflow in Debug mode by directly clicking the Debug button or selecting the first option Debug in the list
2. We can see that the workflow stops as Try Breakpoint for the first row as the condition $Quantity > 15$ is met. In the Debug view we can see the row data as well as the Watch variables.

Execution Results

Logging Step Metrics Metrics Preview data Debug

Row Data					Variables		
#	customer	product	quantity	unitprice	#	Name	Value
1	Thunderboltz FC	football	20	1000	1	Internal.Entry.Current.Directory	file:///D:/Workspace/Data
2					2		
3					3		
4					4		
5					5		
6					6		

3. Now click the first icon Resume as seen in the lower pane.

The screenshot displays the Process Studio interface for a workflow named "HelloWorld_with_Debugging". The workflow consists of several steps: "Read Sales Data", "Try", "Calculator", "Microsoft Excel Output", and "Catch". The "Catch" step is highlighted with a red box and a red "X" icon, indicating an error. The "Catch" step has four outgoing paths: "Success", "Failure", "Error handling", and "Division By Zero". The "Error handling" path is also highlighted with a red box and a red "X" icon. The "Try" step is highlighted with a blue box and a blue "X" icon, indicating a warning or error.

Below the workflow diagram is the "Execution Results" pane. It contains a "Logging" tab, "Step Metrics", "Metrics", "Preview data", and "Debug" tabs. The "Logging" tab is active, showing a "Row Data" table and a "Variables" table.

#	customer	product	quantity	unitprice
1	Thunderboltz FC	football	20	1000
2				
3				
4				
5				
6				

#	Name	Value
1	Internal.Entry.Current.Directory	file:///D:/Workspace/Data
2		
3		
4		
5		
6		

- On Resume the workflow progresses to the next Breakpoint. On the same row the Breakpoint condition is met on Microsoft Excel Output step. from the stepping commands click Resume.

The screenshot displays the Process Studio interface for a workflow named "HelloWorld_with_Debugging". The workflow consists of the following steps:

- Read Sales Data (Completed)
- Try (Completed)
- Calculator (Completed)
- Microsoft Excel Output (Breakpoint, highlighted with a blue box)
- Catch (Error handling step)
- Success (Final outcome)
- Failure (Final outcome)
- Error handling (Sub-step of Catch)
- Invalid Input File (Sub-step of Catch)
- Null Value (Sub-step of Catch)
- Division By Zero (Sub-step of Catch)

The "Microsoft Excel Output" step is currently selected, and the "Catch" step is active, indicating an error has occurred. The "Execution Results" panel at the bottom shows the following data:

Row Data						Variables		
#	customer	product	quantity	unitprice	totalprice	#	Name	Value
1	Thunderboltz FC	football	20	1000	20000.0	1	Internal.Entry.Current.Directory	file:///D:/Wc
2						2		
3						3		
4						4		
5						5		

- The workflow execution stops at the next breakpoint Catch. Since the breakpoint does not have any condition it always stops on such steps. Click Resume from the Stepping commands.

The screenshot displays the Process Studio interface for a workflow named "HelloWorld_with_Debugging". The workflow consists of several steps: "Read Sales Data", "Try", "Calculator", "Microsoft Excel Output", and "Catch". The "Catch" step is highlighted with a blue border, indicating it is the current execution point. It has six outgoing paths: "Success", "Failure", "Error handling", "Invalid Input File", "Null Value", and "Division By Zero". The "Error handling" path is shown with a red dashed line and a red 'X' icon, indicating an error state.

Below the workflow diagram is the "Execution Results" section, which includes tabs for "Logging", "Step Metrics", "Metrics", "Preview data", and "Debug". The "Row Data" table shows the following data:

#	customer	product	quantity	unitprice	totalprice
1	Thunderboltz FC	football	20	1000	20000.0
2					
3					
4					

The "Variables" table shows the following data:

#	Name	Value
1	Internal.Entry.Current.Directory	file:///D:/Wc
2		
3		
4		

- After Catch the Success success step has a Breakpoint without condition. So the workflows stops at this step. Click Resume from the Stepping commands below.

The screenshot displays the Process Studio interface for a workflow named "HelloWorld_with_Debugging". The workflow consists of the following steps:

- Read Sales Data (with a green checkmark)
- Try (with a green checkmark)
- Calculator (with a green checkmark)
- Microsoft Excel Output (with a blue square icon)
- Catch (with a red 'x' icon, indicating a breakpoint)
- Success (with a blue square icon)
- Failure (with a red square icon)
- Error handling (with a red square icon)
- Invalid Input File (with a red square icon)
- Null Value (with a red square icon)
- Division By Zero (with a red square icon)

The "Execution Results" section is visible at the bottom, showing the following data:

Row Data						Variables		
#	customer	product	quantity	unitprice	totalprice	#	Name	Value
1	Thunderboltz FC	football	20	1000	20000	1	Internal.Entry.Current.Directory	file:///D:/W
2						2		
3						3		
4						4		

- For the second row the workflow does not stop on Try since the condition $Quantity > 15$ is not met. It directly goes to the Microsoft Excel step since the condition $totalprice > 1000$ is met. Click Resume from the Stepping commands below.

The screenshot displays the Process Studio interface for a workflow named "HelloWorld_with_Debugging". The workflow consists of the following steps:

- Read Sales Data (Completed)
- Try (Completed)
- Calculator (Completed)
- Microsoft Excel Output (Completed)
- Catch (Active)

The Catch step has several outgoing paths:

- Success
- Failure
- Error handling
- Invalid Input File
- Null Value
- Division By Zero

The "Execution Results" section is visible at the bottom, showing the following data:

Row Data						Variables		
#	customer	product	quantity	unitprice	totalprice	#	Name	Value
1	Google	basketball	15	900	13500.0	1	Internal.Entry.Current.Directory	file:///D:/Wc
2						2		
3						3		
4						4		
5						5		

- The workflow stops at Catch since there is a breakpoint without a condition on this step. Click Resume from the Stepping commands below.

Execution Results

Logging Step Metrics Metrics Preview data Debug

Row Data						Variables		
#	customer	product	quantity	unitprice	totalprice	#	Name	Value
1	Google	basketball	15	900	13500.0	1	Internal.Entry.Current.Directory	file:///D:/Wc
2						2		
3						3		
4						4		
5						5		

- The workflow stops at Success since there is a breakpoint without a condition on this step. Click Resume from the Stepping commands below.

Execution Results

Logging Step Metrics Metrics Preview data Debug

Row Data							Variables		
#	customer	product	quantity	unitprice	totalprice	Nu	#	Name	Value
1	Google	basketball	15	900	13500		1	Internal.Entry.Current.Directory	file:///D:/Wc
2							2		
3							3		
4							4		
5							5		

10. For the second row the condition $quantity > 10$ is not met on the Try step, hence the workflow execution does not pause on Try step. However, the condition $totalprice > 1000$ is met on Microsoft Excel Output step hence workflow pauses here; and then on Catch and Success Steps with Breakpoints but without conditions.
11. Click Resume All from the Stepping commands below to execute the workflow without any pauses.

The screenshot displays the Process Studio interface for a workflow named 'HelloWorld_with_Debugging'. The workflow diagram shows a sequence of steps: Read Sales Data, Try, Calculator, Microsoft Excel Output, Catch, and then a split into Success, Failure, and Error handling. The Catch step has three sub-steps: Invalid Input File, Null Value, and Division By Zero. The execution results section is visible at the bottom, showing a table with row data and variables.

Execution Results

Logging Step Metrics Metrics Preview data Debug

Row Data						Variables		
#	customer	product	quantity	unitprice	totalprice	#	Name	Value
1	AutomationEdge	cricket bat	10	500	5000.0	1	Internal.Entry.Current.Directory	file:///D:/W
2						2		
3						3		
4						4		

12. On Resume All the workflow execution completes at once without any further pauses.

The screenshot displays the AutomationEdge Process Studio interface. The main window shows a workflow diagram with the following steps: Read Sales Data, Try, Calculator, Microsoft Excel Output, and Catch. The Catch step has three outgoing paths: Success, Failure, and Error handling. The Error handling path has three outgoing paths: Invalid Input File, Null Value, and Division By Zero. The right-hand pane shows the Breakpoints and Watch Variables section, with a list of debugging events. The Execution Results section at the bottom shows a table with the following data:

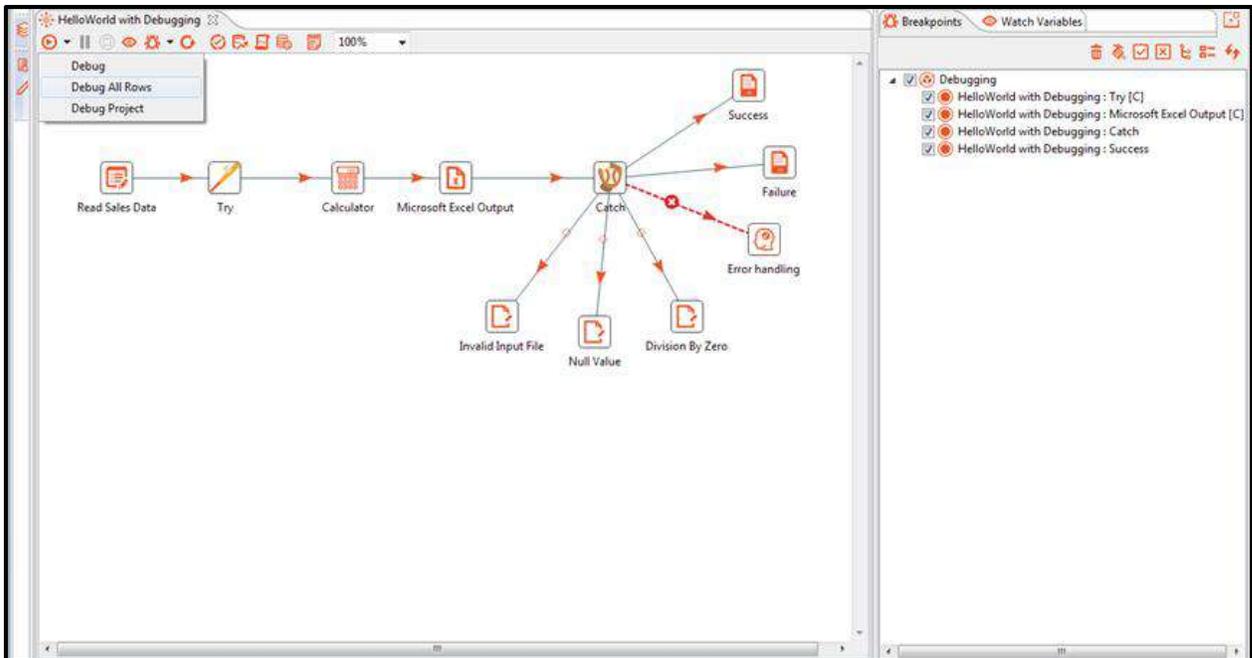
#	customer	product	quantity	unitprice	totalprice	Number of Errors	Error Description	Err
1	Thunderboltz FC	football	20	1000	20000.0	<null>	<null>	<n
2	Google	basketball	15	900	13500.0	<null>	<null>	<n
3	AutomationEdge	cricket bat	10	500	5000.0	<null>	<null>	<n

13. This completes the demonstration of executing a workflow in Debug mode with Debug option by clicking the Debug icon or selecting the first option Debug from the list.

17.4 Debug All Rows

In this section we will execute **HelloWorld with Debugging** with the **Debug All Rows** option. In this case, each row will be paused at each step (except input type steps).

1. Select Debug All Rows from the Debug icon drop down to execute the workflow.



- Microsoft Excel Input is an input step hence, workflow execution does not pause at this step. It pauses at the Try step. Click Resume from the Stepping commands.

Execution Results

Logging Step Metrics Metrics Preview data Debug

Row Data					Variables	
#	customer	product	quantity	unitprice	#	Name
1	Thunderboltz FC	football	20	1000	1	Internal.Entry.CurrentDirectory
2					2	
3					3	
4					4	
5					5	

#	Name	Value
1	Internal.Entry.CurrentDirectory	file:///D:/Workspace/D
2		
3		
4		
5		

- As expected the workflow pauses at every step in every row irrespective of whether any condition is met or not.

The screenshot displays the Process Studio interface for a workflow named 'HelloWorld_with_Debugging'. The workflow diagram consists of the following steps: 'Read Sales Data' (green checkmark), 'Try' (blue pencil), 'Calculator' (blue calculator), 'Microsoft Excel Output' (orange document), and 'Catch' (orange shield). The 'Catch' step has five outgoing paths: 'Success' (orange document), 'Failure' (orange document), 'Error handling' (orange document with a red 'x'), 'Invalid Input File' (orange document), and 'Null Value' (orange document). The 'Error handling' path is highlighted with a red dashed line and a red 'x'.

Below the workflow diagram is the 'Execution Results' section, which includes tabs for 'Logging', 'Step Metrics', 'Metrics', 'Preview data', and 'Debug'. The 'Row Data' table shows the following data:

#	customer	product	quantity	unitprice
1	Google	basketball	15	900
2				
3				
4				
5				

The 'Variables' table shows the following data:

#	Name	Value
1	Internal.Entry.Current.Directory	file:///D:/Workspace/
2		
3		
4		
5		

- The execution can be complete step by step or Resume all to complete workflow execution.

17.5 Debug Project

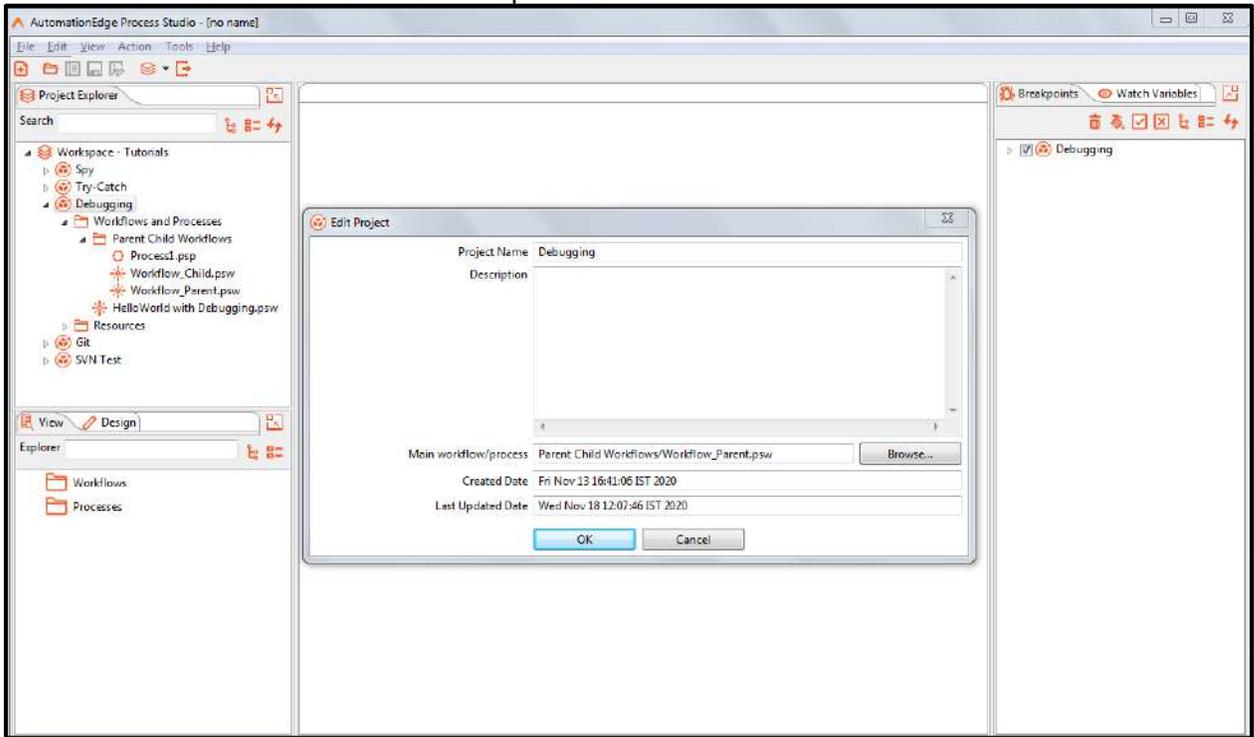
Debug Project workflow execution option starts workflow execution from the main workflow specified in the project and start debugging from the main workflow. It behaves exactly like the Debug option (i.e. it pauses at Breakpoints when condition is met and at Breakpoints without any condition).

In this section we will demonstrate how workflow execution of parent is invoked with **Debug Project** option.

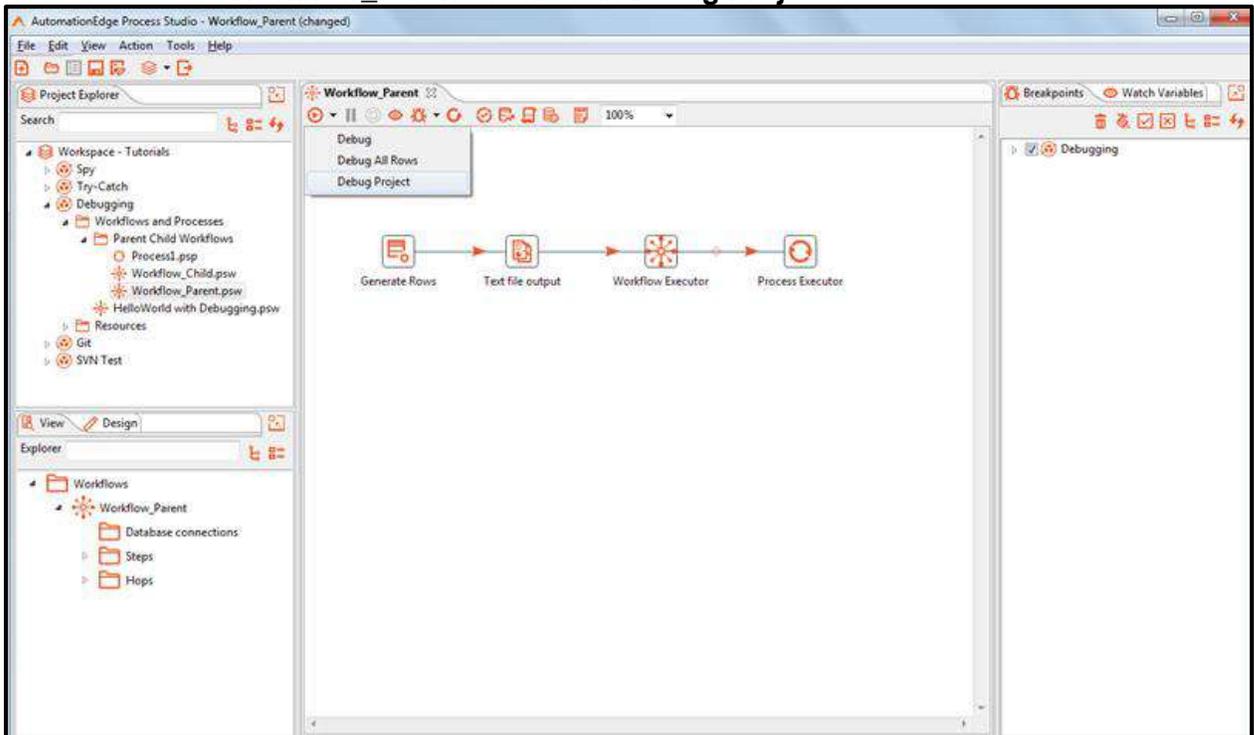
Note:

The main file specified should not be a process (.psp). Process files as the main file will be supported in upcoming releases.

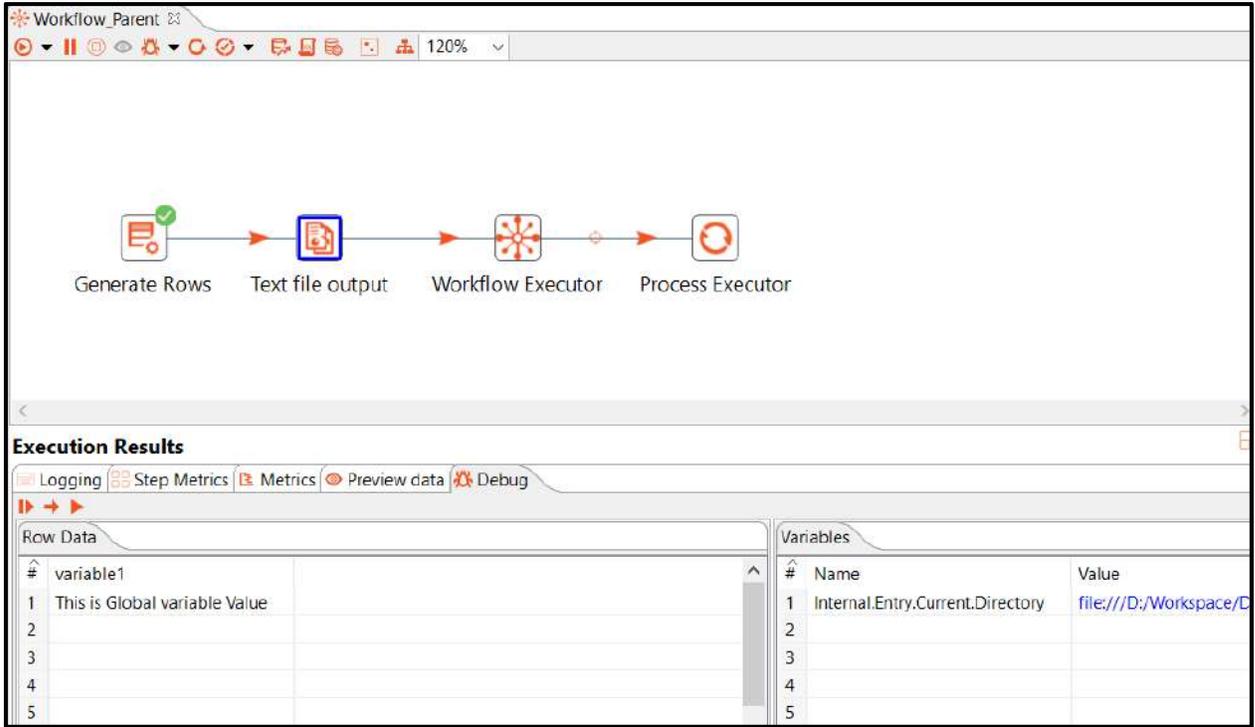
1. Right Click the project **Debugging** to check that **Workflow_Parent** in the folder **Parent Child Workflows** folder is the main workflow/process.



2. Breakpoints are defined at a Text File Output step in a parent workflow as well as a Text File Output step in a child workflow.
3. Now execute the **Workflow_Parent** workflow in **Debug Project** mode.



- The workflow pauses at the first Breakpoint i.e. Text File output step. Click Resume from the Stepping commands. The next Breakpoint condition in in the child workflow defined in the Workflow Executor step.



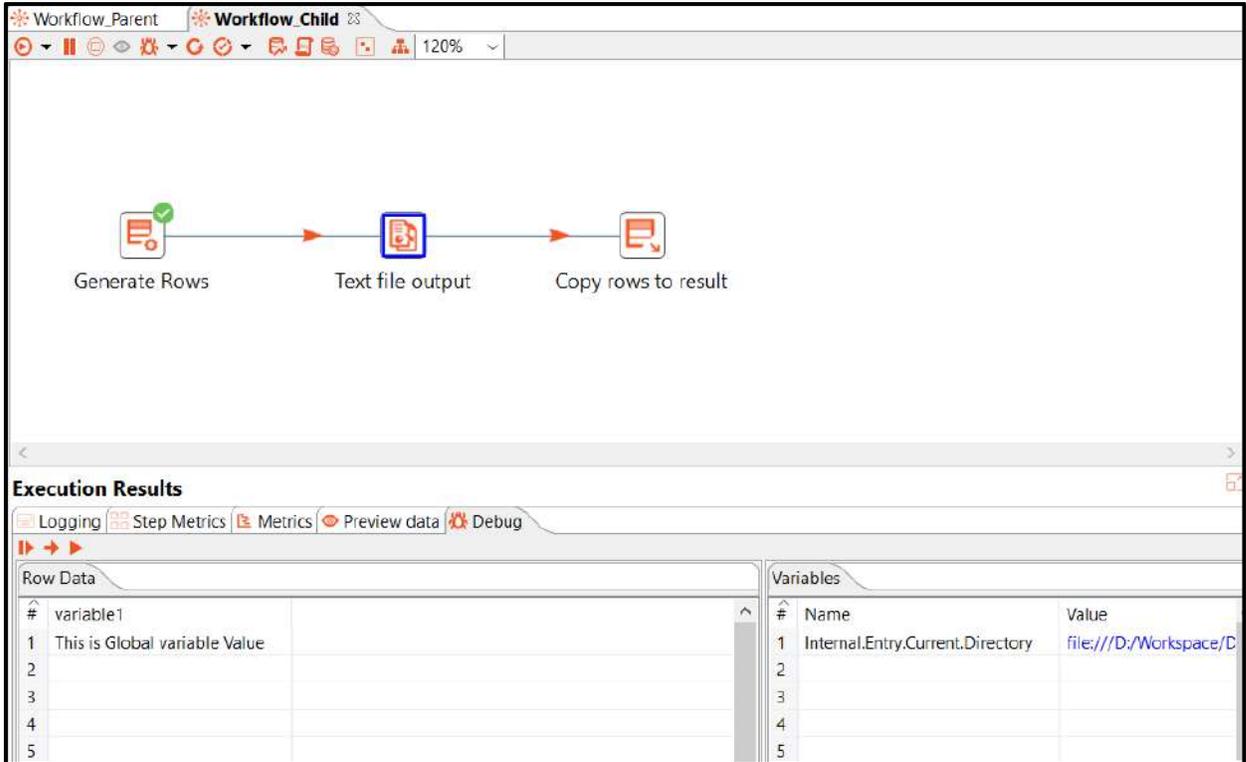
The screenshot displays the Process Studio interface for a workflow named "Workflow_Parent". The workflow diagram shows four steps: "Generate Rows", "Text file output", "Workflow Executor", and "Process Executor". The "Text file output" step is highlighted with a blue box, indicating it is the current breakpoint. The "Workflow Executor" step has a red diamond icon, suggesting a child workflow is being executed.

Below the workflow diagram, the "Execution Results" section is visible. It includes tabs for "Logging", "Step Metrics", "Metrics", "Preview data", and "Debug". The "Preview data" tab is active, showing two tables: "Row Data" and "Variables".

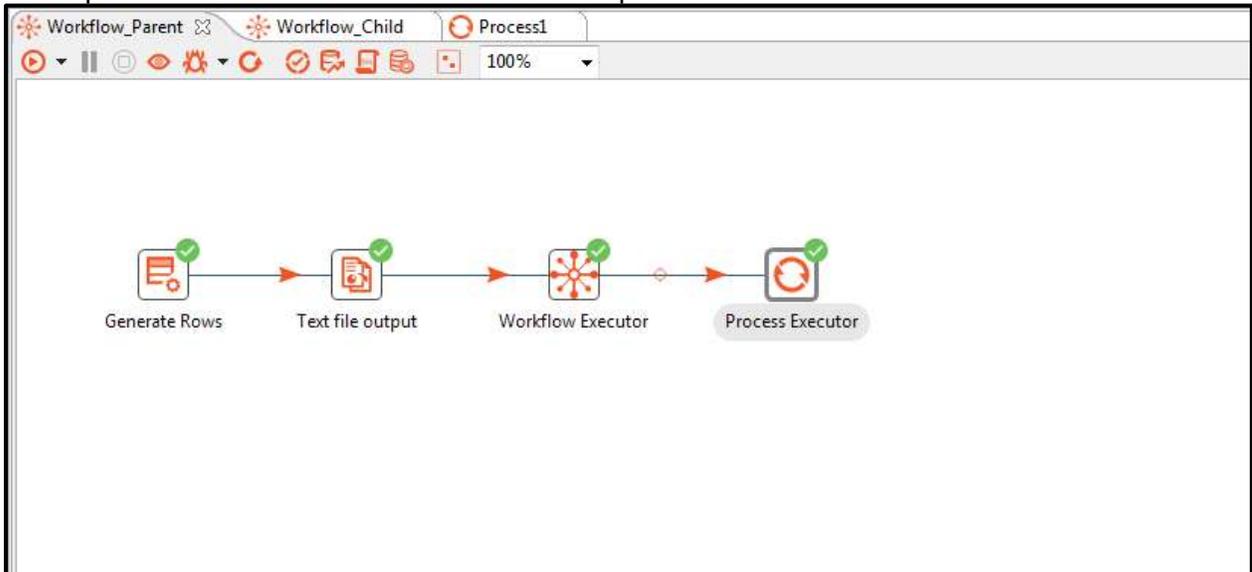
#	variable1
1	This is Global variable Value
2	
3	
4	
5	

#	Name	Value
1	Internal.Entry.Current.Directory	file:///D:/Workspace/D
2		
3		
4		
5		

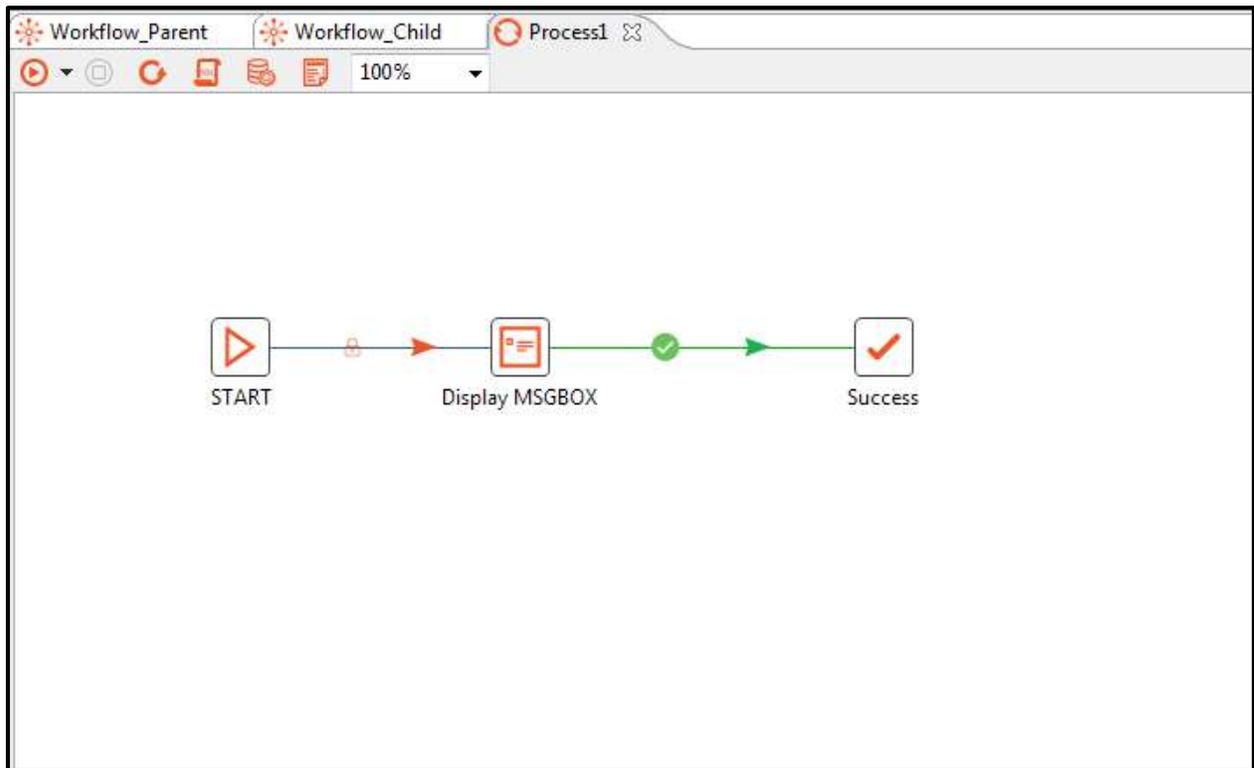
- The child workflow Workflow_Child is automatically opened in a new tab. The workflow pauses at the Text File output step in the Child workflow. Click Resume. Since there are no more Breakpoints in the child workflow the execution goes back to the parent workflow.



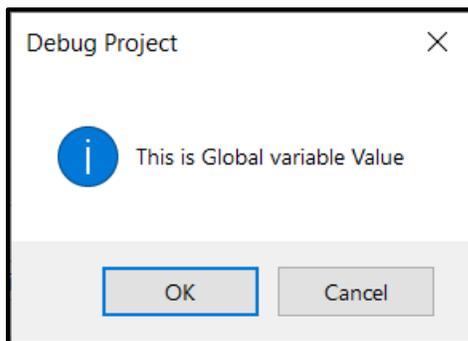
- In the parent workflow the Process executor step executes.



- The Process executor step has a display message box step.



- Accordingly, a popup message is displayed as below.



- This completes demonstration of executing a project with **Debug Project** option.

Note:

In case of Parent-Child workflows when debugging is started from Parent and opens the child workflow to complete execution. It is advisable not to close the child workflow while the workflow is still executing a step in the child workflow. Child workflow can be closed once the workflow is complete or the workflow is terminated. Else, if you close the child workflow and then try to terminate the parent workflow it closes process studio with "Widget is disposed" error message.

18 Project 11: Git Repository Integration

In this section we shall create a project called Git. We will demonstrate Process Studio integration with Git Repository and perform the following Subversion Controls using Process Studio.

- Create Git Repository
- Commit, Push, Pull
- Branches
- Merge
- Conflict
- Visual Diff

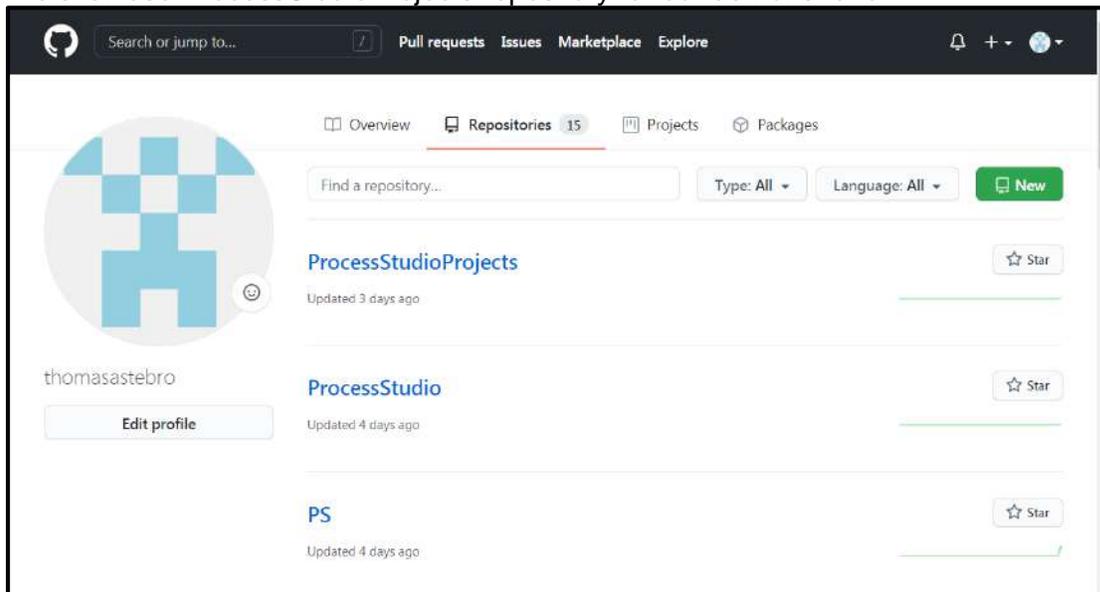
Let us begin this section with creating an online Git Repository.

18.1 Repository

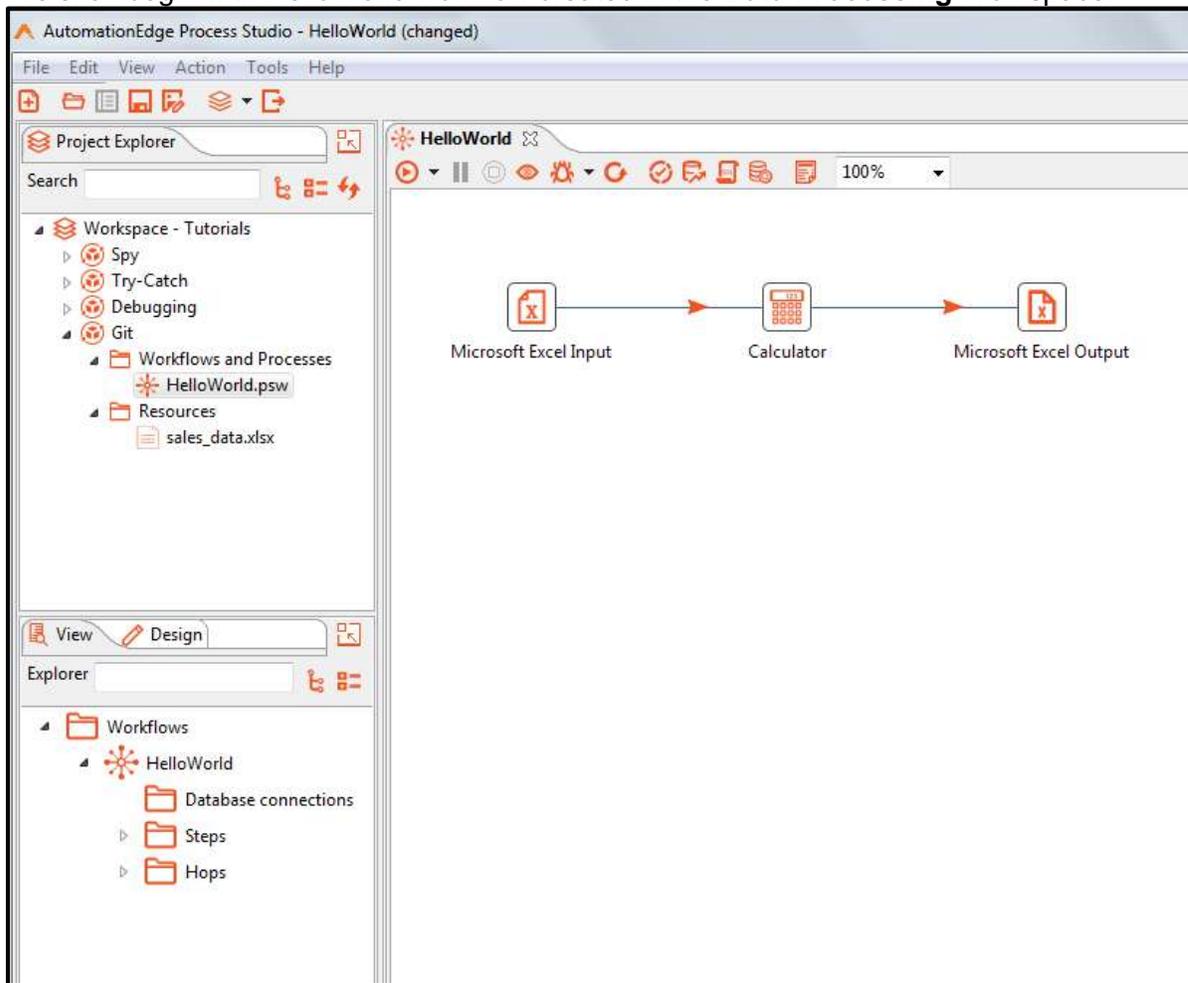
18.1.1 Online Git Repository

In this section we shall create an online GIT repository.

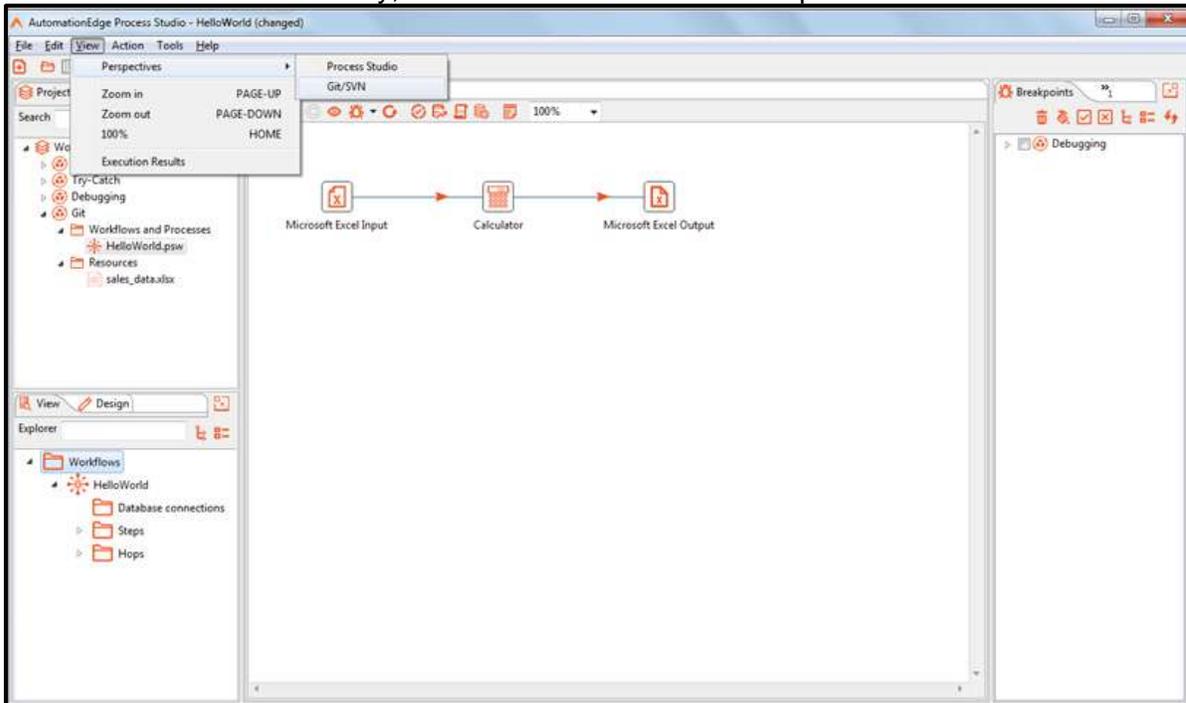
1. Create an online Repository account at github.com and add a repository.
2. We shall use ProcessStudioProjects repository for our demonstration.



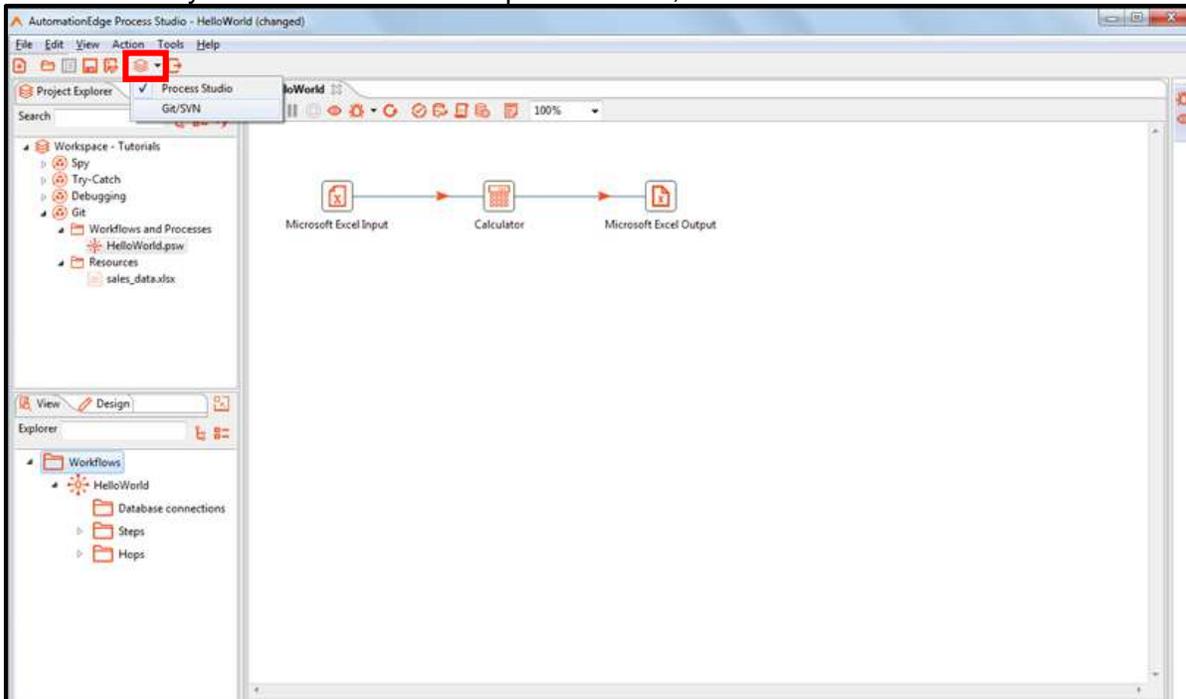
3. We shall begin with HelloWorld workflow created in the **Data Processing** Workspace.



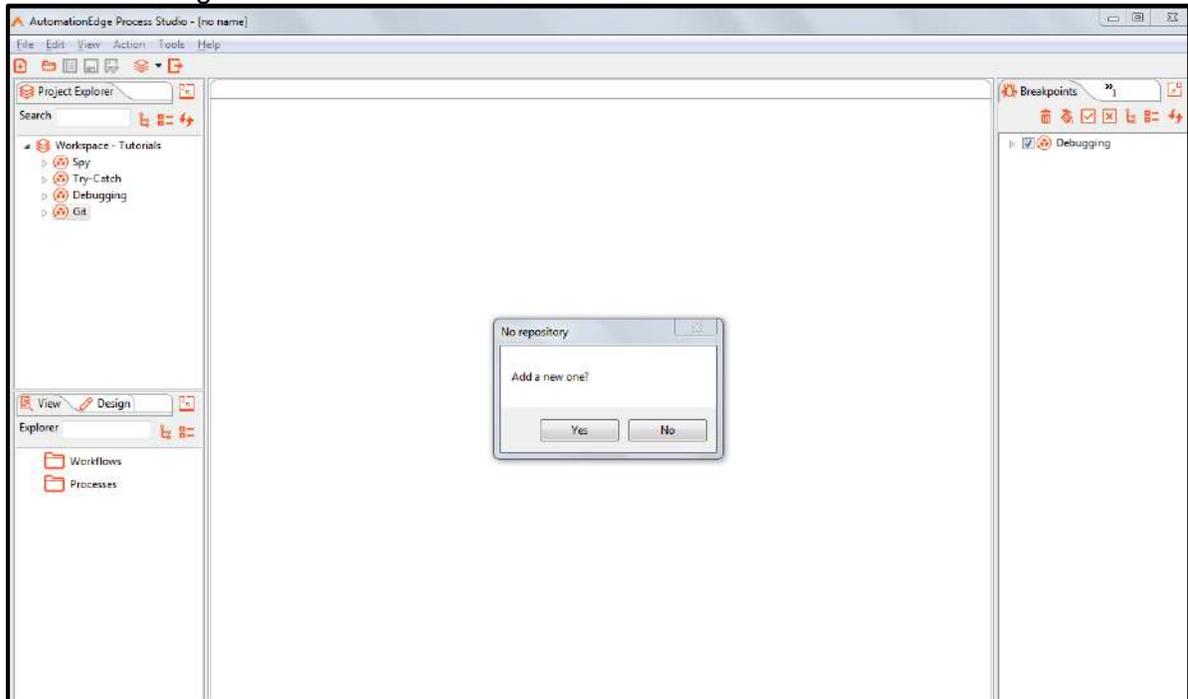
- Under View menu there is a menu option Perspectives. You can see two Perspectives – Process Studio and Git/SVN. Currently, we are in Process Studio Perspective.



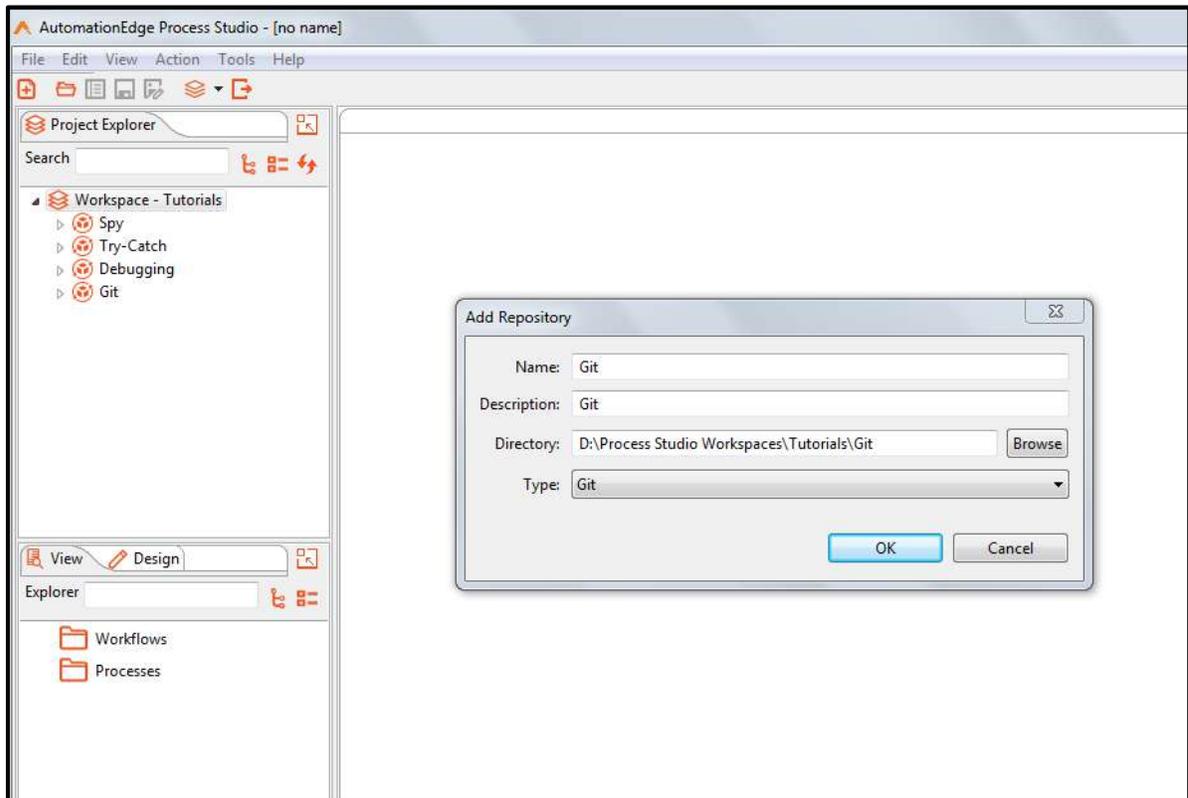
- Perspective icon is also available on the toolbar as seen below where it is highlighted in red. From the drop down you may select one of the two Perspectives – Process Studio or Git/SVN.
- We are already in Process Studio Perspective. Now, select Git/SVN.



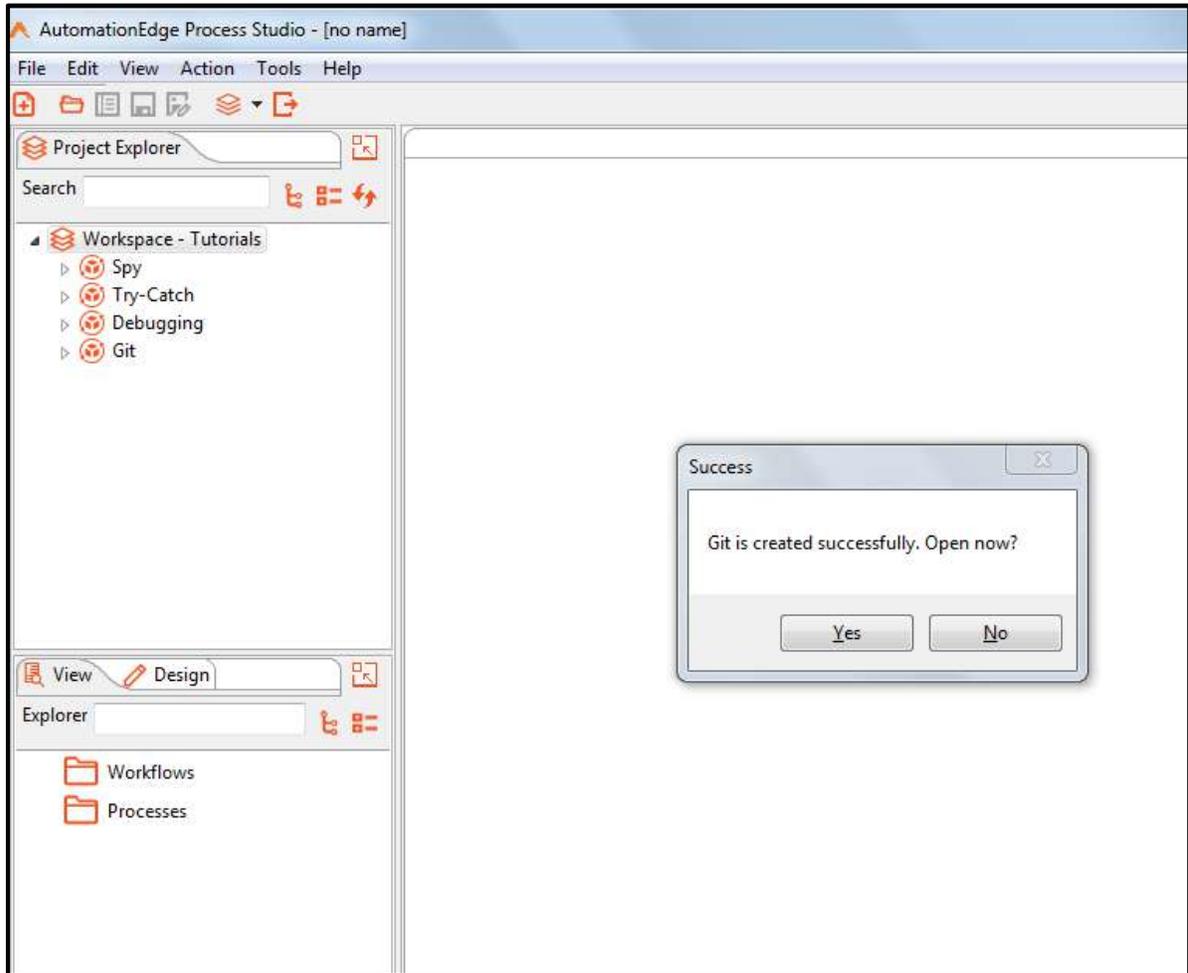
7. Currently we do not have any repository in the file system. Accordingly, a popup 'No repository' is shown asking to Add a new one? Select Yes.



8. Provide new Repository details as seen below such as Repository Name, Description, Directory and type (Git or SVN). It is important to note that the Directory for the repository should be same as the Project directory. So any changes in the Project would reflect in the Git/SVN perspective.
9. These changes in Process Studio or Git/SVN Perspective can be committed to the local repository and pushed to the online Repository as well.
10. Click OK.

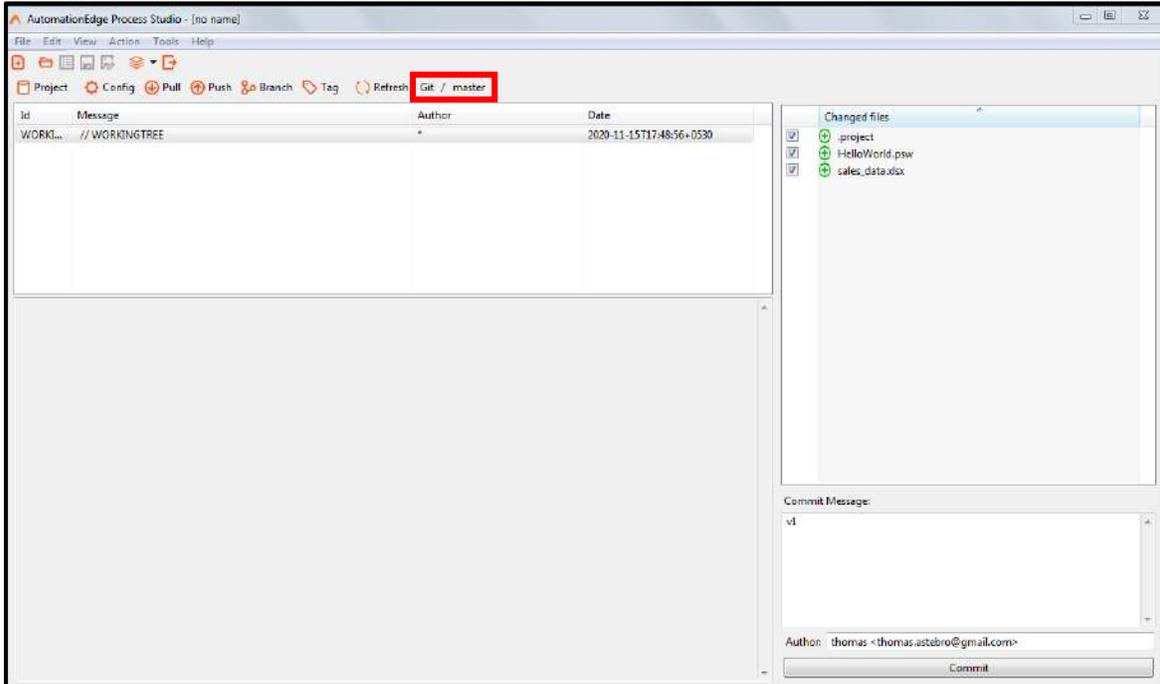


11. A Success popup 'Git create successfully' appears. Click Yes to open repository.

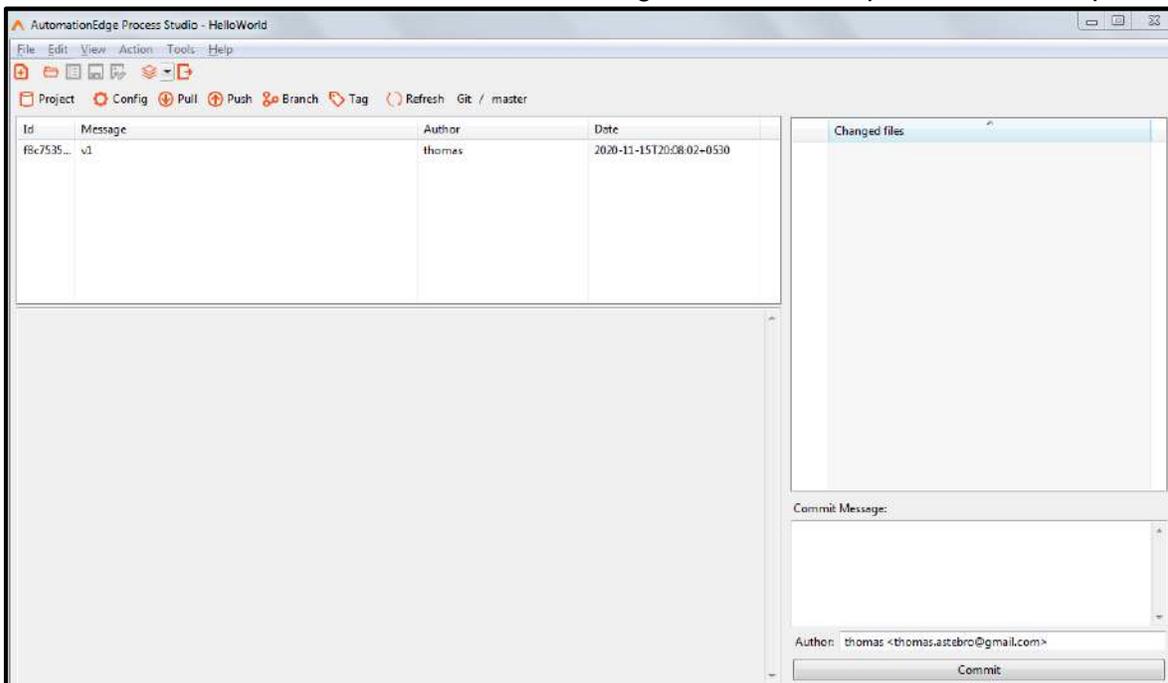


18.2 Commit

12. We can now see the Git repository is open on the master branch. This is highlighted in red below.
13. In the Working Tree on the right pane you can see Changed files. The green circles indicate that these are newly added files.
14. You can provide a commit message (e.g. v1, as provided here) and click the Commit button.

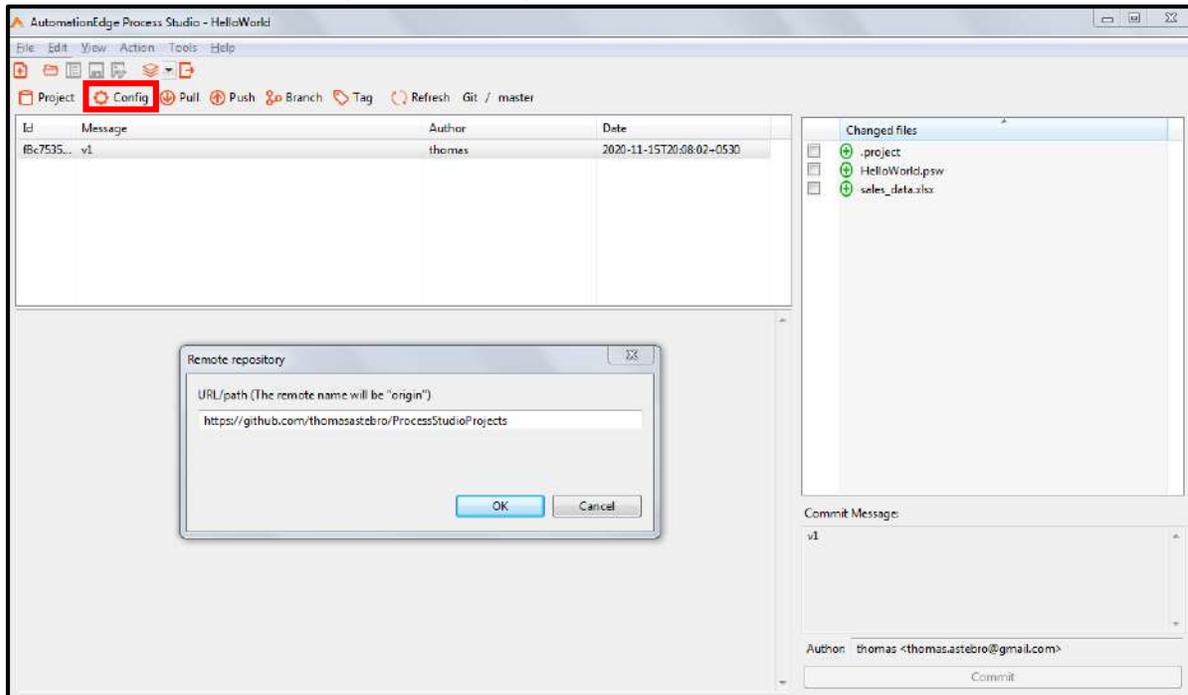


15. You can now see the first commit with the message v1 in the main pane of the Git repository.

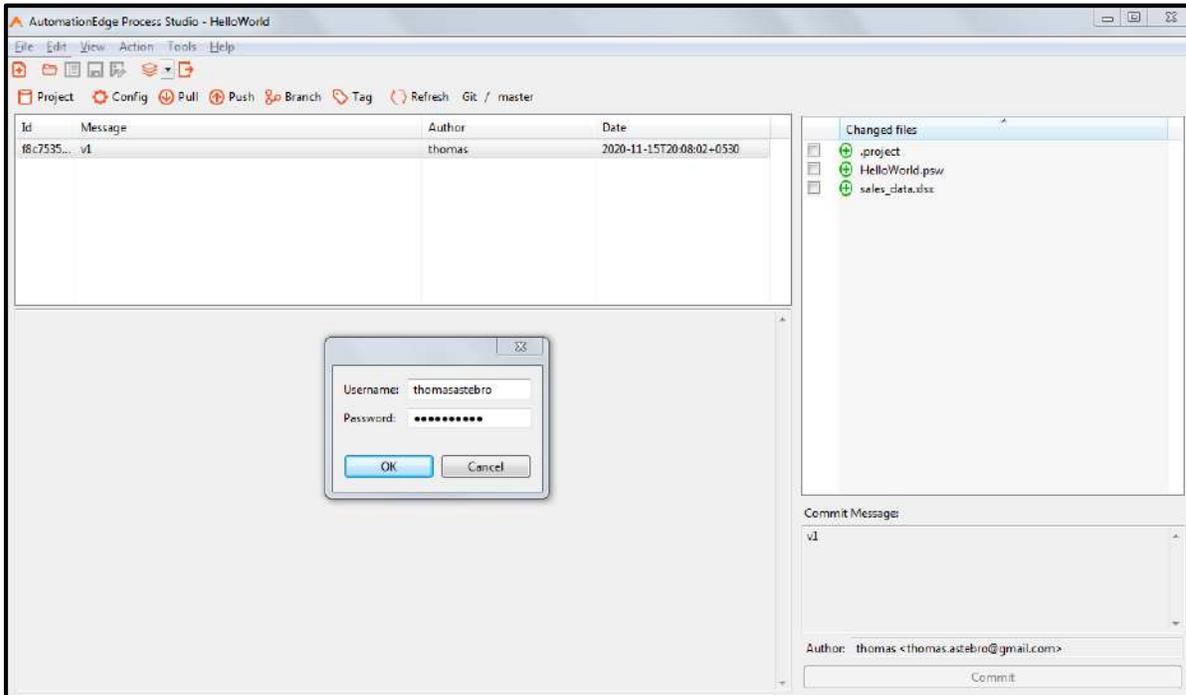


18.3 Push

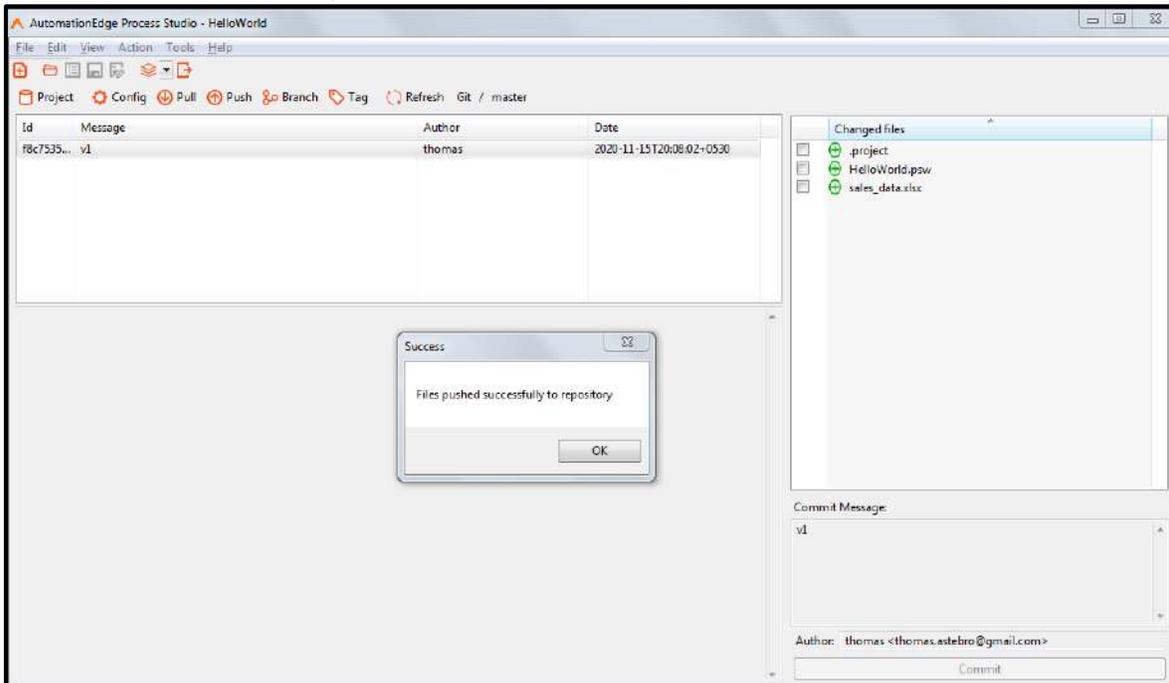
16. We wish to push these changes to the online repository. Before we can do this we need to setup remote repository. Click on Config icon below to set a Remote Repository.
17. Provide the online path of your repository as seen below.



18. It pops up a credentials dialog. Provide your online repository username and password.

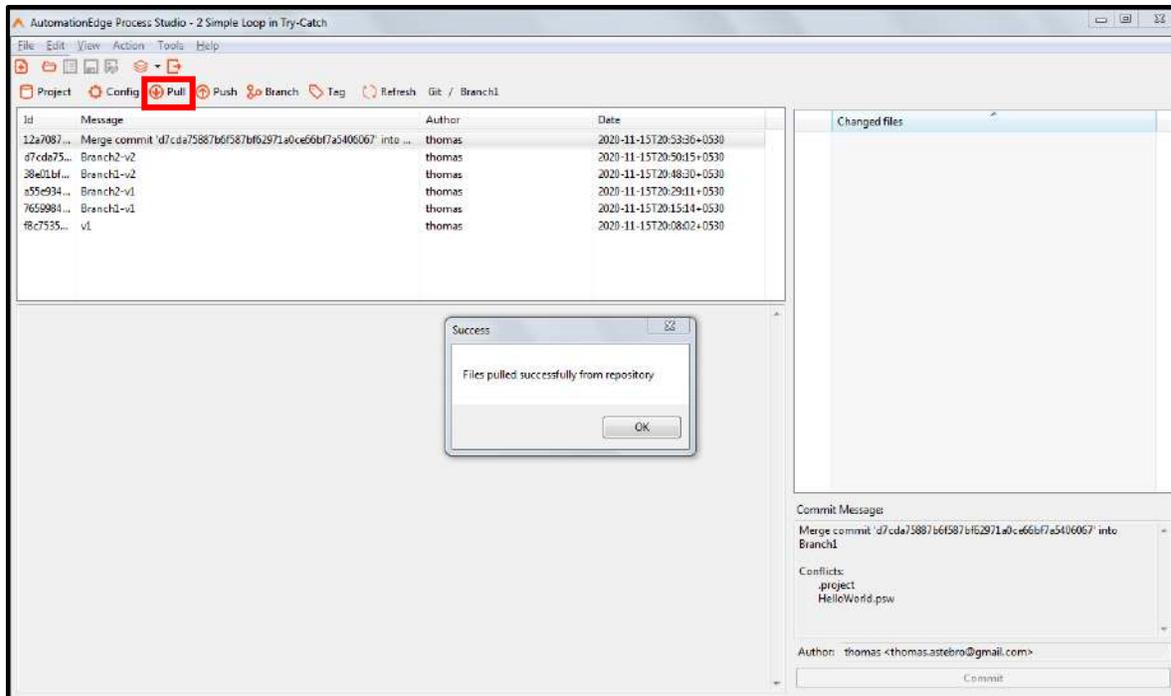


19. A popup appears showing that files were successfully pushed to the repository.



18.4 Pull

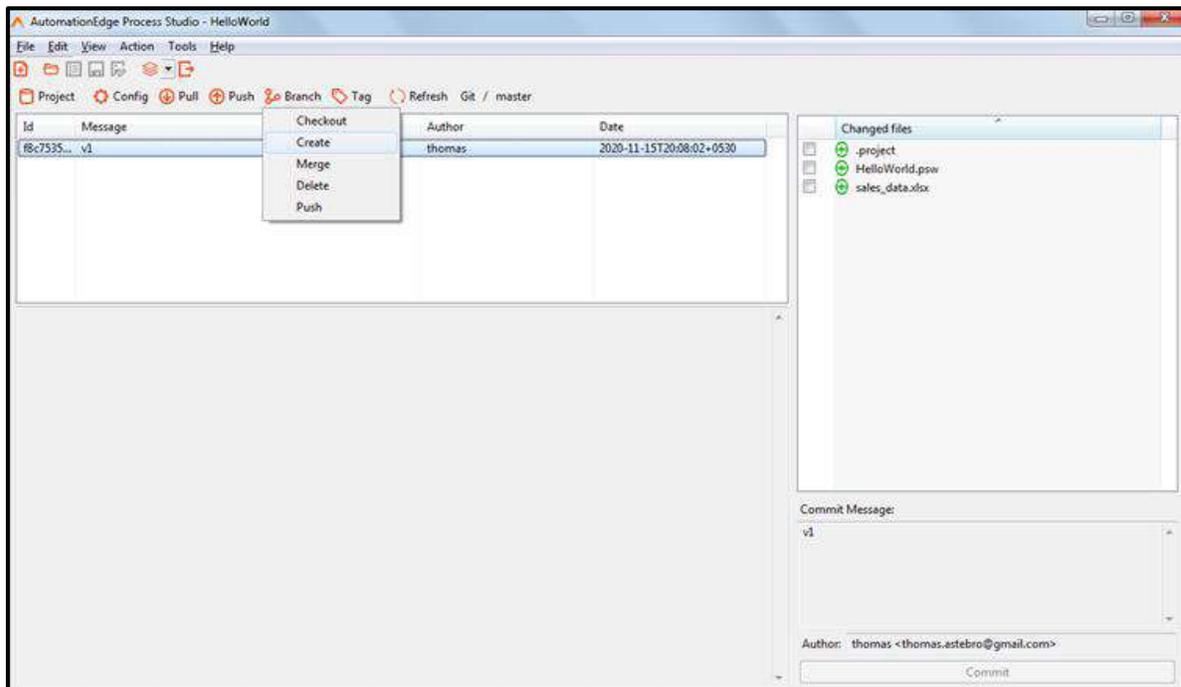
20. We can click the Pull below to pull files from the online repository. The popup below shows files successfully pulled from repository.



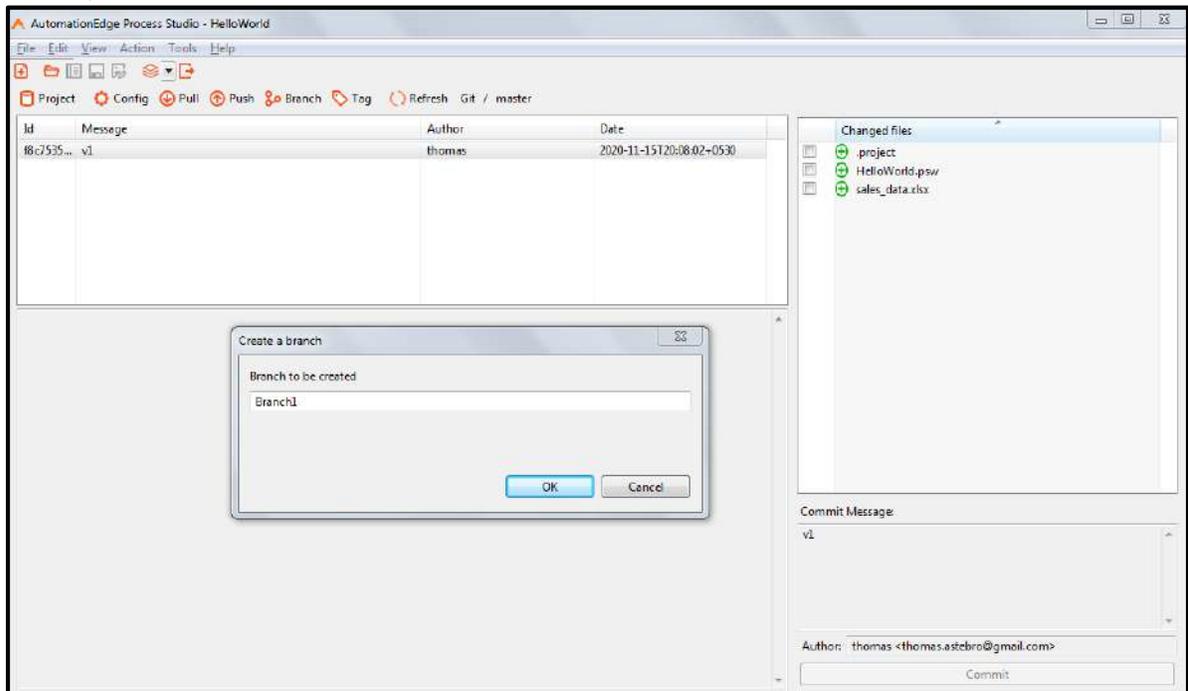
18.5 Branches

You can perform Branch related actions in Process Studio.

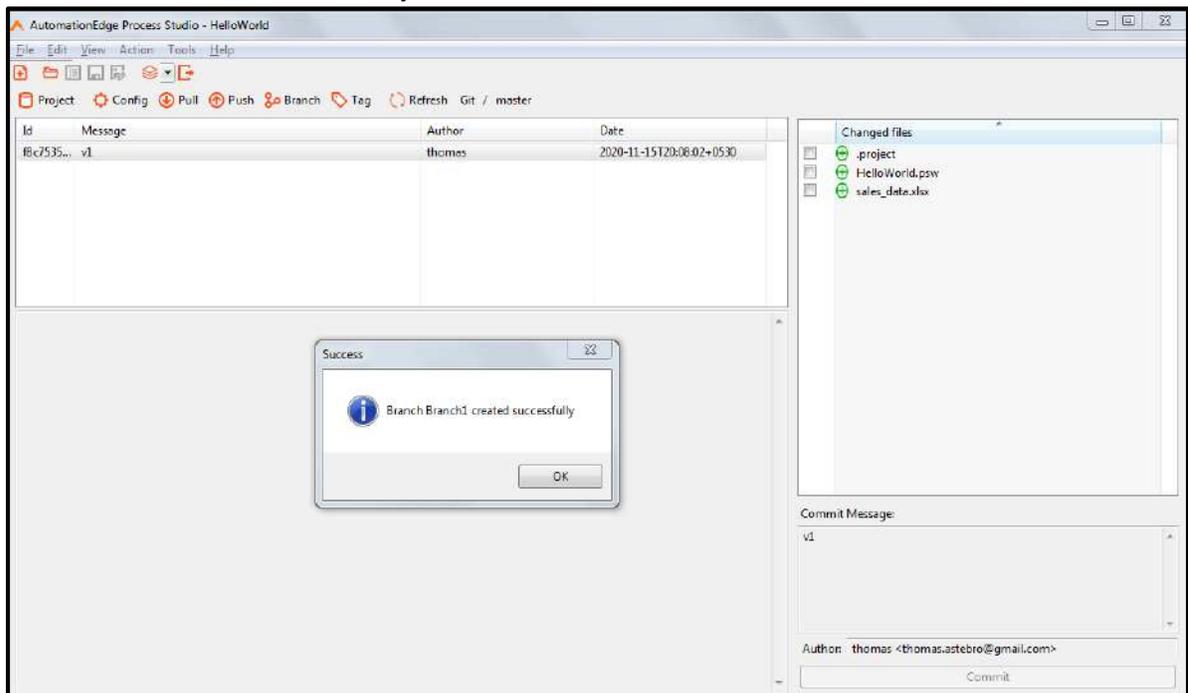
21. Click on Branch to see the options as below.
22. Let us first Create a Branch. Click Create.



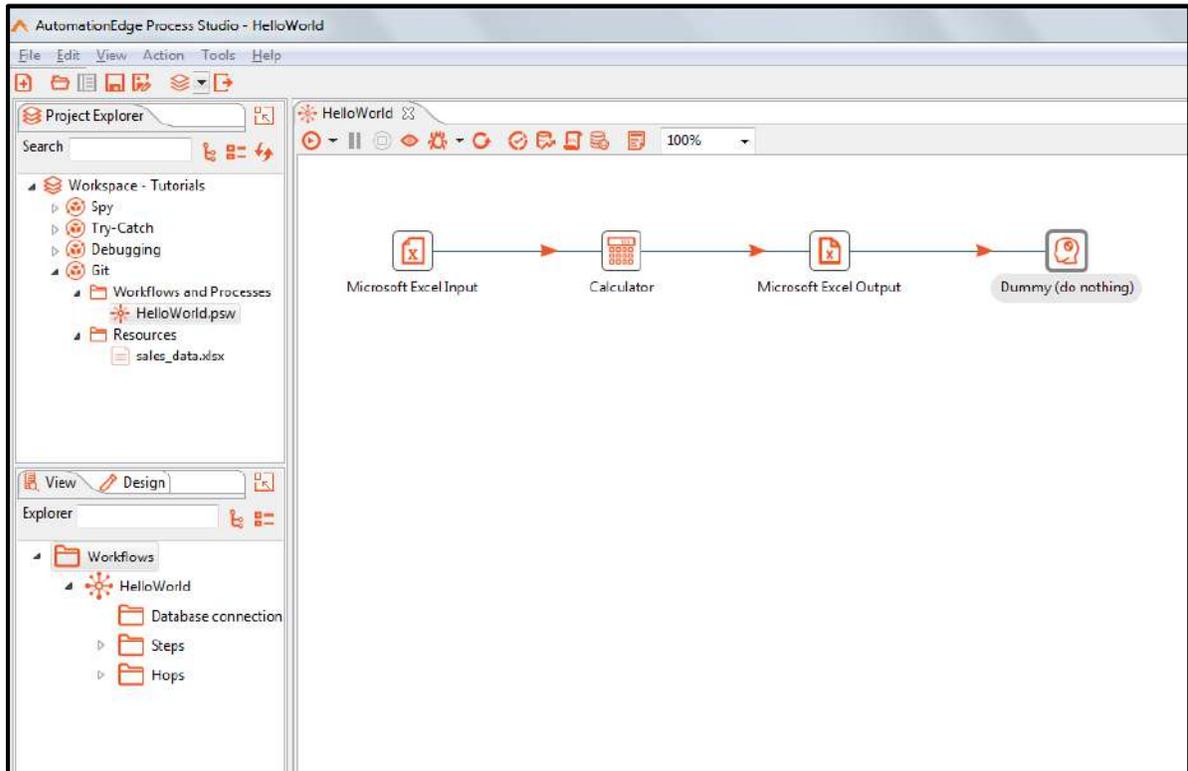
23. Specify a branch name. We specified Branch1.



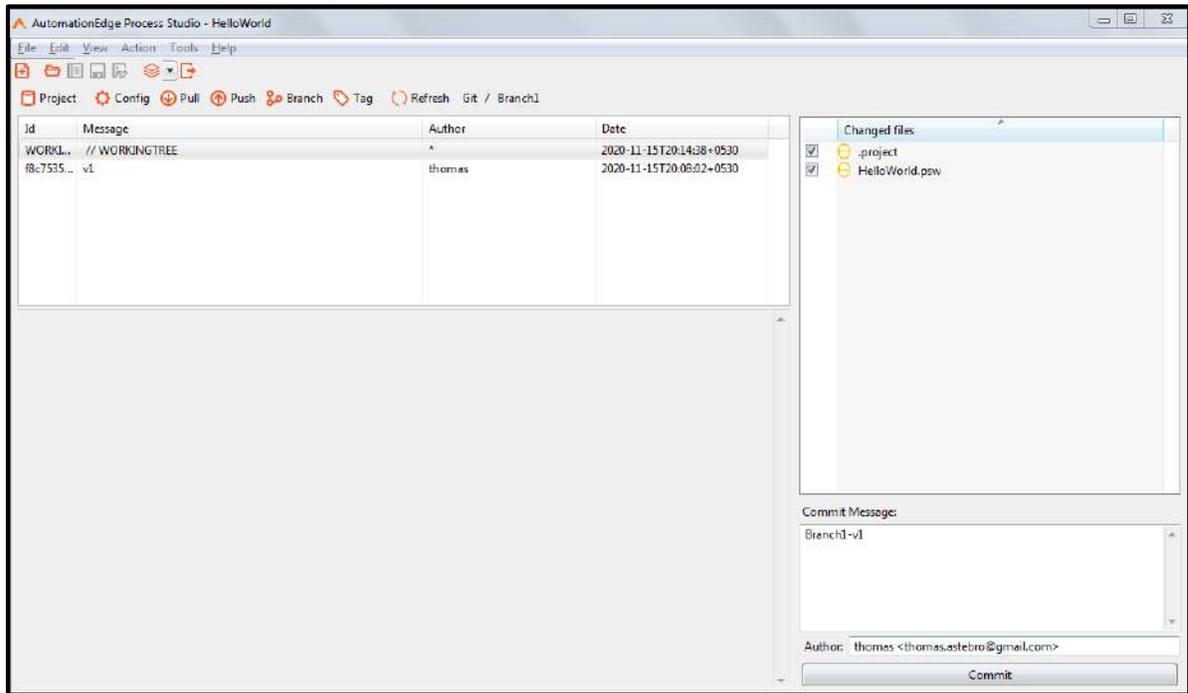
24. Branch1 is created successfully and Branch1 is checked out.



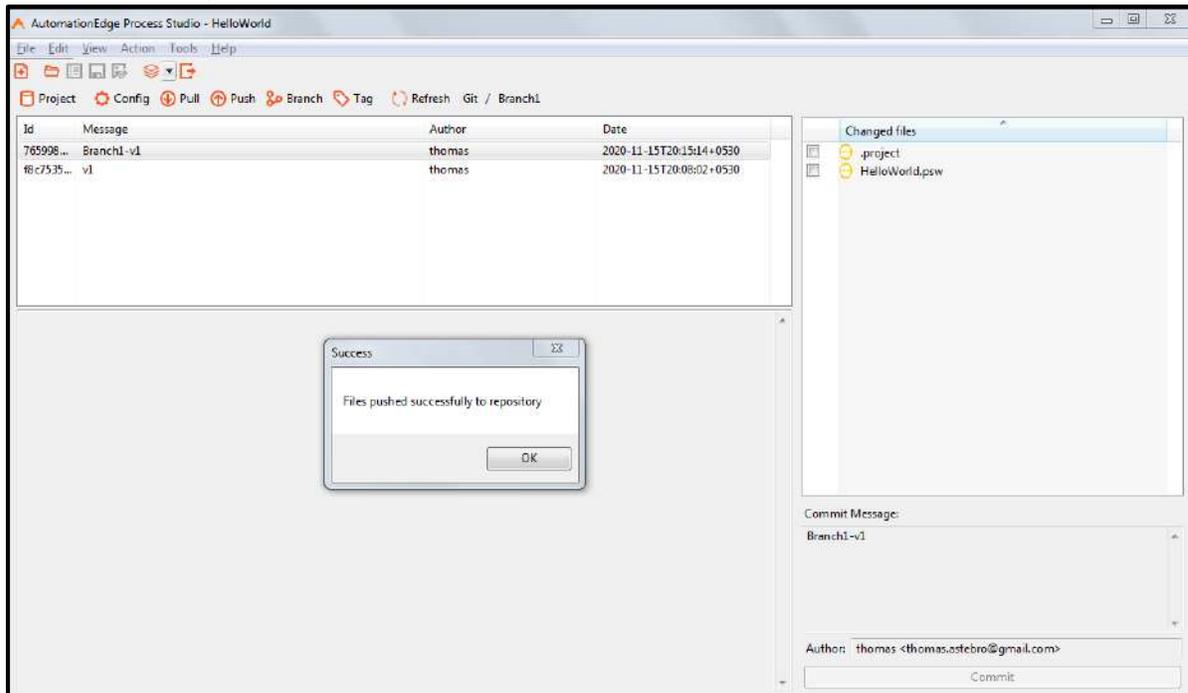
25. Go to the Process Studio perspective and make changes as desired. We have added a step Dummy.



26. Go to the Git/SVN Perspective as seen below. Click Refresh if required. You can see the working Tree for Branch1 and the changed files on the right pane.
27. Provide a commit message and click Commit.



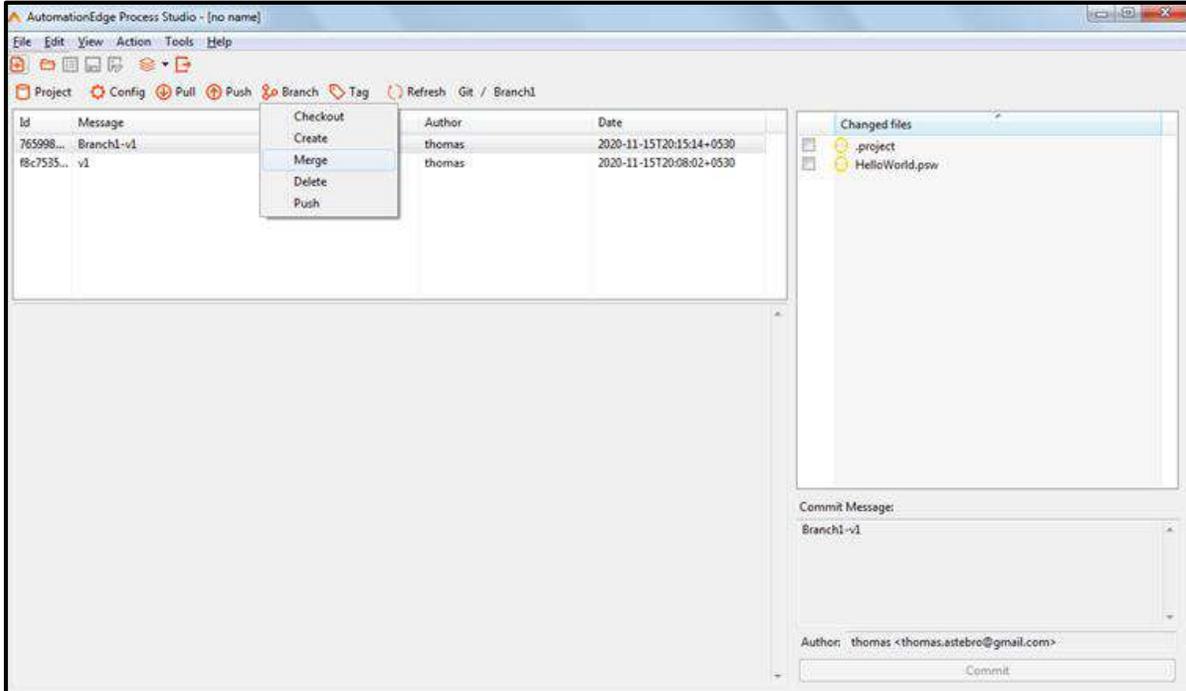
28. Now select your commit point (Branch1-v1) and click Push. Files are successfully pushed to the online repository as seen below.



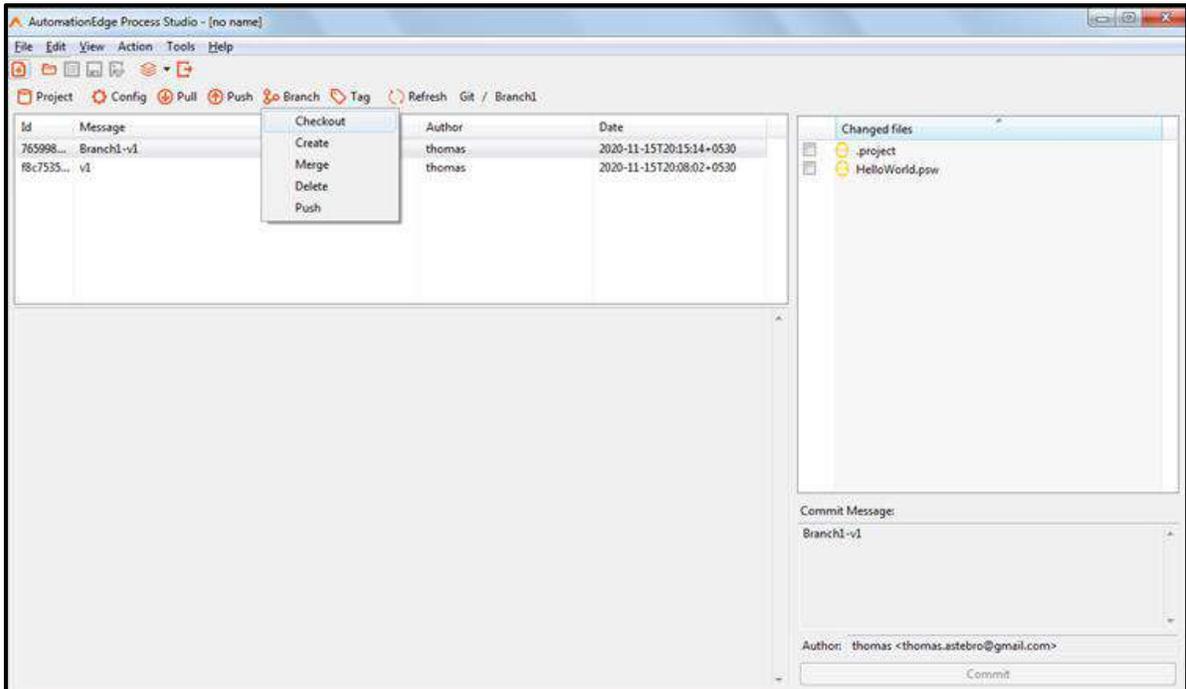
18.6 Merge Scenarios

18.6.1 Merge without Conflict

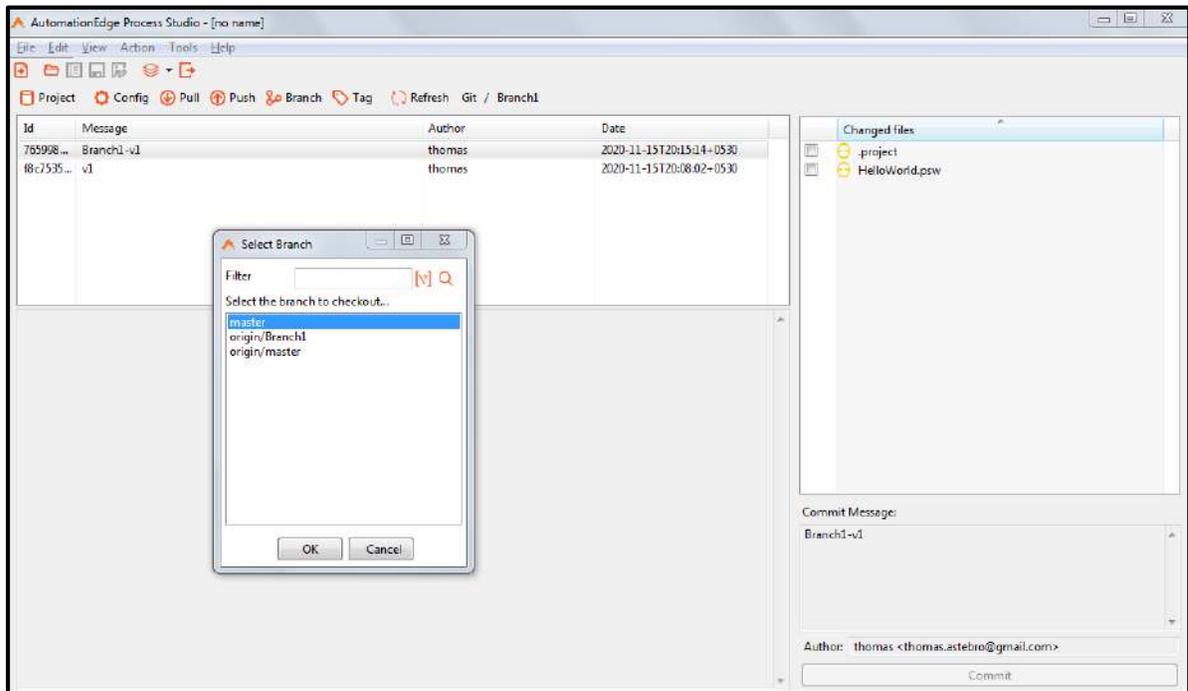
1. Click on Branch and you have various option as seen below.



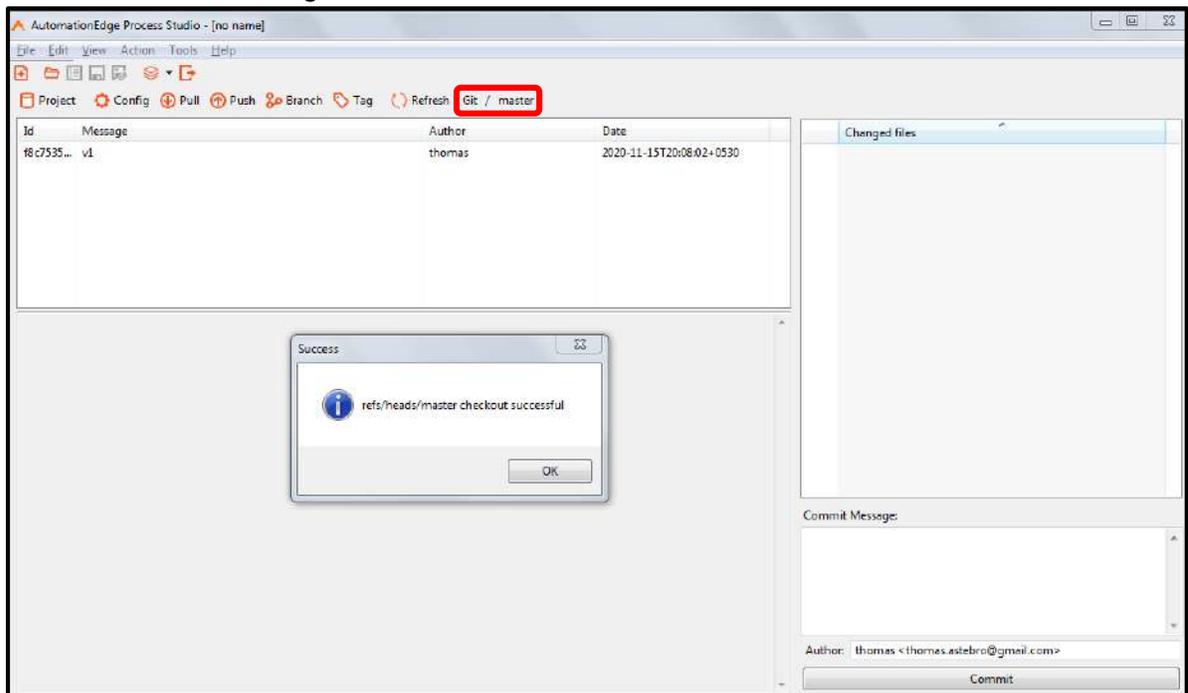
2. We wish to merge Branch1 into the master branch. We will begin with Checkout of master branch. Click on Branch and Checkout option as seen below.



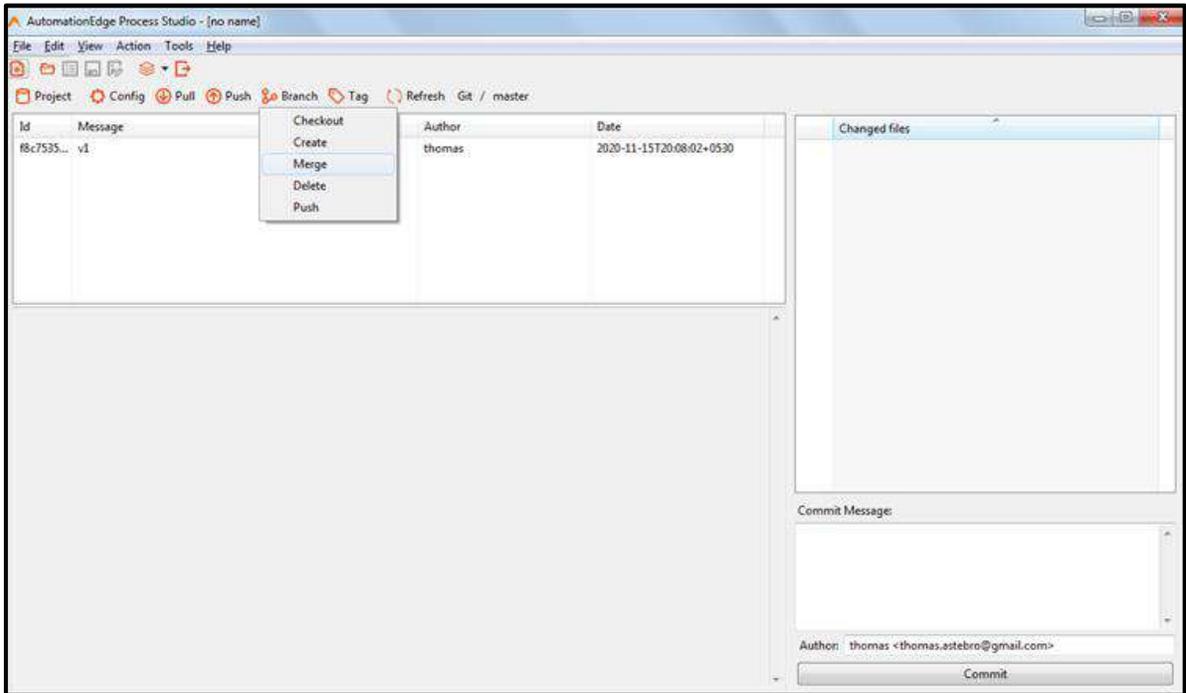
3. Select master as the checkout branch.



4. We can see the message that master checkout is successful.

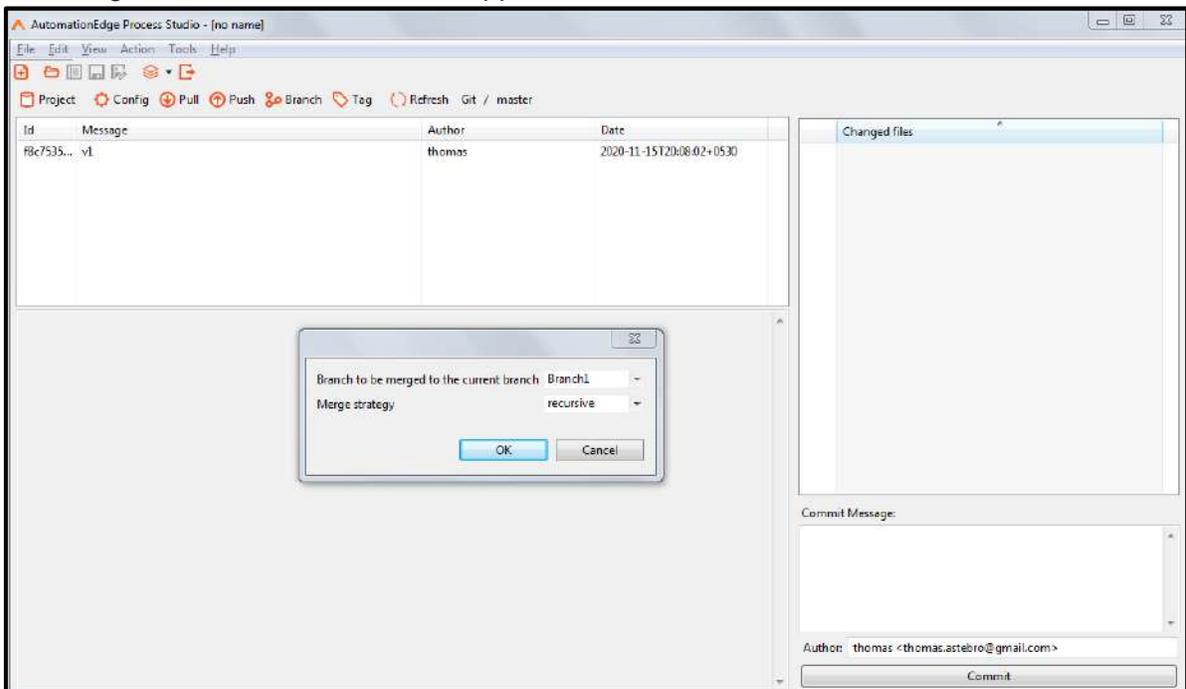


5. Now under Branch, click Merge.

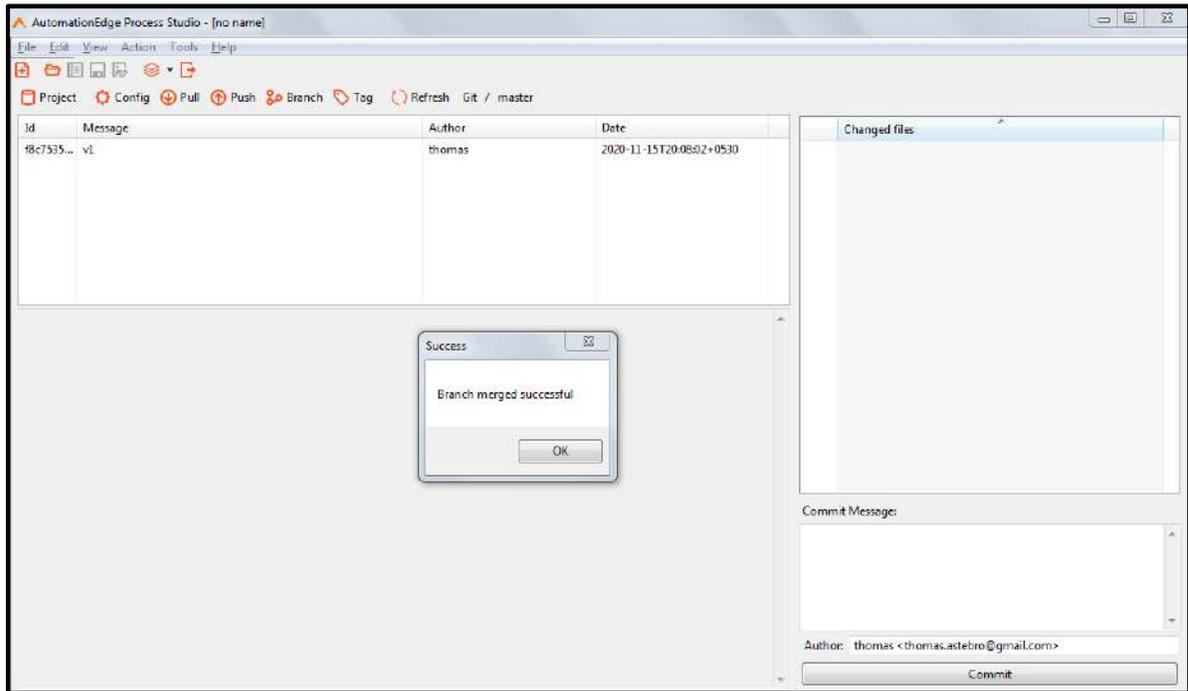


6. In the 'Branch to be merged to current branch' select Branch1 and merge strategy as **Recursive**.

7. Merge Strategy options are 'ours' and 'theirs'. **Ours** option forces conflicts to be auto-resolved by favouring our version. **Theirs** is the opposite of Ours which favours their version.

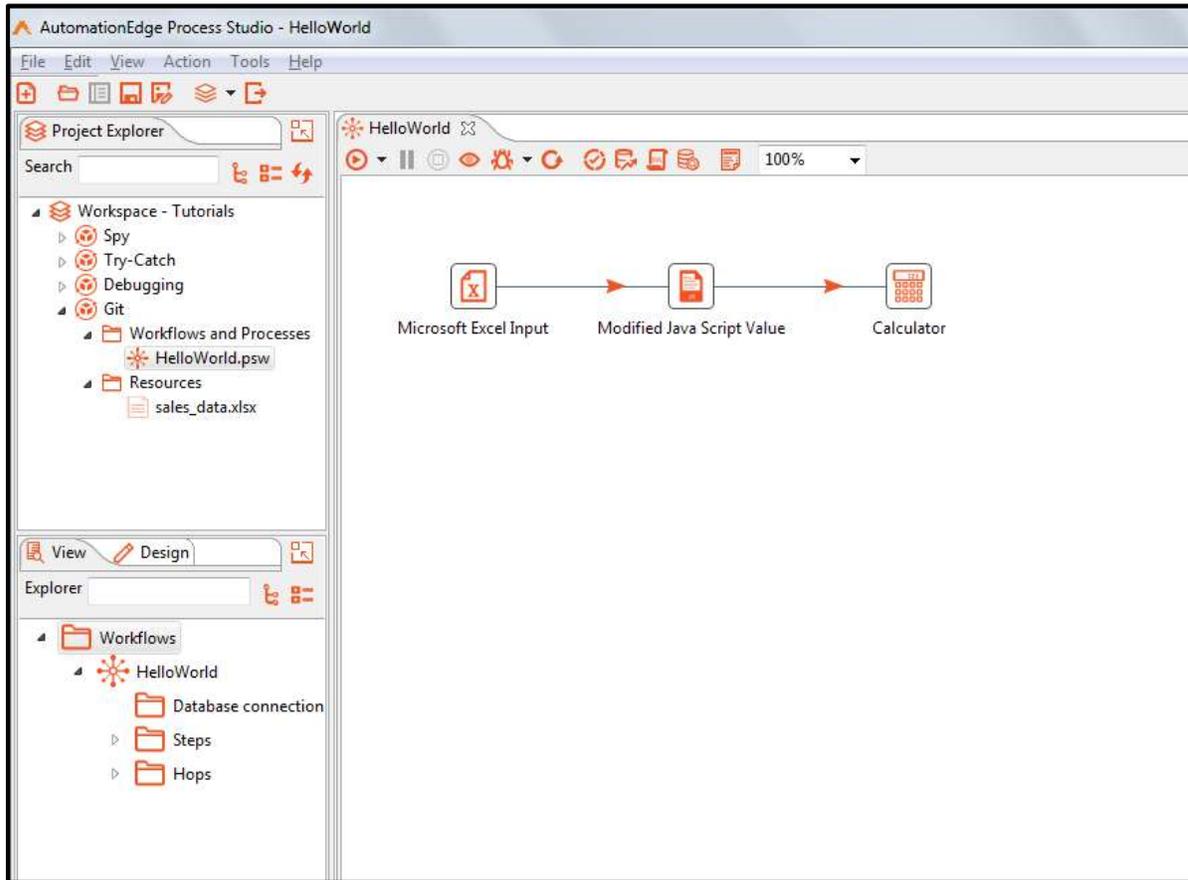


8. Branch merged successful option is seen below. You will now be able to see changes in the mater as well.

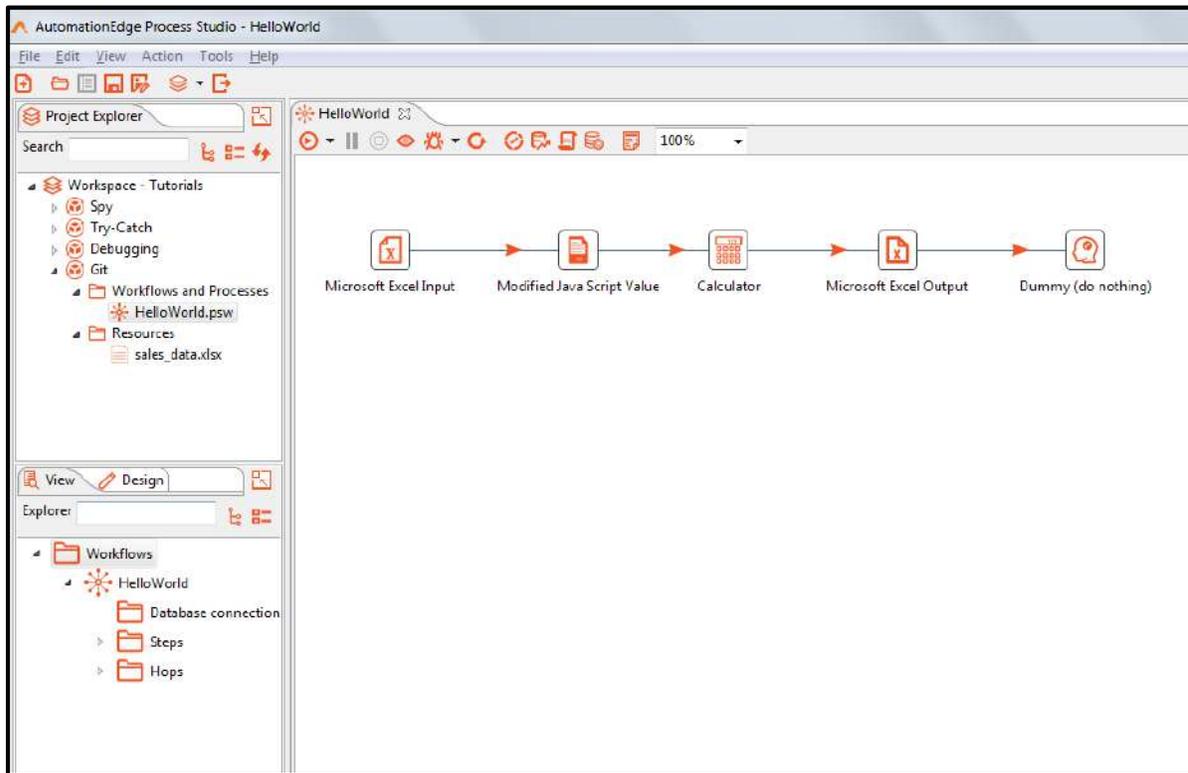


18.6.2 Merge having Conflict

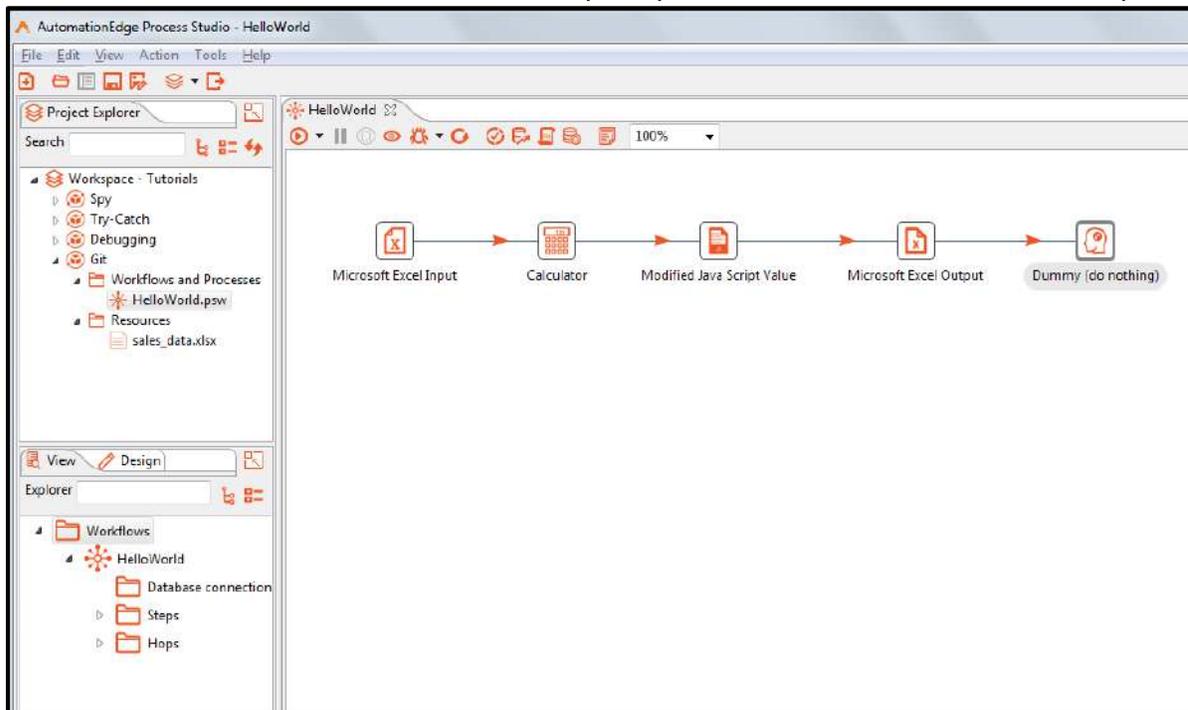
9. Now let us start with the master branch having a workflow as below.



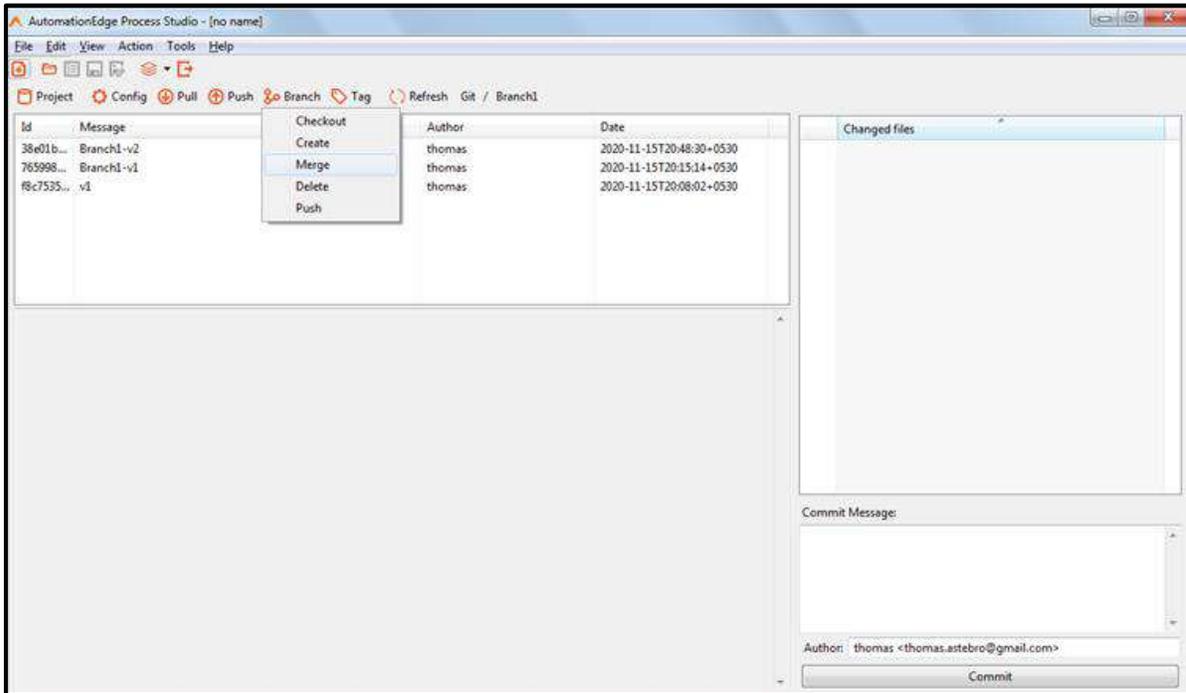
10. Switch to Branch1 and add a Modified Java Script step as seen below before Calculator step.



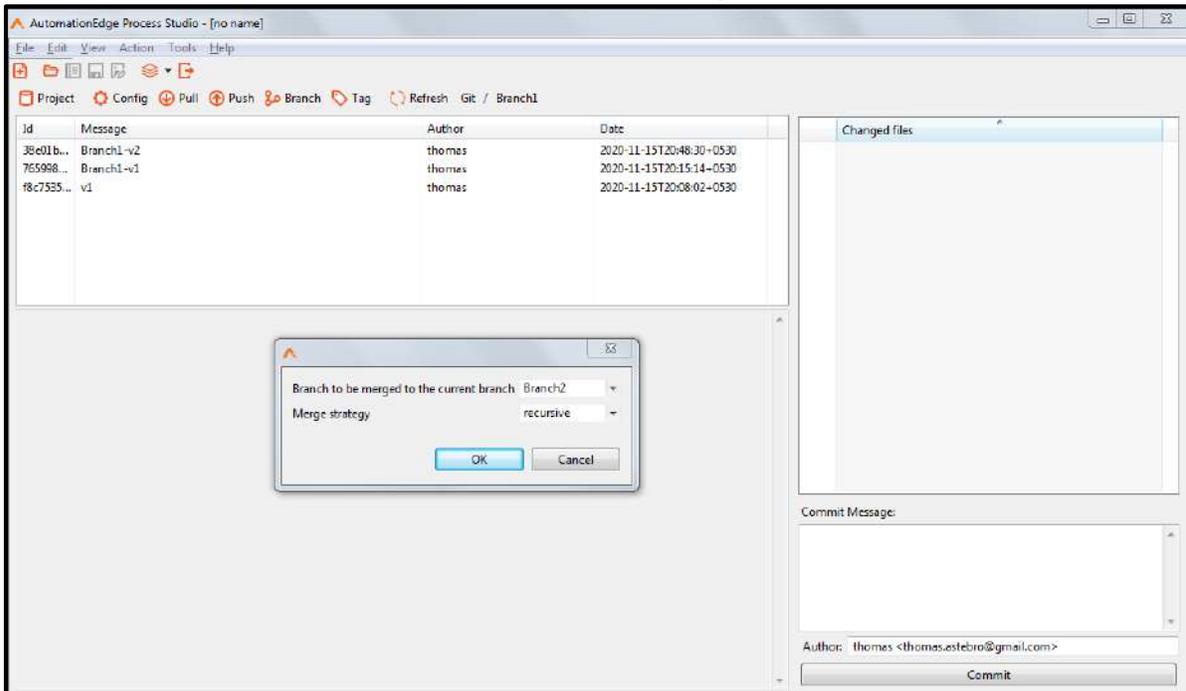
11. Checkout Branch2. Add a Modified Java Script step as seen below after Calculator step.



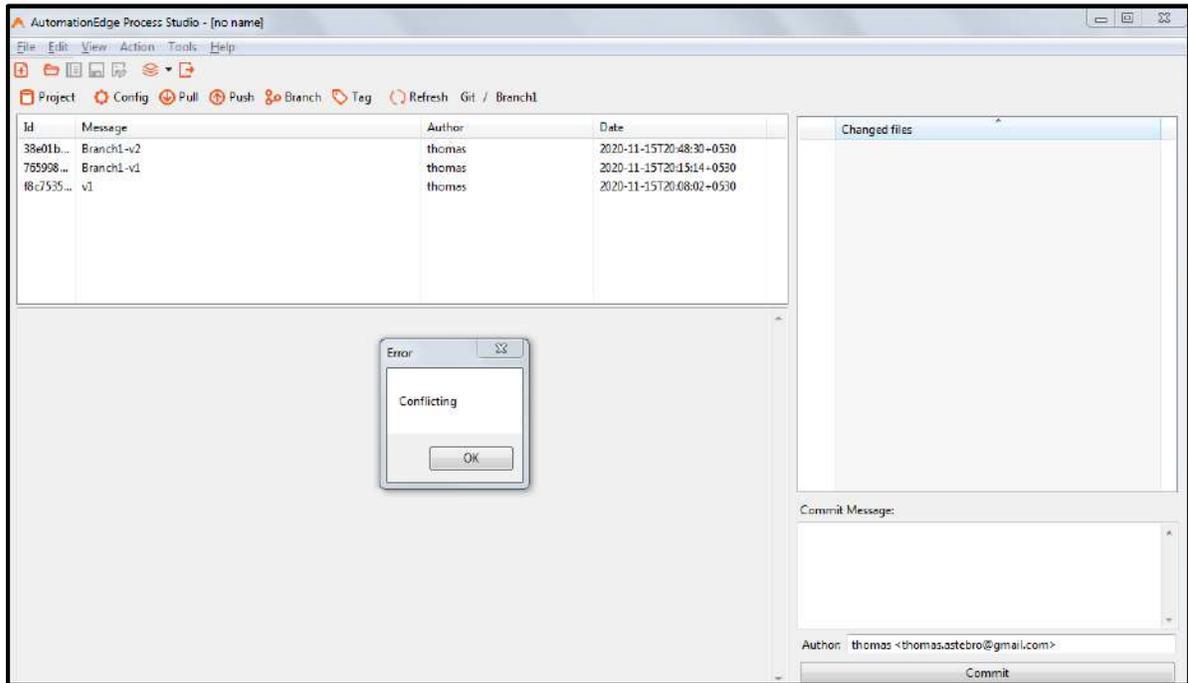
12. Let us merge the branches. Select Merge.



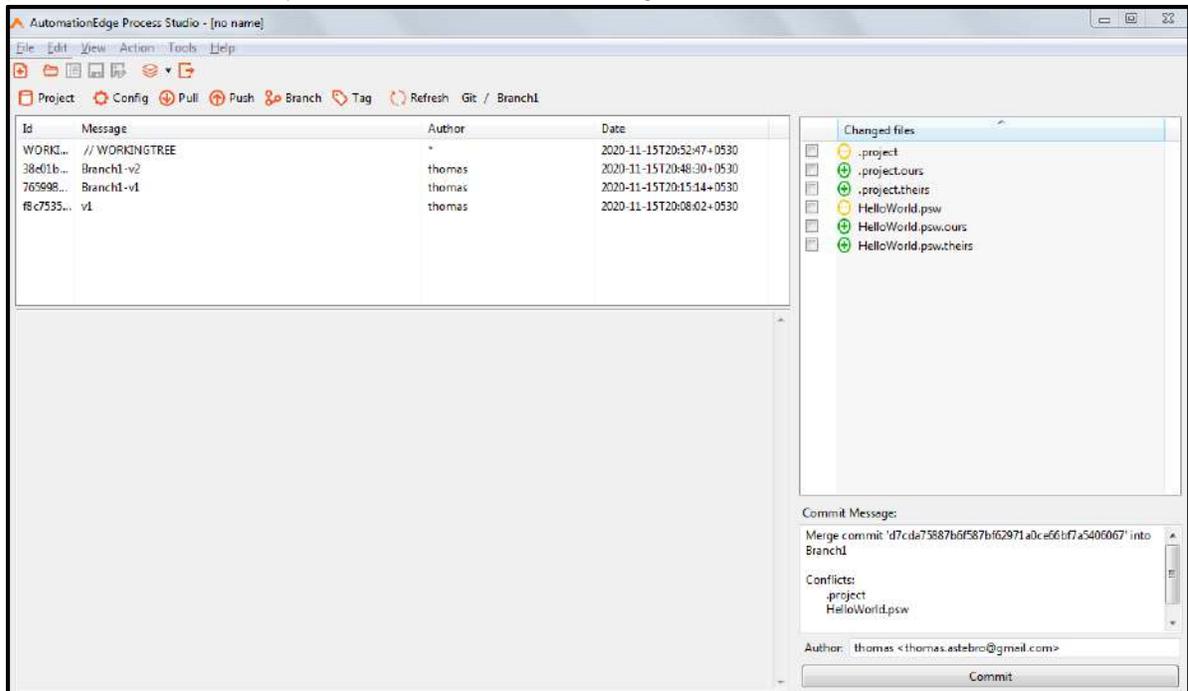
13. Currently, we are on Branch1. Select Branch to be merged to the current branch as Branch2.



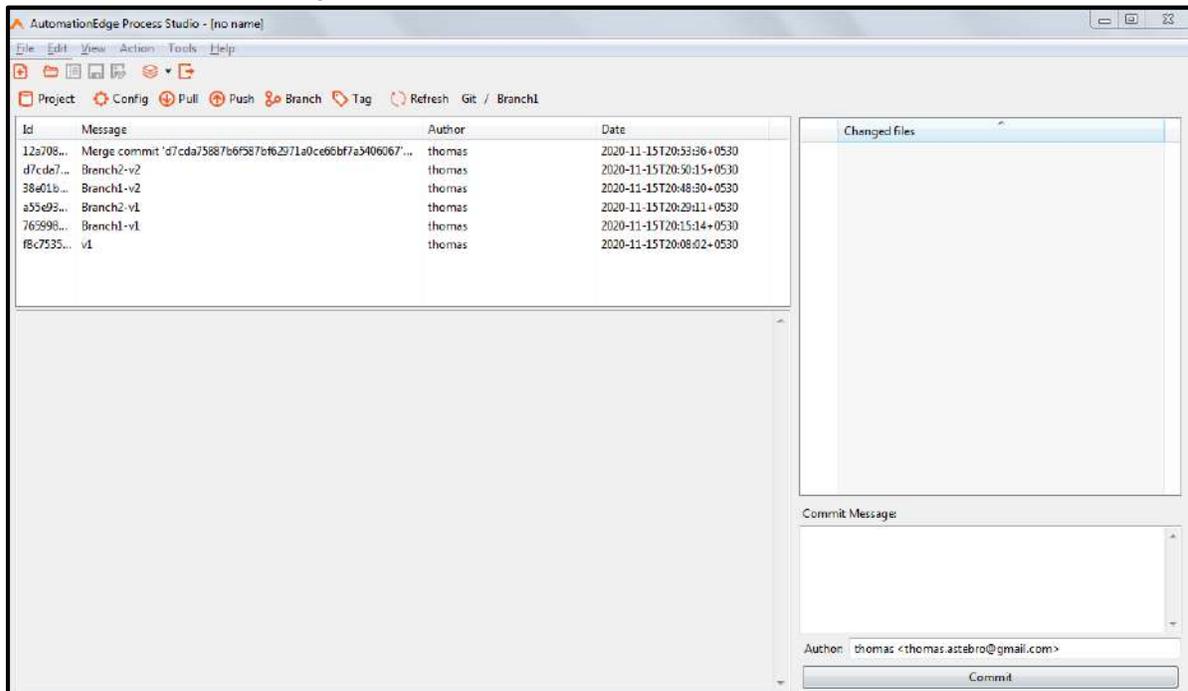
14. An Error message is shown showing these are Conflicting branches. Click Refresh.



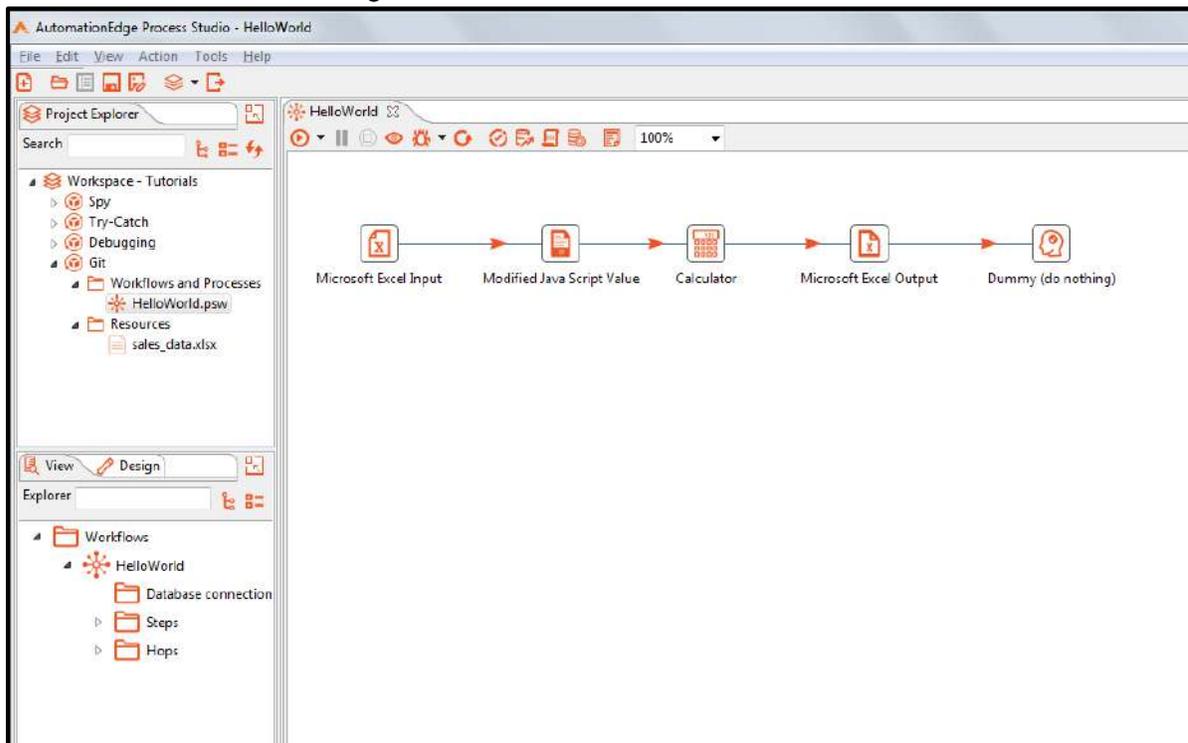
15. The WORKINGTREE is seen below. The files in the two branches are seen in the Changed Files pane. Select the files you wish to have in the merged branch.



16. You can now see a Merge commit.



17. 'Ours' branch has been merged as can be seen from the workflow below.

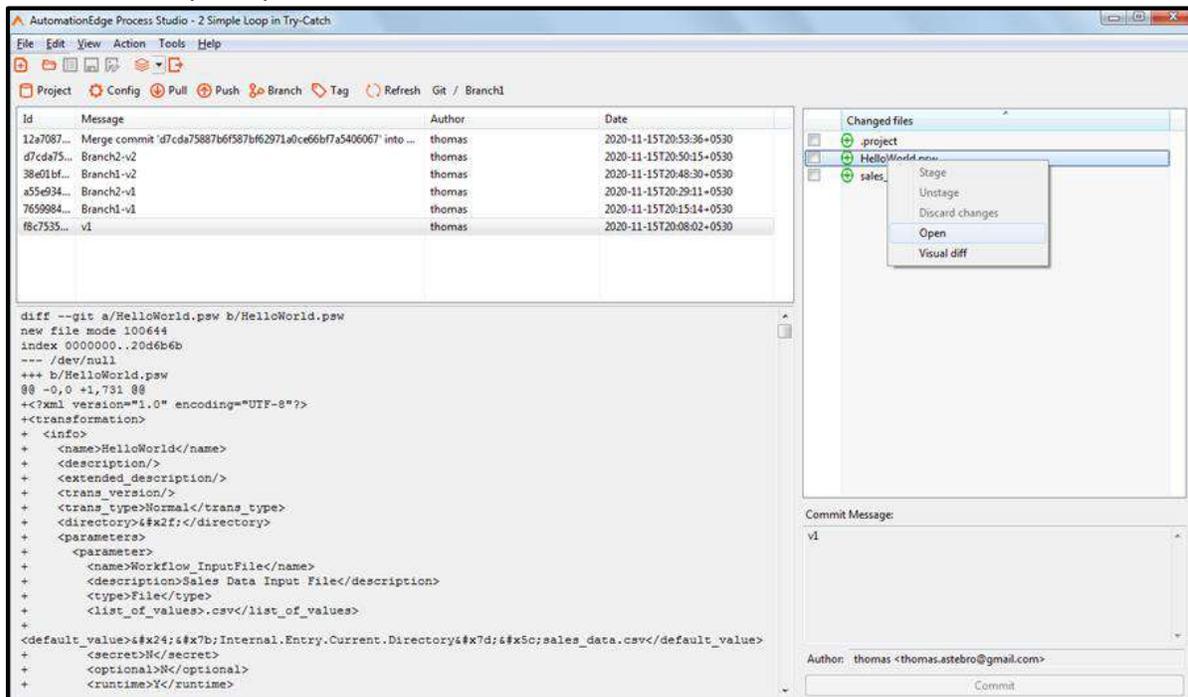


18. These completes Branches. Tags can be used similar to Branches but without merge options.

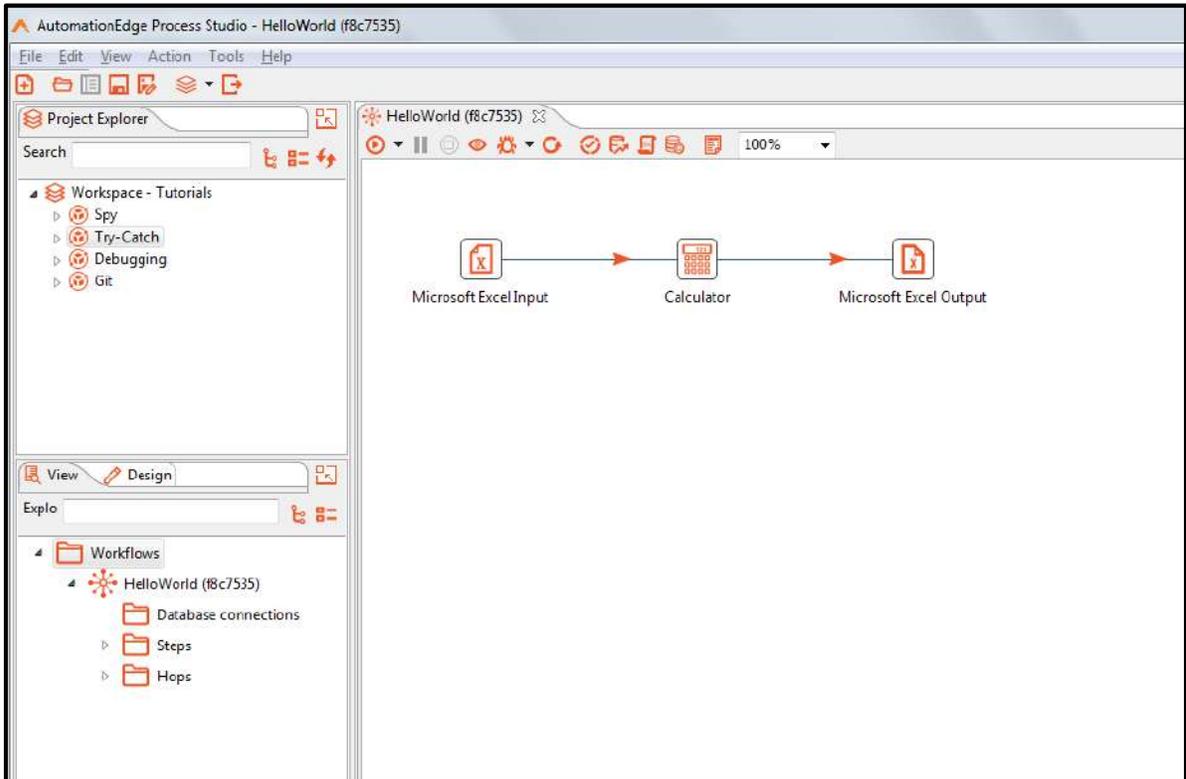
18.7 Visual Diff

You can open workflow files or see a visual difference from the origin at any commit point. Following are the steps to open or see a visual difference of workflow files.

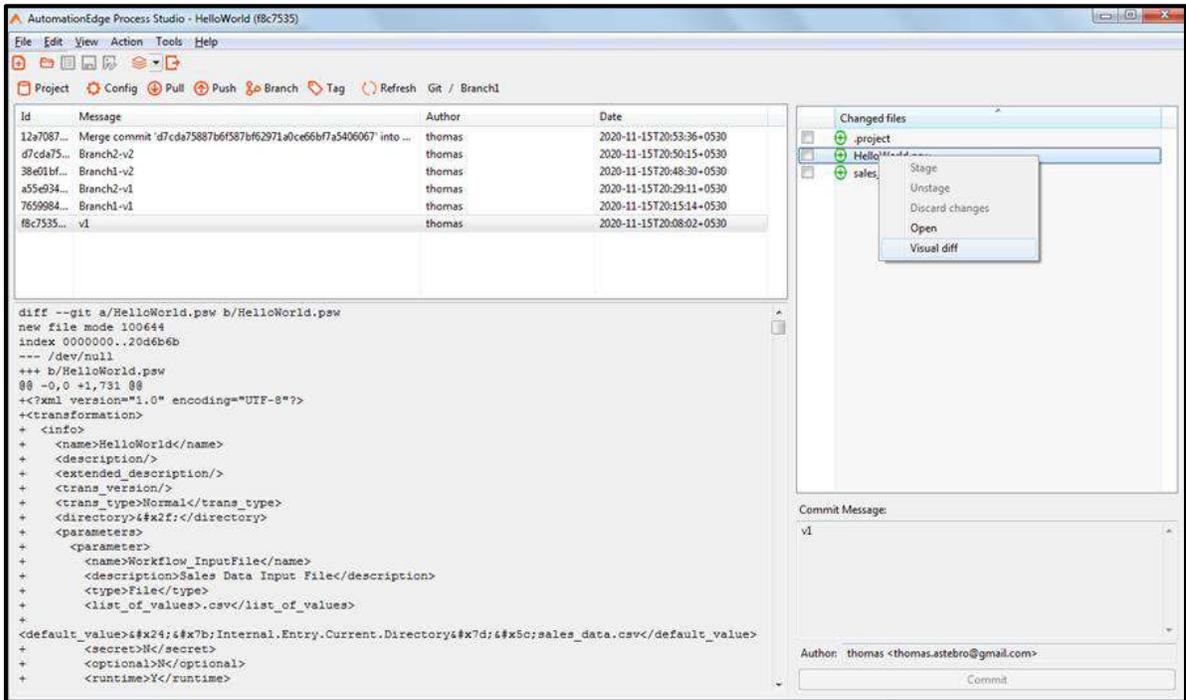
19. Click on a commit point and you can see the Changed files pane on the right hand side. Green icon indicates new file and yellow icon indicates changed file. Right click on any workflow file to open the committed version of it or so see the visual difference from the earlier commit.
20. Click on the Open option first.



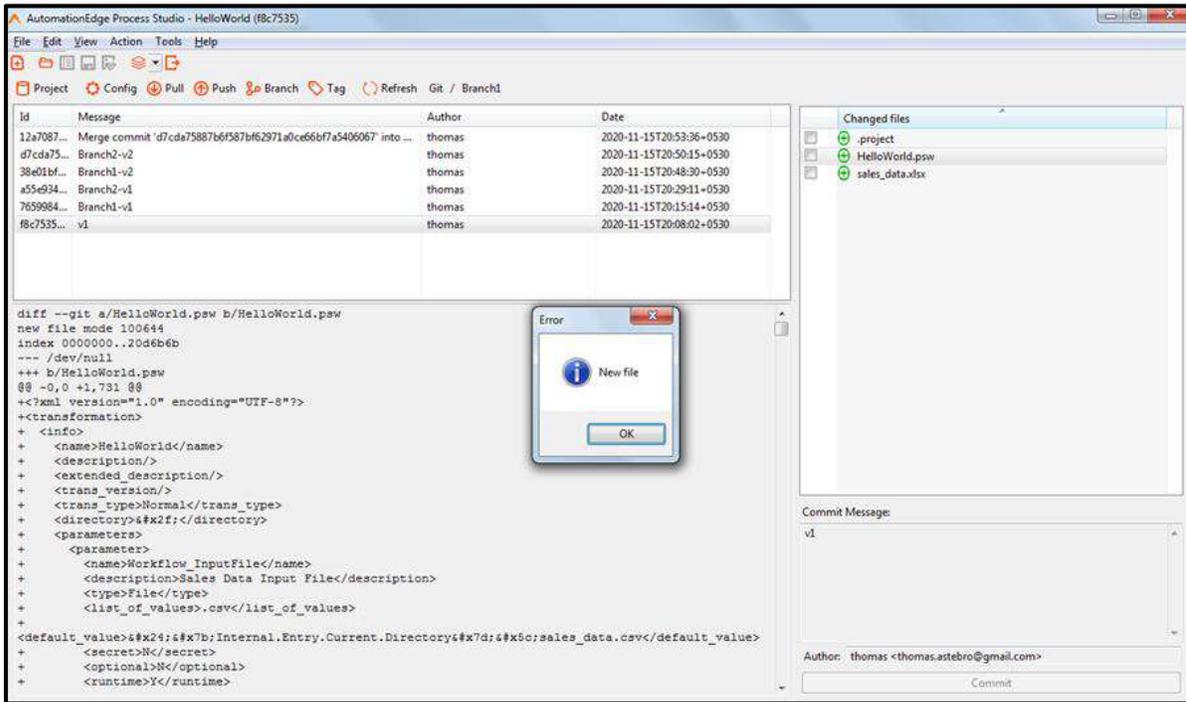
21. HelloWorld workflow file is opened as seen below.



22. Next click on Visual Diff.

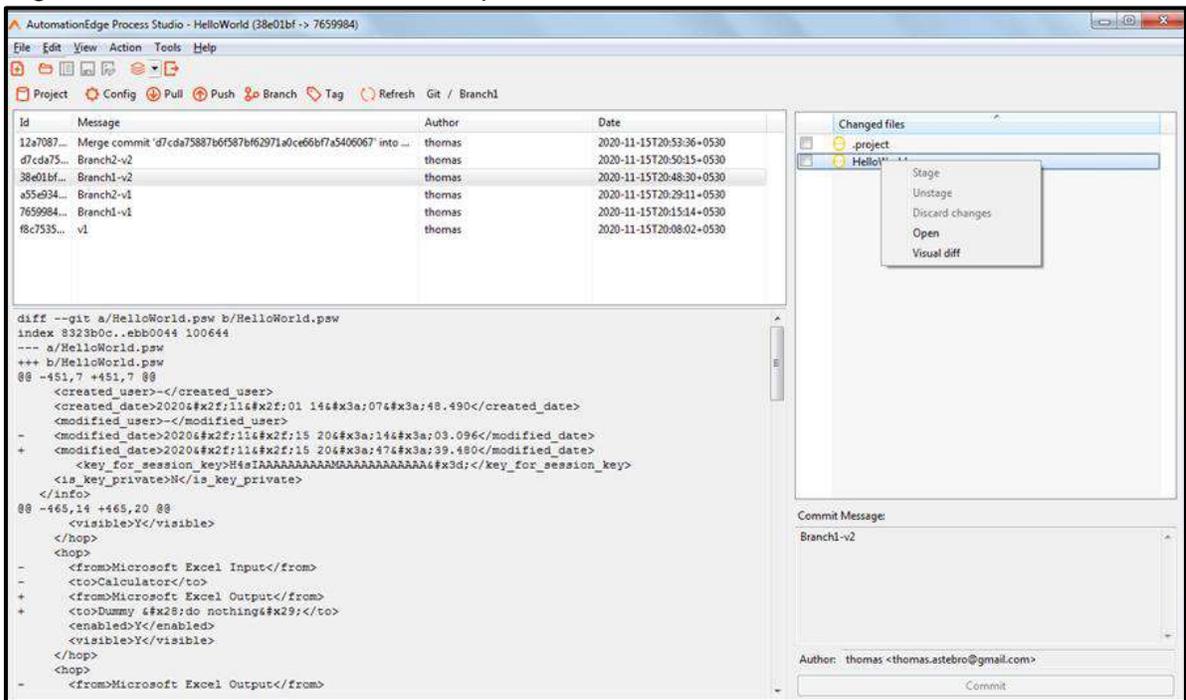


23. Since this is a new file there is no Visual Diff available and a popup message appears indicating it is a new file.

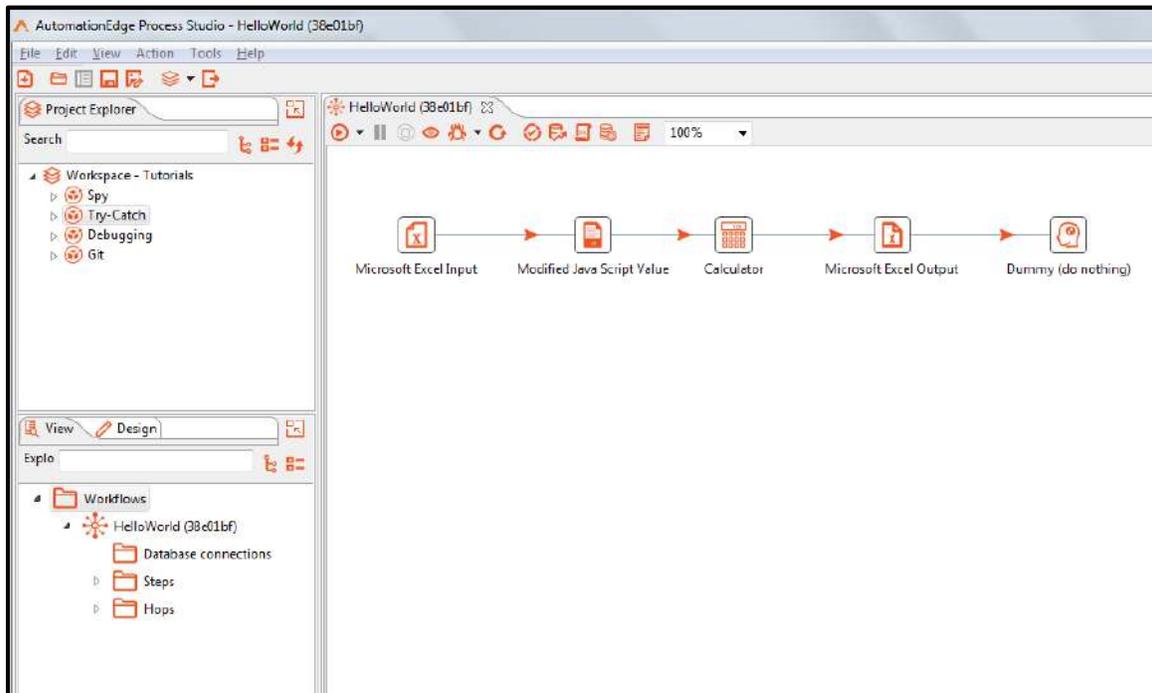


24. Next let us go to another commit point Branch1-v2.

25. Right click a workflow file and click Open.

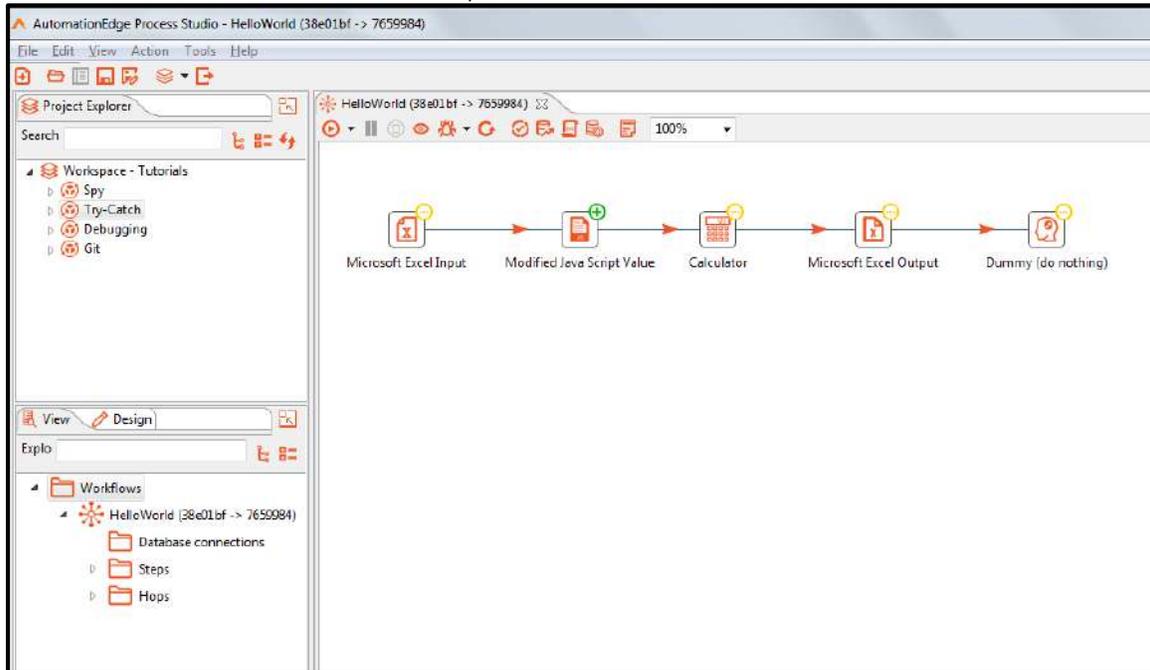


26. The workflow file at this commit point is opened as seen below.



27. Next click on Visual Diff to see the following workflow. It shows new steps marked with green circles and changed steps marked with yellow circles. Any deleted steps would be marked with red circles.

- When any one commit is selected, the diff is between the selected commit and its first parent commit.
- When multiple commits are selected, the diff is between the newest commit and the oldest commit.
- When no commit is selected, it is assumed that WORKINGTREE is selected.



In this section we have demonstrated how to use Git with Process Studio to manage your workflows. We have covered the features available to manage Process Studio workflows and Git Repository.

19 Project 12: SVN Repository Integration

In this section we shall create a project called SVN. We will demonstrate Process Studio integration with SVN Repository and perform the following Subversion Controls using Process Studio.

- Create SVN Repository
- Commit, Update (Pull), Push is disabled because making a commit always pushes changes to the remote repository.
- Visual Diff
- Branch

Others controls such as Merge and Conflict for SVN will be implemented in upcoming releases.

It is assumed that you have access to a remote SVN repository.
Let us begin the first section with downloading and installing SVN Client.

19.1 Download and Install Subversion (SVN) Client

19.1.1 SlikSVN

Install SlikSVN Windows Command Line Client to fetch a library file to be copied to Process Studio.

- SlikSVN on Windows:

Download SlikSVN Client (“Slik-Subversion-1.9.7-x64”) - from (<https://sliksvn.com/download/> or <https://sliksvn.com/pub/>) and install it.

Copy C:\Program Files\SlikSvn\bin\libsvnjavahl-1.dll to Process Studio path libswt\win64 (64-bit) or to libswt\win32 (32-bit)

- SlikSVN on Linux (Debian/Ubuntu):

```
$ sudo apt-get install libsvn-java
```

Append /usr/lib/x86_64-linux-gnu/jni/ to LIBPATH in process-studio.sh as follows:

```
LIBPATH=$LIBPATH:/usr/lib/x86_64-linux-gnu/jni/  
export LIBPATH
```

For more information on how to install libsvn-java 1.9.X to Ubuntu 14.04 refer - <https://tecadmin.net/install-subversion-1-9-on-ubuntu/>

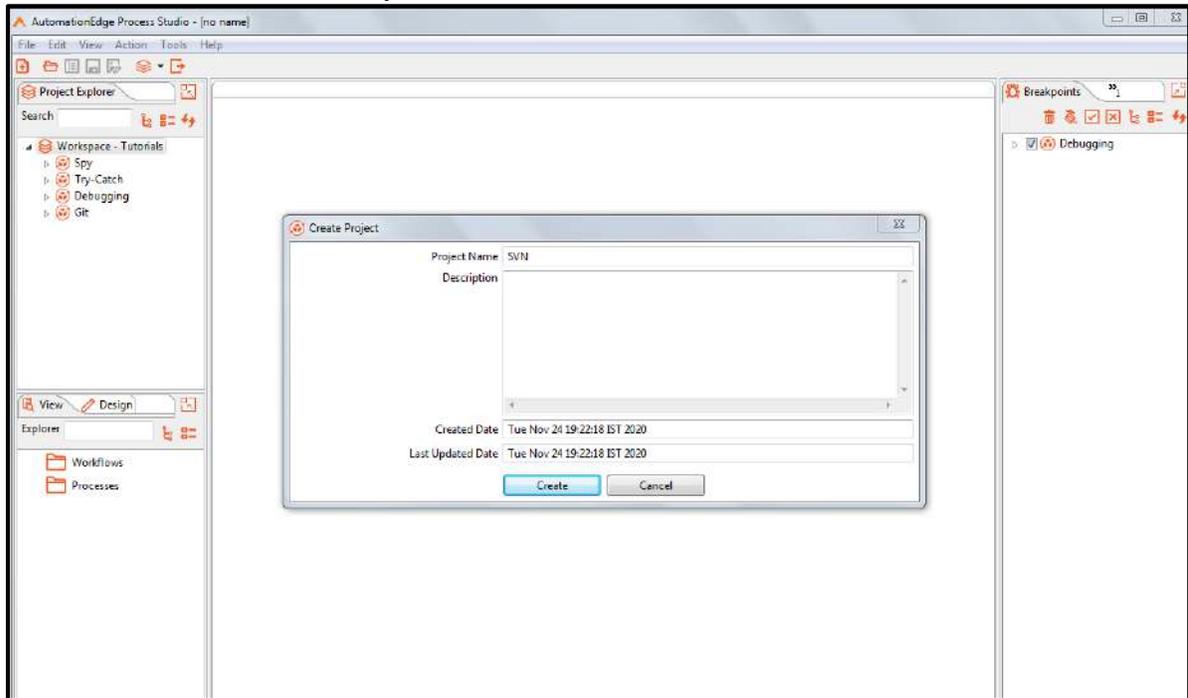
19.1.2 TortoiseSVN

For our exercise we shall download and install TortoiseSVN Subversion (SVN) Client which gives a friendly user interface for SubVersion Control.

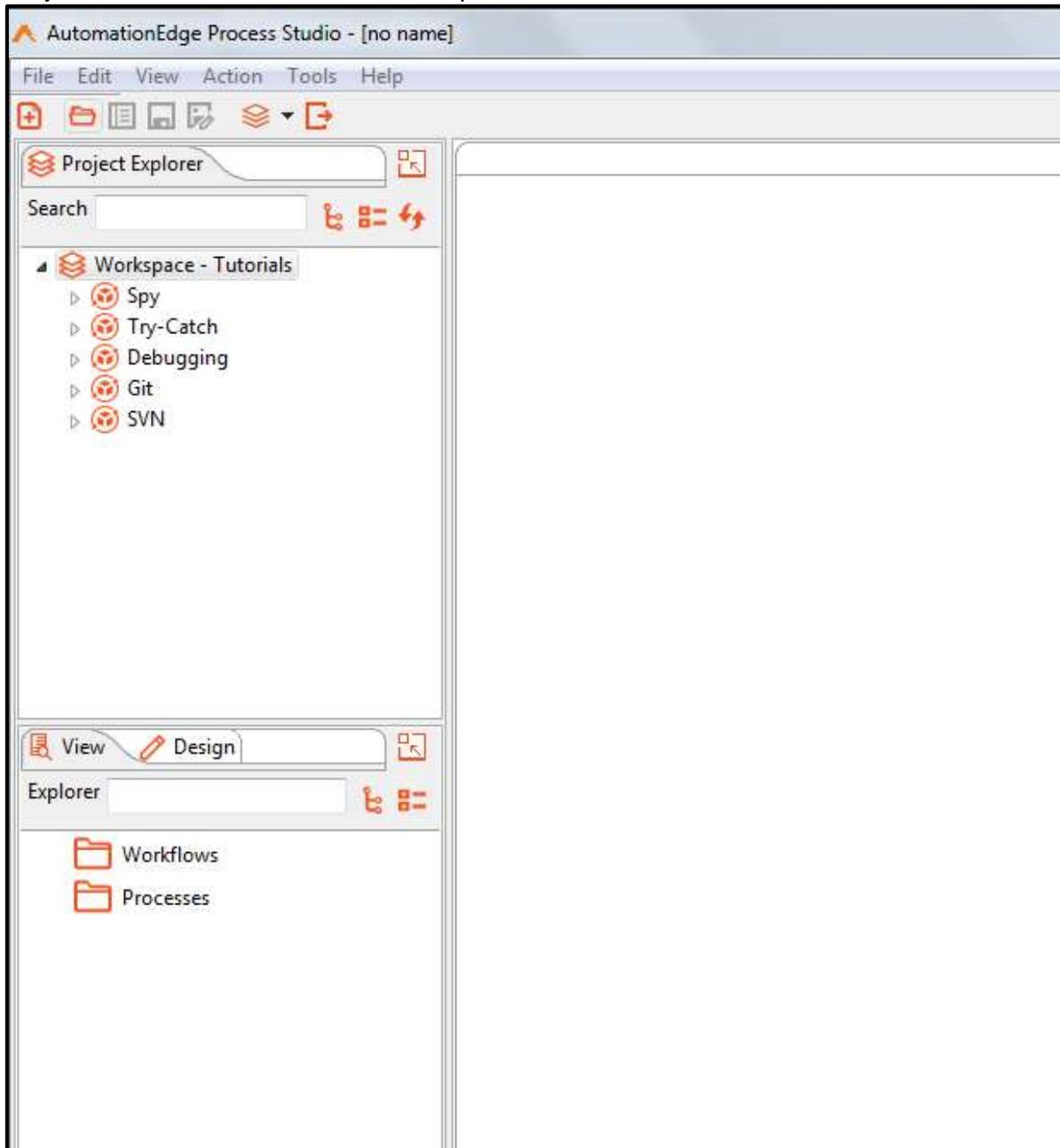
Download TortoiseSVN (TortoiseSVN-1.14.0.28885-x64-svn-1.14.0 or the version suitable for your Windows Operating System) from <https://tortoisesvn.net/downloads.html>.

19.2 Create Project and Checkout SVN repository

1. Launch Process Studio. Here we have opened the Tutorials Workspace.
2. We will create a Project to be synchronized with an SVN repository.
3. Here we have named the Project SVN.



4. Project SVN is now visible in the Workspace.

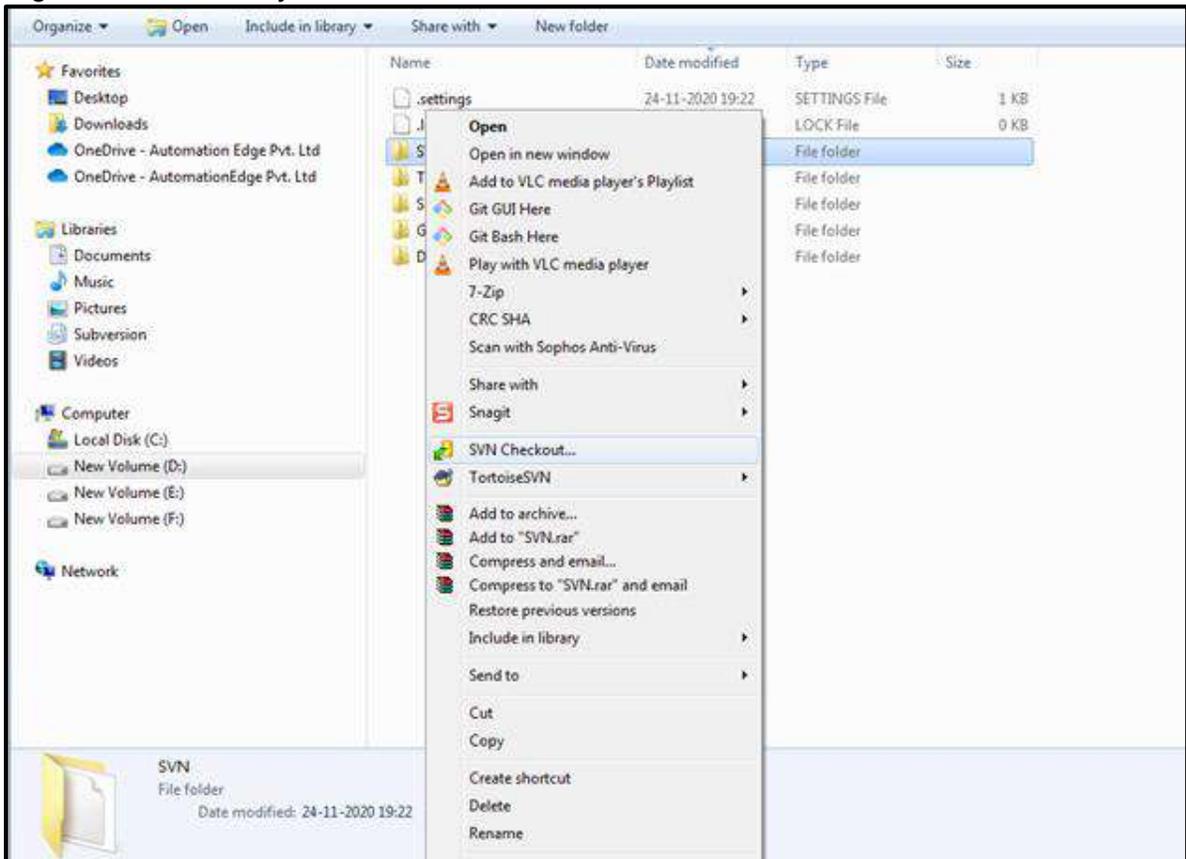


5. This is the Workspace directory where you can also see the project directory SVN.

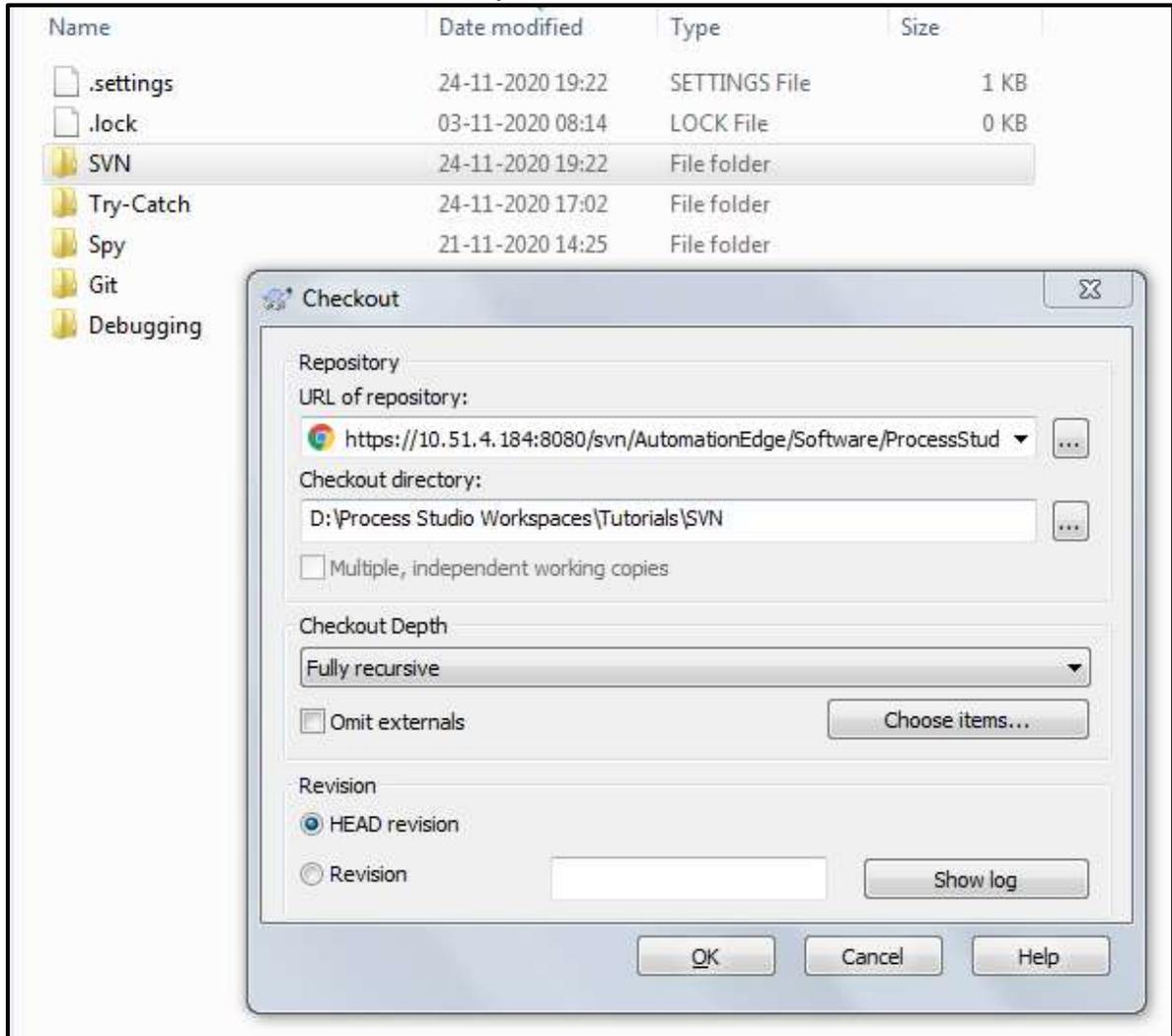
Name	Date modified	Type	Size
.settings	24-11-2020 19:22	SETTINGS File	1 KB
.lock	03-11-2020 08:14	LOCK File	0 KB
SVN	24-11-2020 19:22	File folder	
Try-Catch	24-11-2020 17:02	File folder	
Spy	21-11-2020 14:25	File folder	
Git	18-11-2020 12:07	File folder	
Debugging	18-11-2020 11:41	File folder	

6. It is assumed that you have installed SVN Client such as TortoiseSVN.

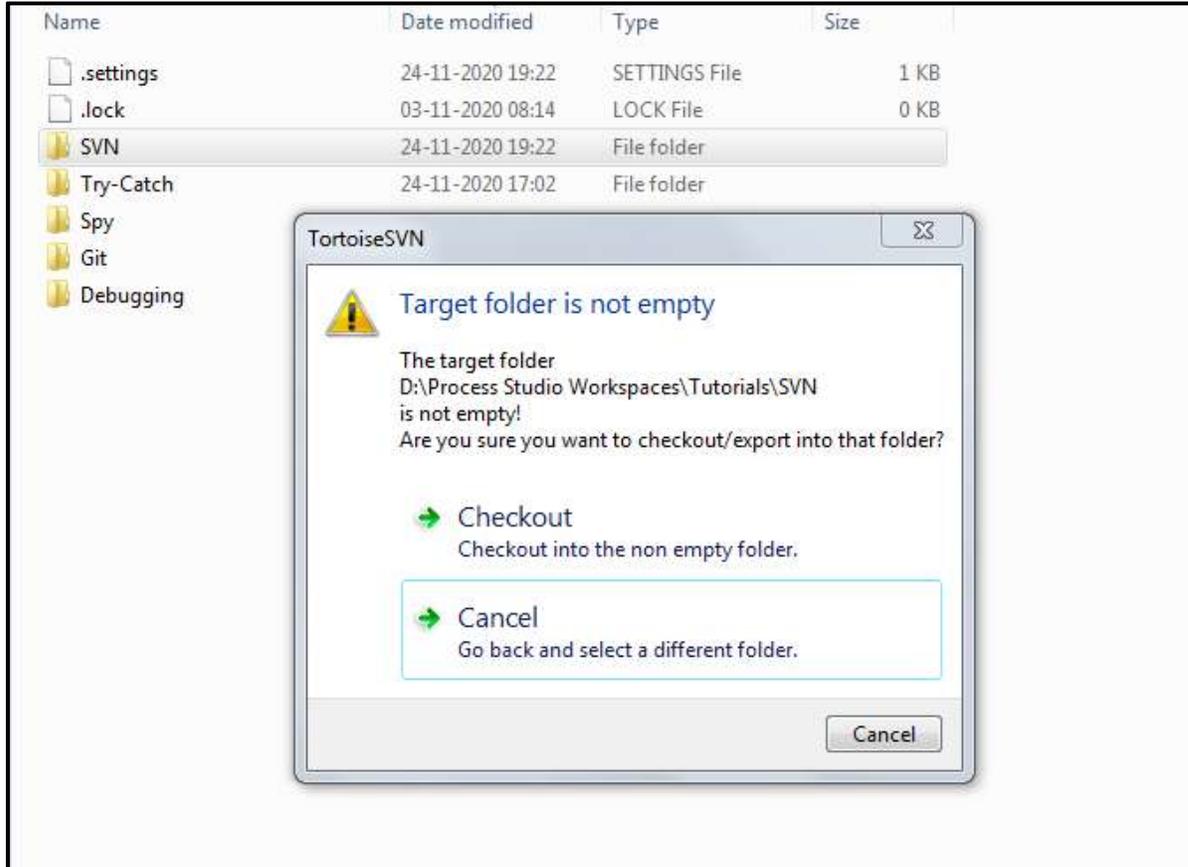
7. Right click on the Project folder and select SVN Checkout...



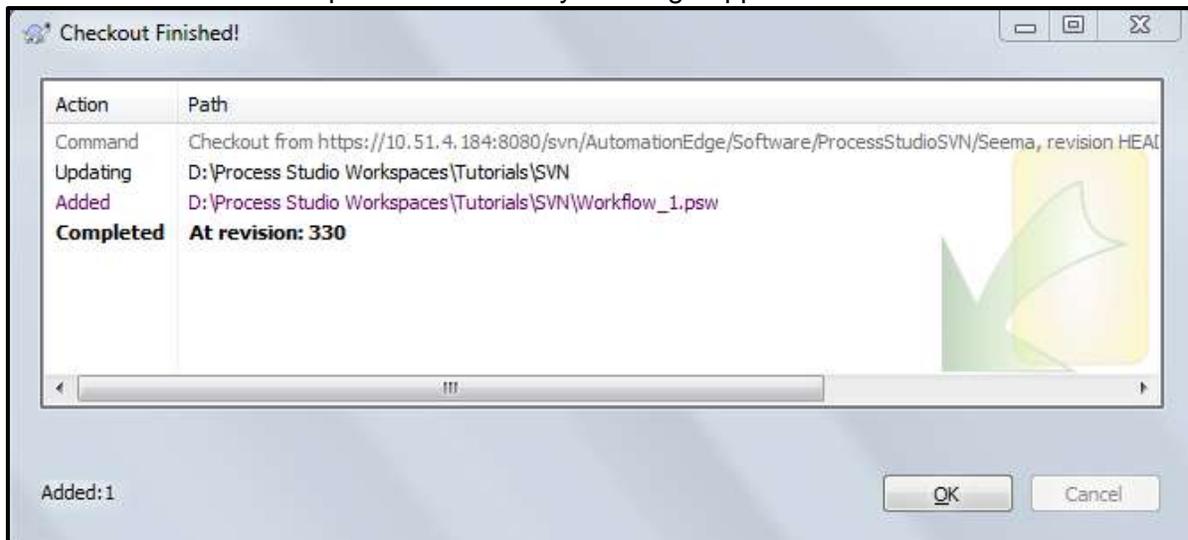
8. In the Checkout window provide configuration details as below. Provide URL of SVN Repository a checkout directory which can be same as Project directory or a subdirectory in the Project directory. If you create a subdirectory it appears as a folder in the Project.
9. Fully The default value for Checkout Depth is Fully recursive for a full checkout including all folders and sub-folders. (The other options are Immediate children, including folders, Only file children, Only this item)
10. Head revision radio button is chosen by default to check out the latest versions of all the files.



11. Since our Project directory is the same as SVN checkout directory there is already a .project file in this location. You get a warning message that target folder is not empty. Click Checkout.



12. Checkout command completed successfully message appears.



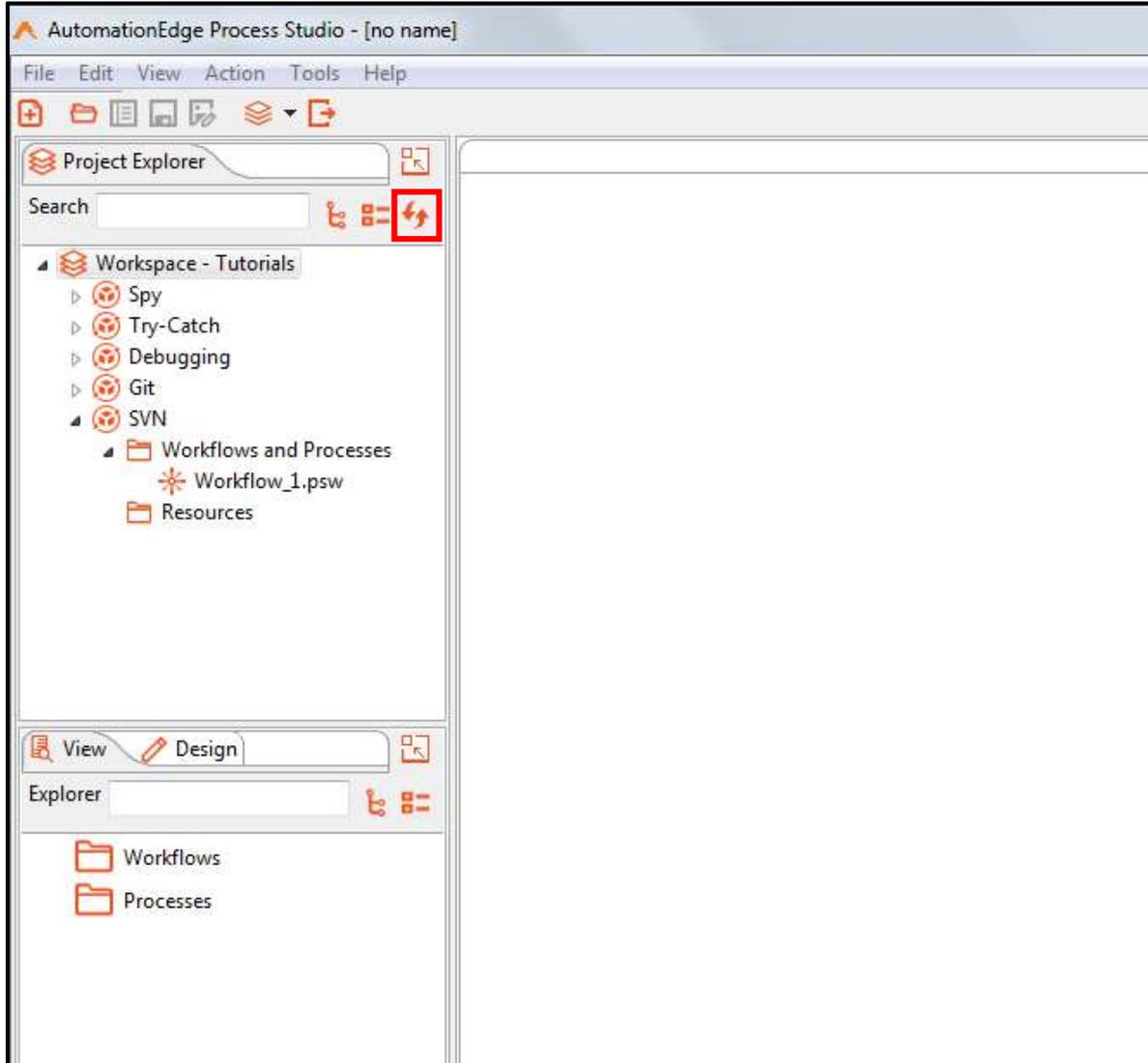
13. The checked out directory is seen with a green tick as seen below.

Name	Date modified	Type	Size
 .settings	24-11-2020 19:22	SETTINGS File	1 KB
 .lock	03-11-2020 08:14	LOCK File	0 KB
 SVN	24-11-2020 19:30	File folder	
 Try-Catch	24-11-2020 17:02	File folder	
 Spy	21-11-2020 14:25	File folder	
 Git	18-11-2020 12:07	File folder	
 Debugging	18-11-2020 11:41	File folder	

14. The remote SVN has a workflow file as seen below which is pulled in the local checkout directory.

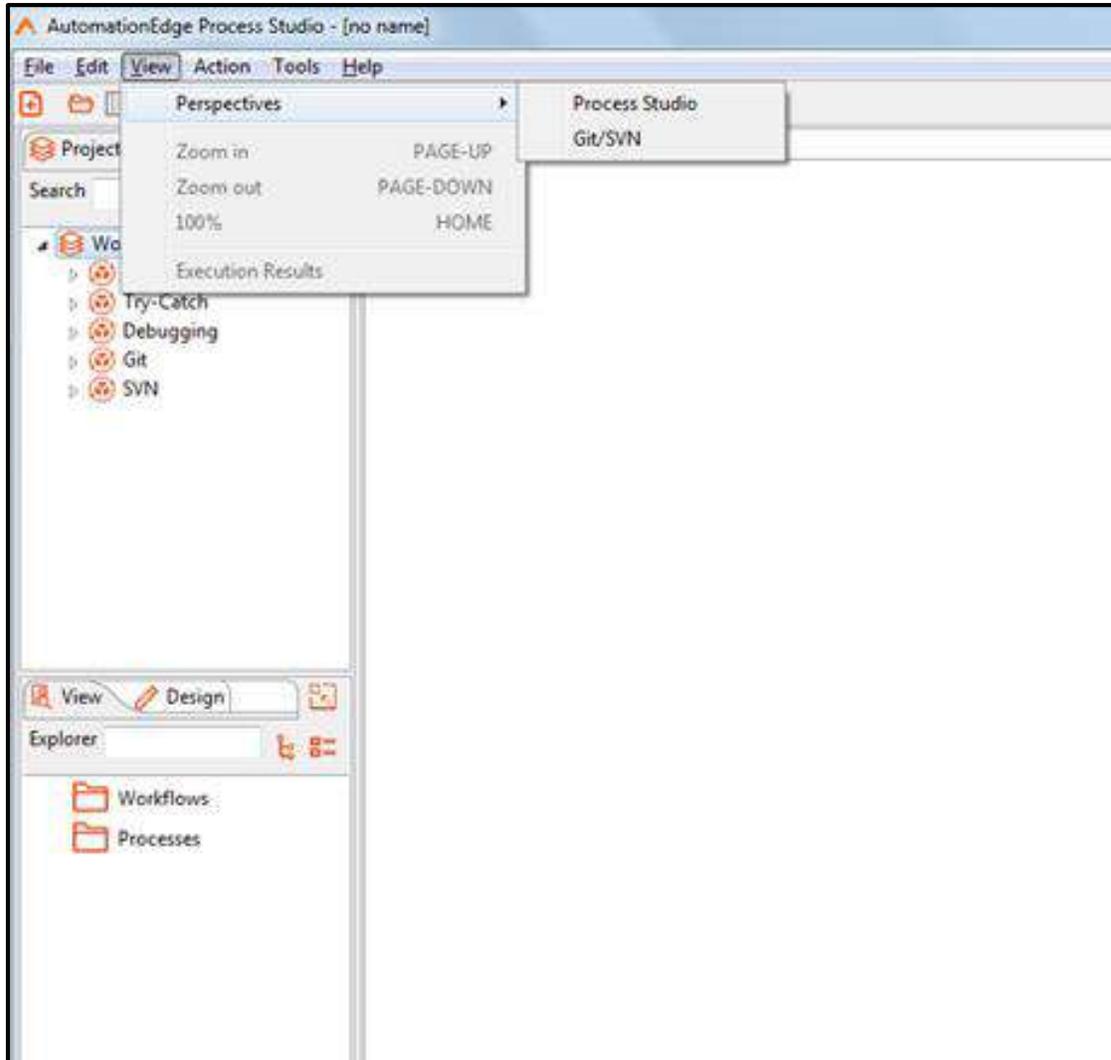
Name	Date modified	Type	Size
 .project	24-11-2020 19:22	PROJECT File	1 KB
 Workflow_1.psw	24-11-2020 19:30	PSW File	15 KB

15. Refresh Project explorer with refresh icon as highlighted in red below. You can also see the workflow in the SVN project now.

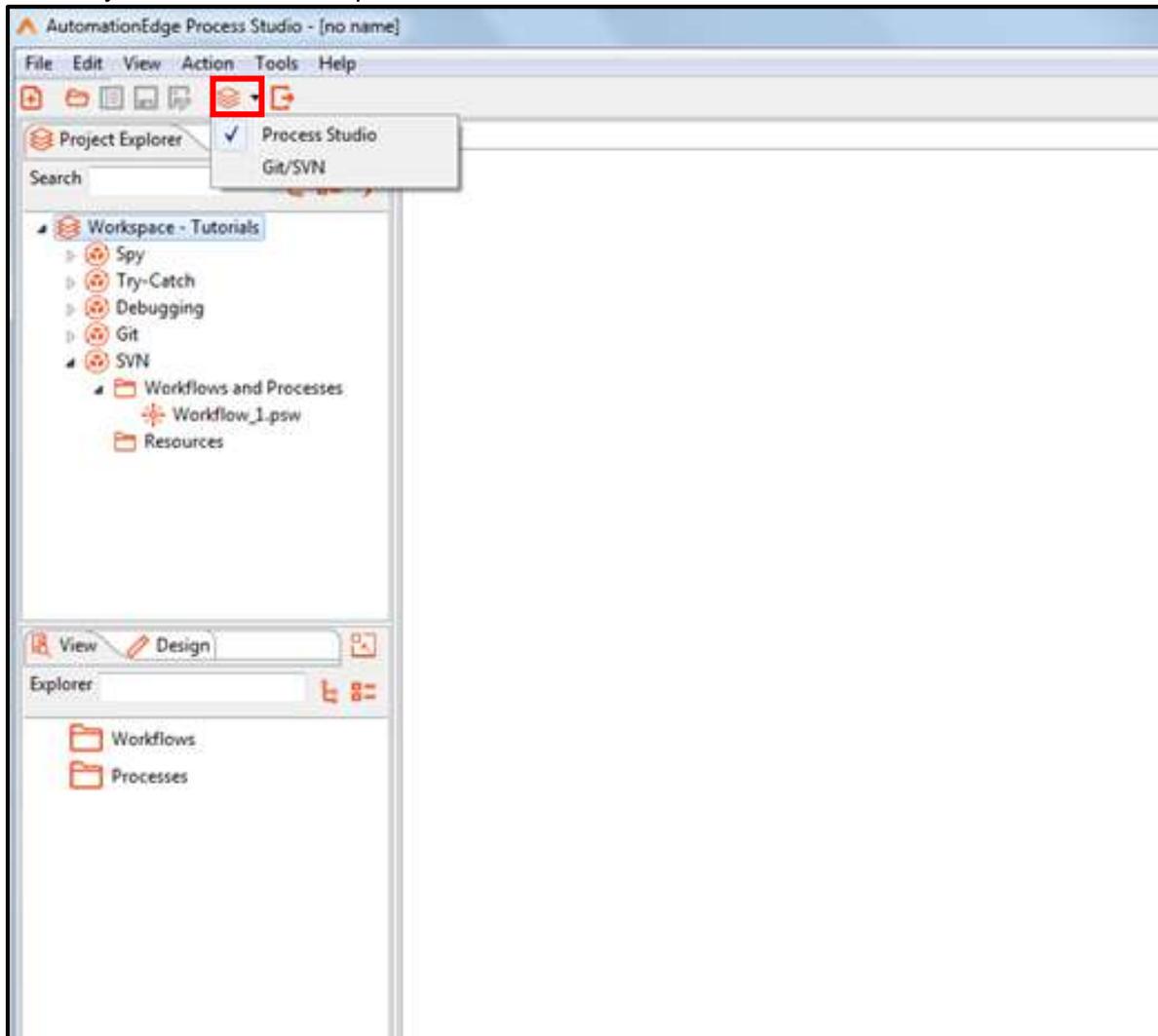


19.3 Open SVN repository in Process Studio

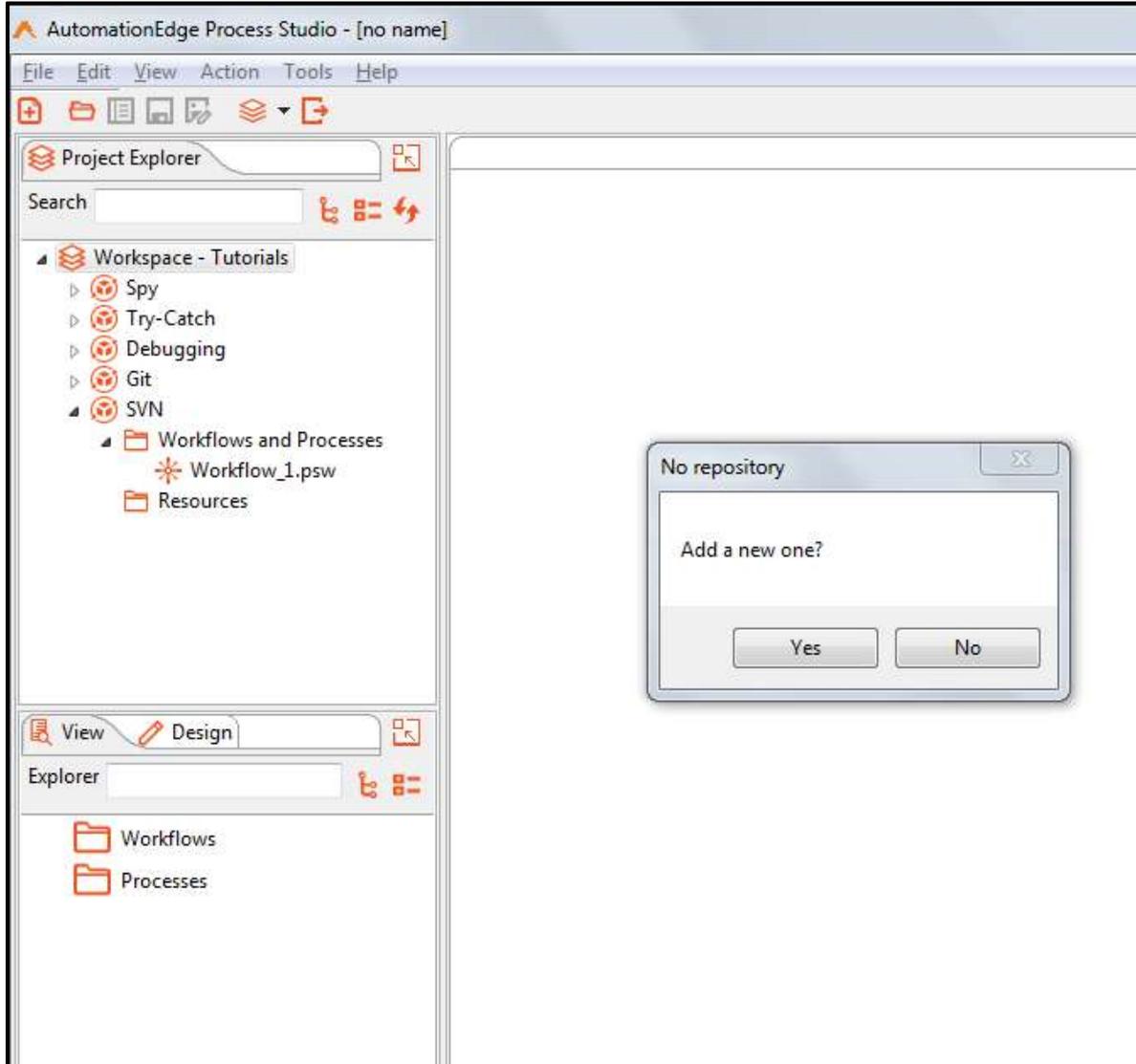
1. Now let us open this SVN repository in Process Studio.
2. Process Studio has two views or Perspectives – Process Studio and Git/SVN as seen below under View menu.



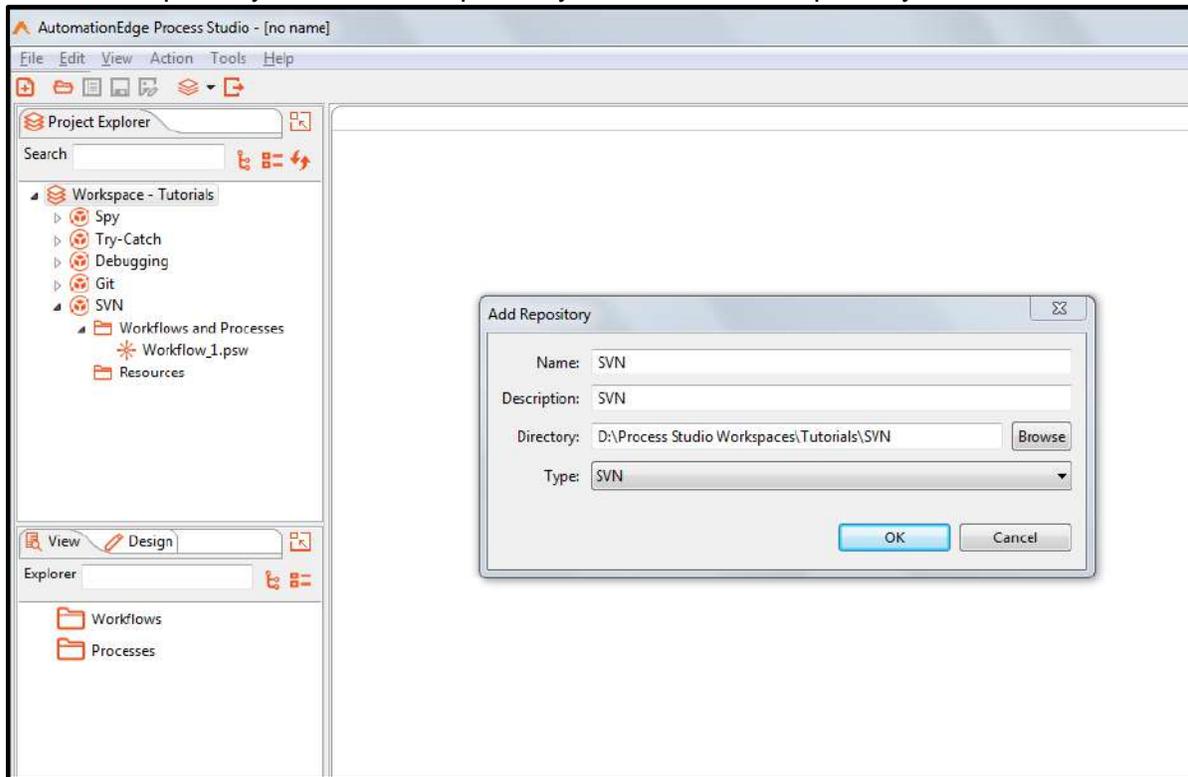
3. You may also click the Perspectives icon as seen below. Select Git/SVN.



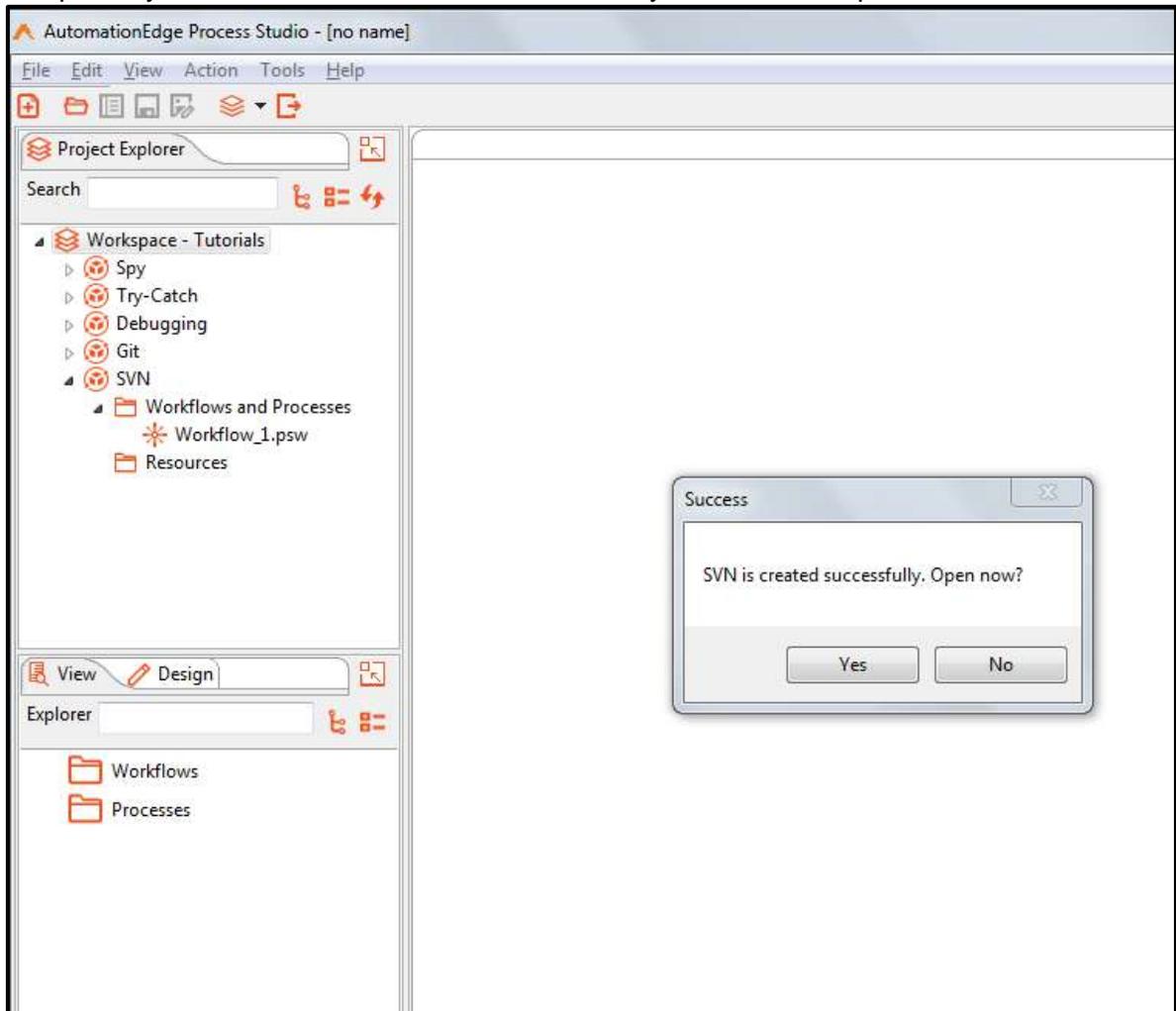
4. Since this is the first time we are switching to Git/SVN perspective it asks to add a new repository.



5. Provide a repository name and the path of your checked out repository.

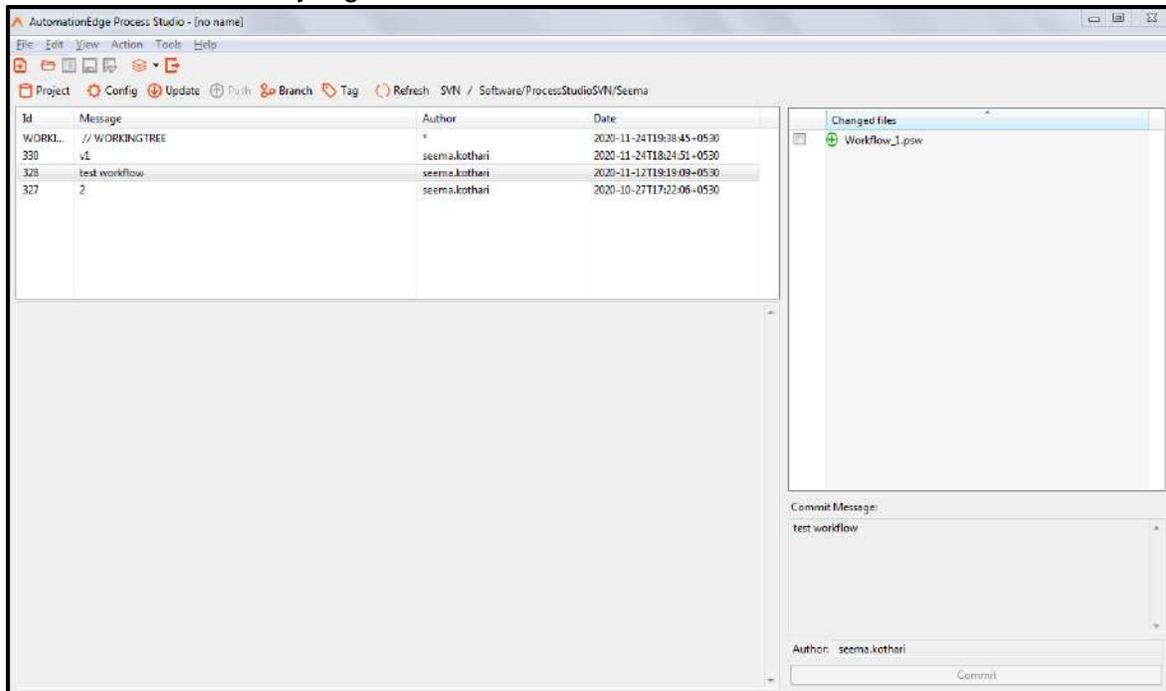


- A repository with name SVN is created successfully. Click Yes to open it.



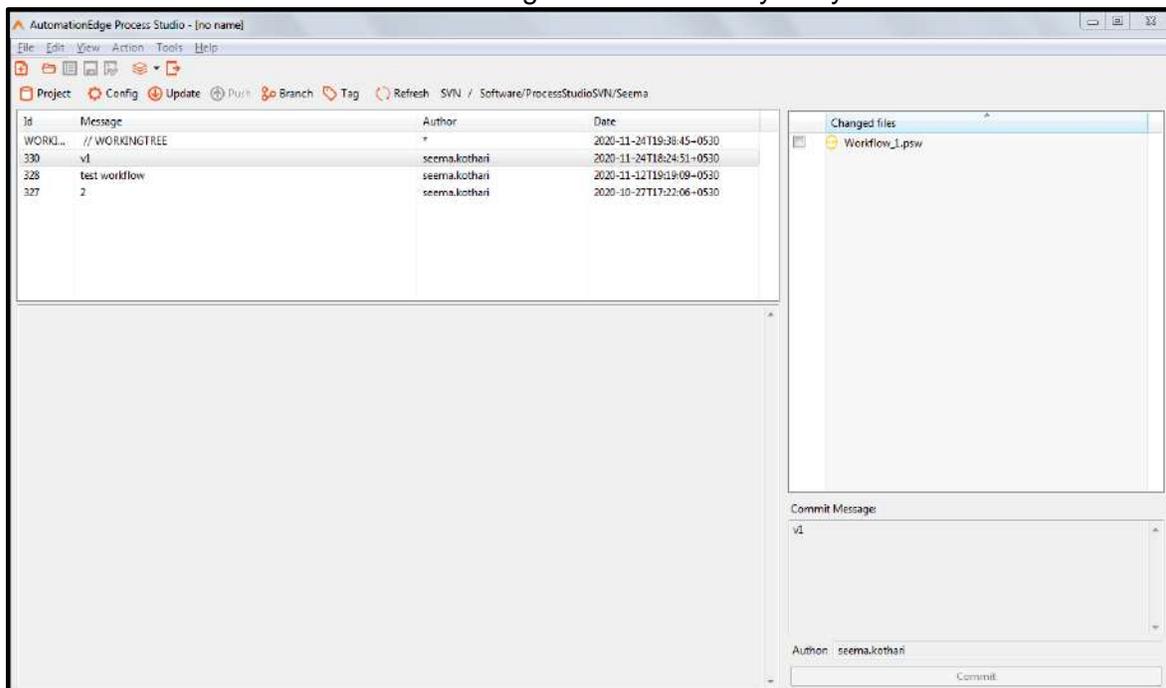
19.3.1 New files with green icon

7. You can now see a Working Tree and some previous commits.
8. In the 'test workflow' commit you can see one changed workflow file on the right. It is a new file addition as indicated by a green icon.



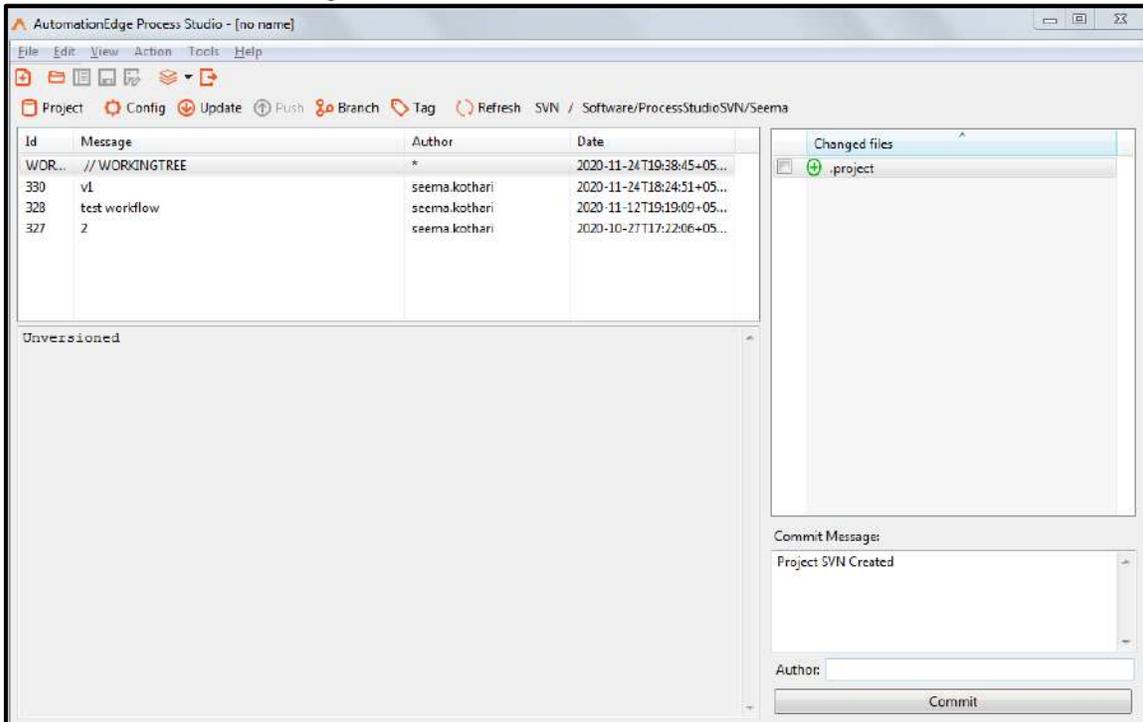
19.3.2 Modified files with yellow icon

9. In the v1 commit the workflow was changed as indicated by the yellow icon.

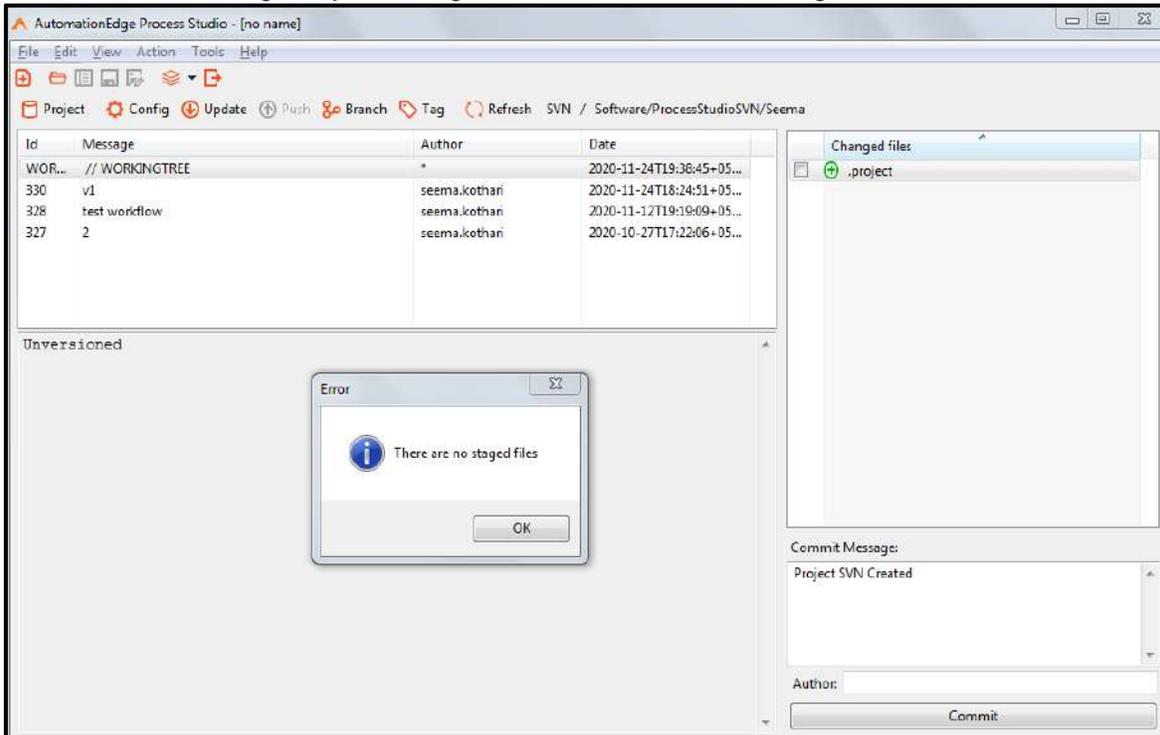


19.3.3 SVN Commit

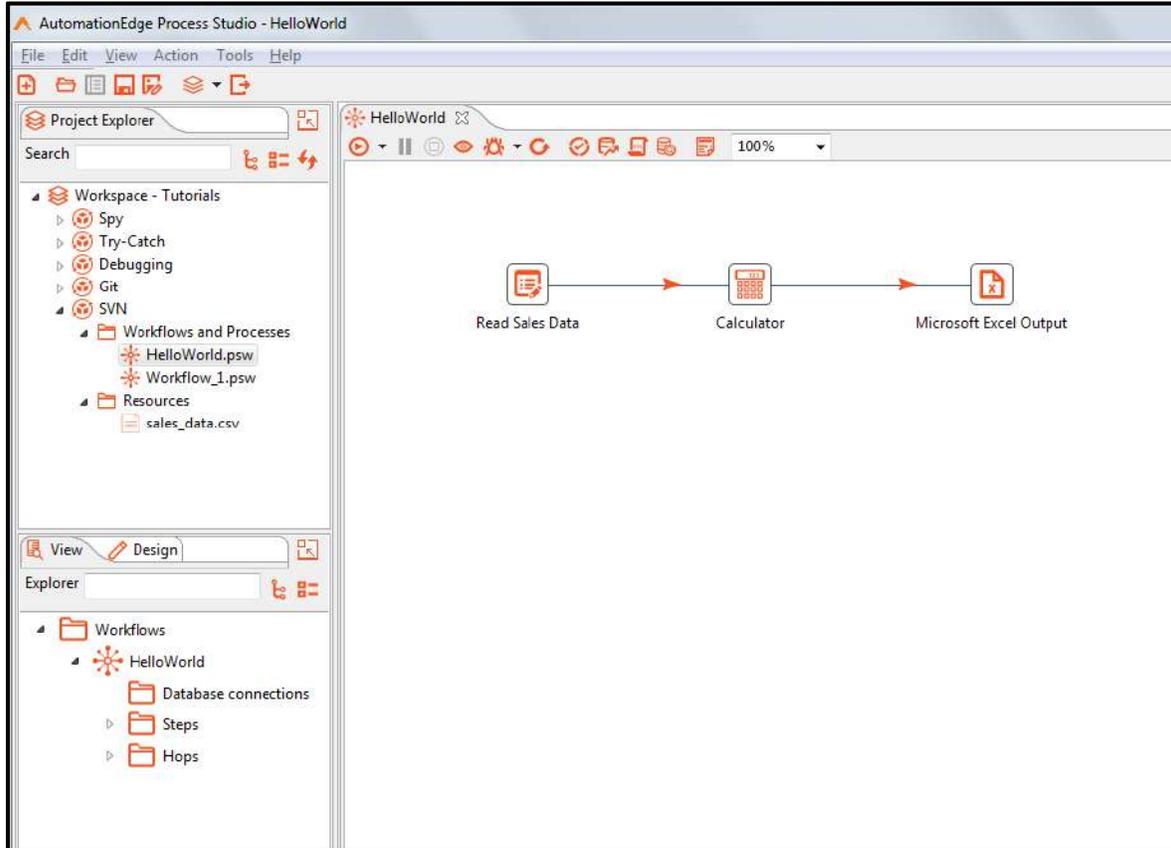
10. In the working tree you can see a new project file as indicated by a green icon.
11. Provide a commit message and click Commit.



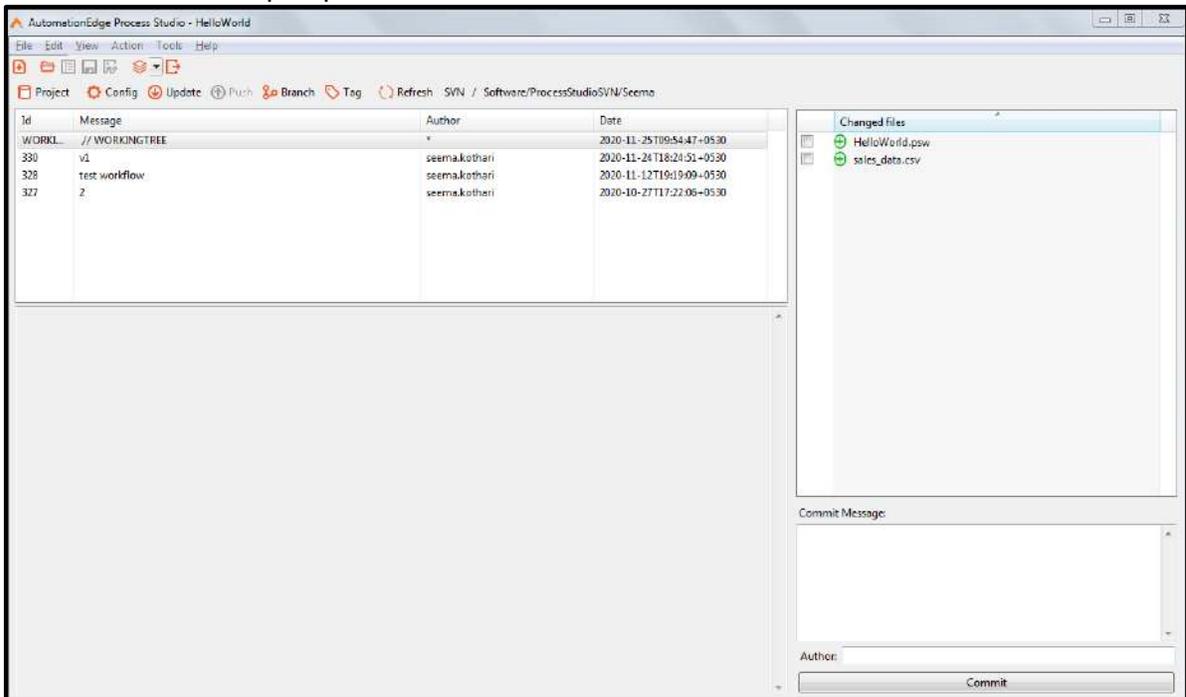
12. Since we have not checked the changed files we get a message there are no staged files.
13. Files need to be staged by enabling checkbox next to the changed files for commit.



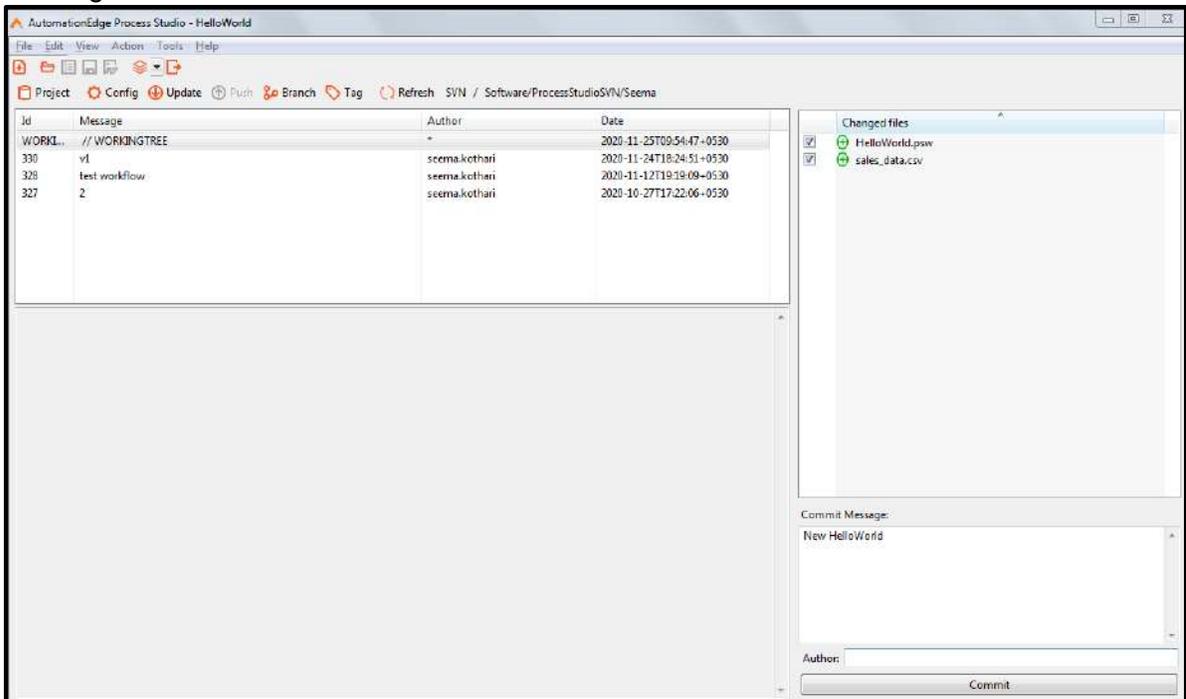
14. Now go back to Process Studio Perspective. Add a workflow and resource file as seen below.



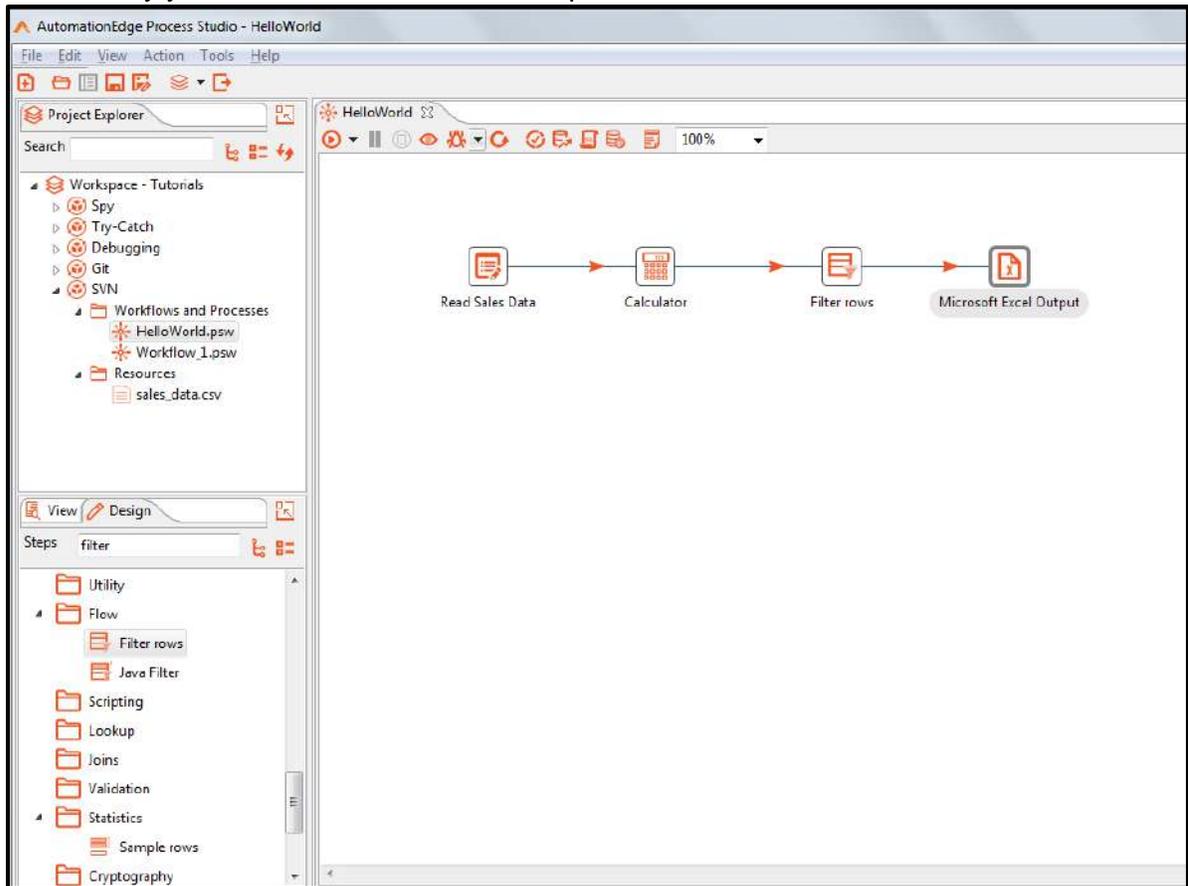
15. Go back to Git/SVN perspective.



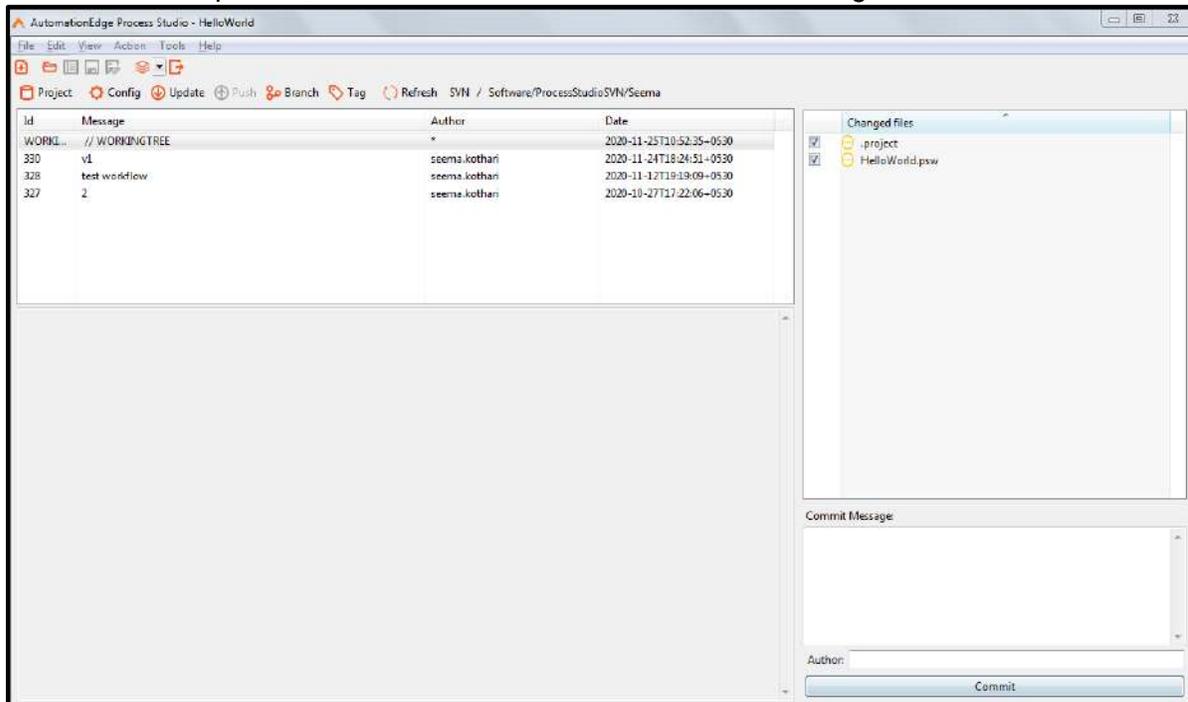
16. Now enable checkbox next to the new files, provide a commit message and press commit. The working tree is now committed.



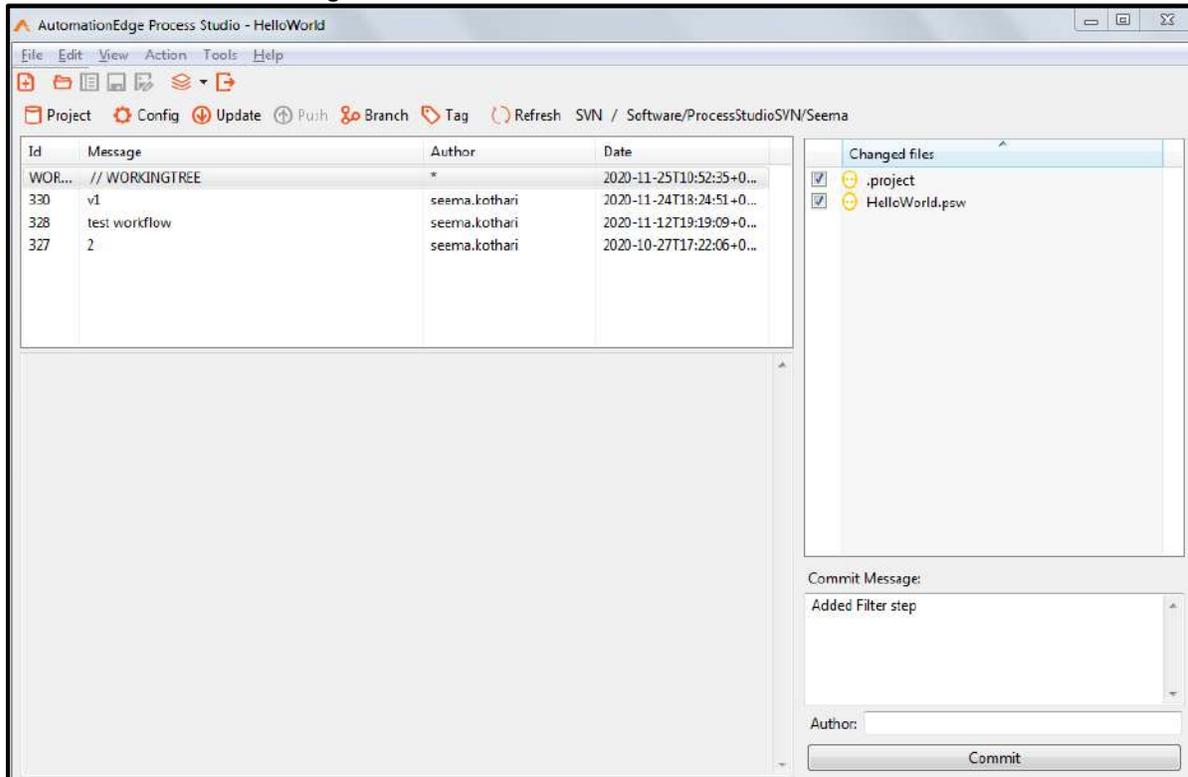
17. Now modify your workflow to add a filter step.



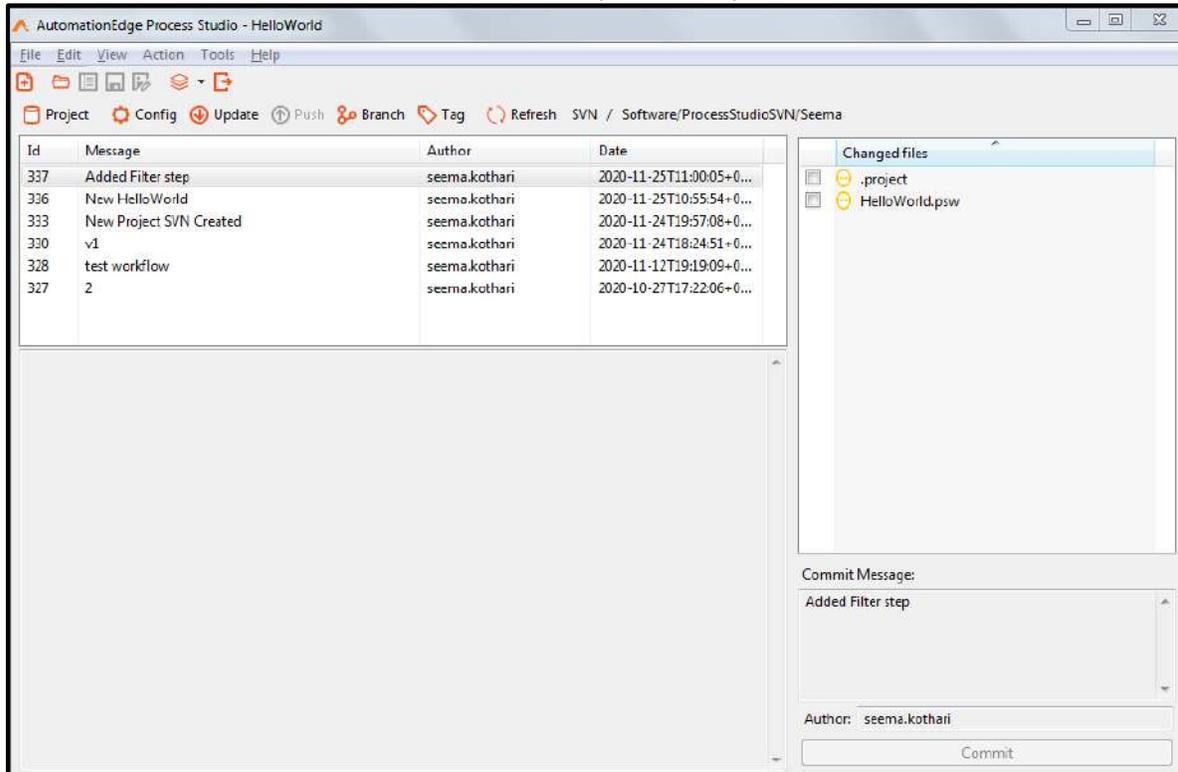
18. In the SVN Perspective, click Refresh. You can now see the changes HelloWorld workflow file.



19. Provide a commit message and click commit.



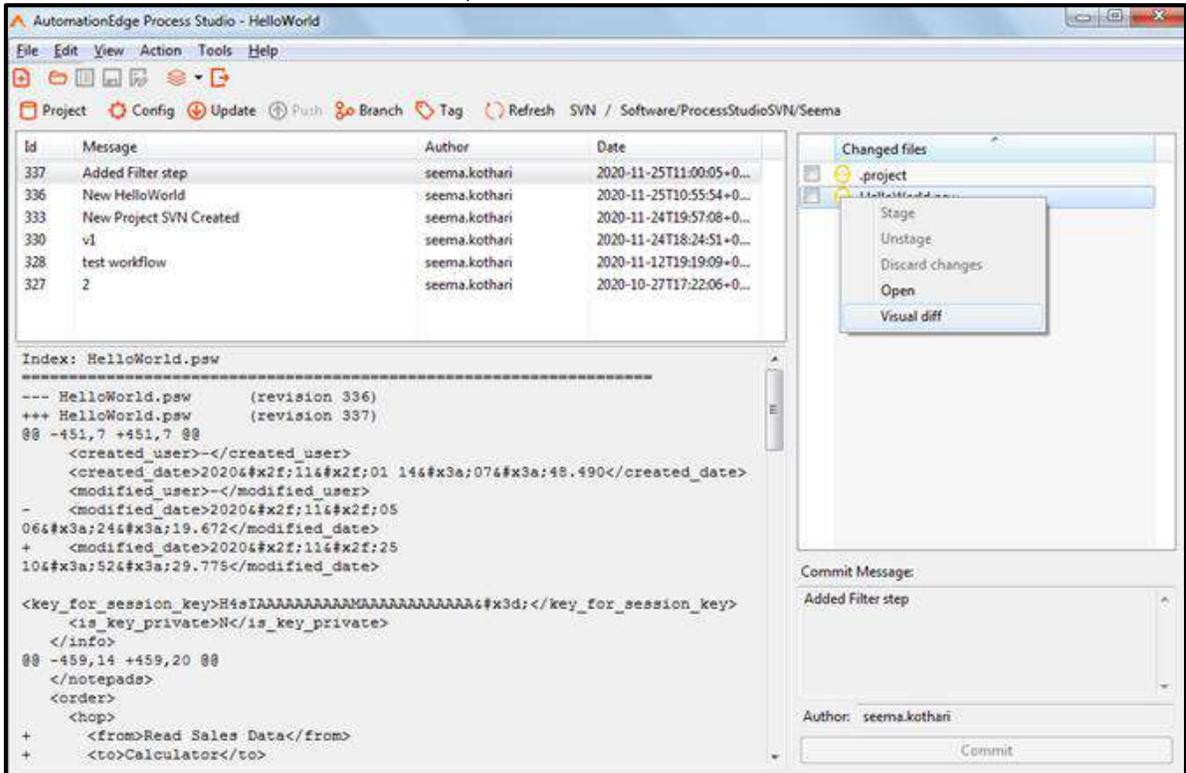
20. There is no explicit Push in for SVN. Commit automatically pushes changes to SVN repository.
21. Click Update to pull the changes from the repository and see the commit points.
22. Click Refresh. You can see the 'Add Filter Step' commit point.



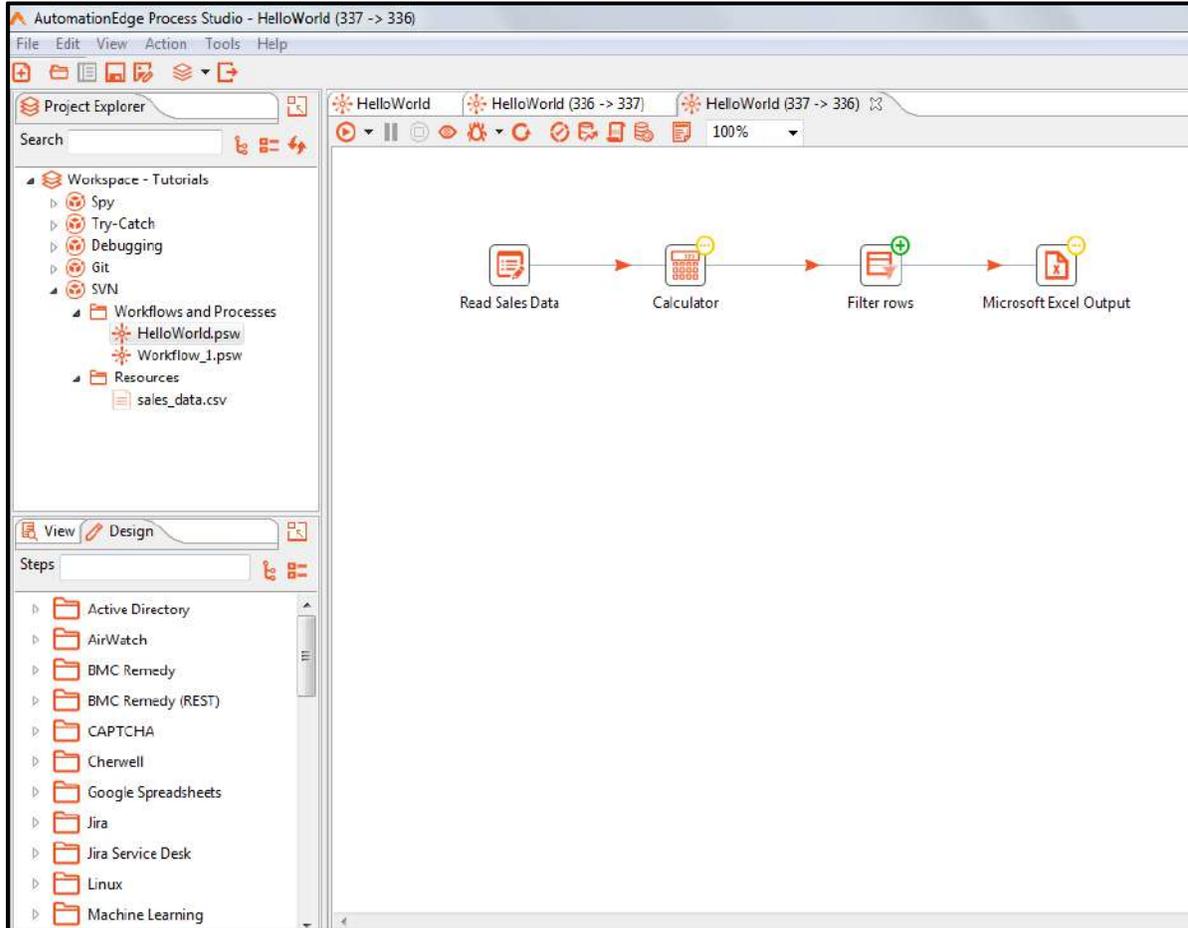
19.3.4 Visual Difference

23. For on any of the messages for the commits. Right click on the corresponding changed files and click Visual diff.

- When any one commit is selected, the diff is between the selected commit and its first parent commit.
- When multiple commits are selected, the diff is between the newest commit and the oldest commit.
- When no commit is selected, it is assumed that WORKINGTREE is selected.

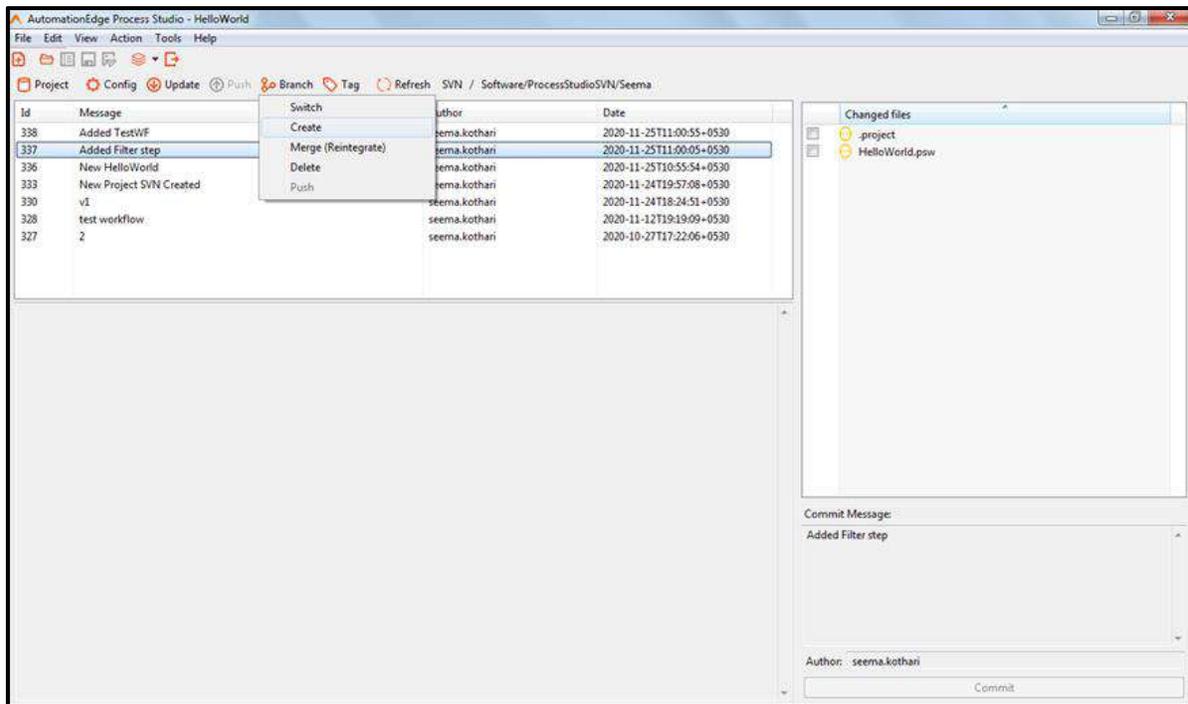


24. The new additions are in green, the changes are in yellow. If there are any deletions, they would be seen in red.

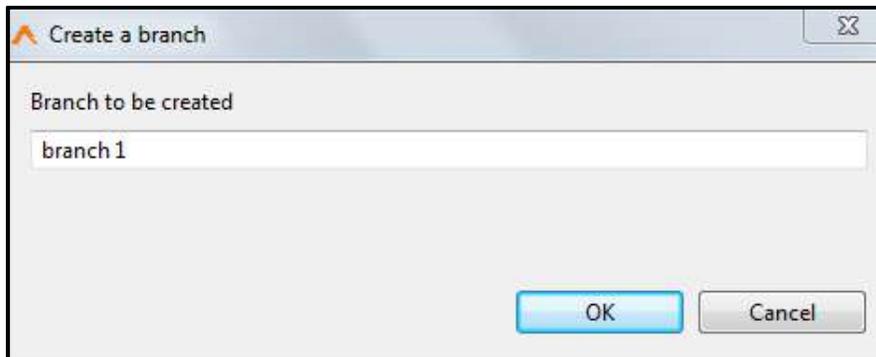


19.3.5 Branch

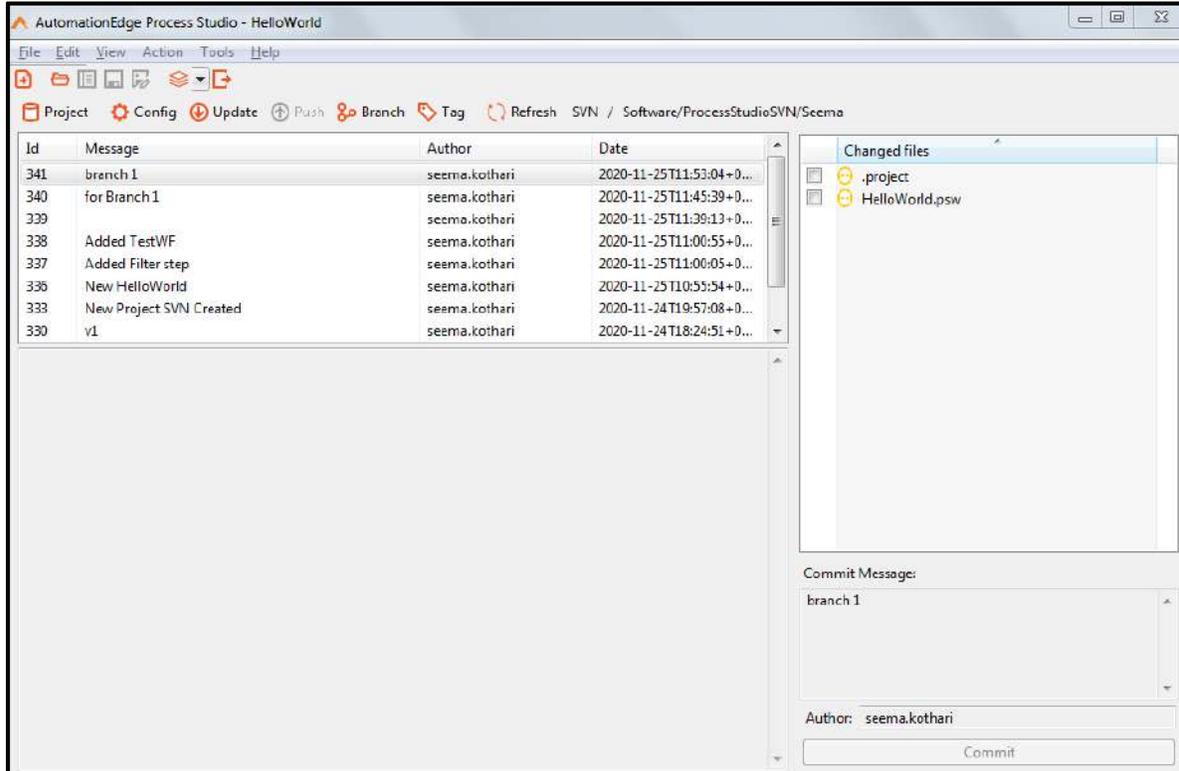
25. Click on Branch and select Create.



26. Create a Branch - branch 1.



27. Click Update to pull and view changes in the SVN repository. We can see that branch1 is created.



28. Tag can also be used similar to Branch but without merge.

Thus we have seen how to integrate SVN repository in Process Studio. We have also discussed Commit to repository, Update from repository, Visual difference and Branch creation. Further Branch options such as merge and conflict will be available in upcoming releases.



Contact

AutomationEdge Technologies Inc



General: info@automationedge.com

Support: support@automationedge.com

Trainings: training.support@automationedge.com



Website: <https://automationedge.com>



India: +91-20-6632-1000

USA: +1-281-220-1057