Take Home A Robot

Google Calendar Conflict Bot

Bot Step-by-Step Guide

UiPath Forward Americas 2018

UiPath

Getting started

Install the free version of UiPath Studio

Use this link to get started:

https://www.uipath.com/community

Once you have UiPath Studio Community or Trial edition installed, you are ready to go!

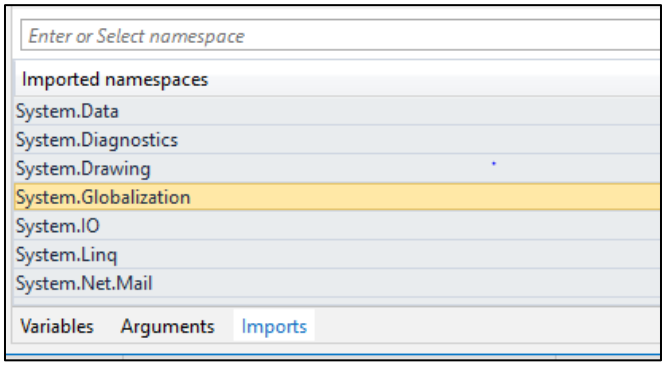
The Google Calendar Conflict Bot

The Google Calendar Conflict Bot takes as input an email address, and a month selection. It navigates to the selected month of the Google calendar, extracting events within the viewport and it then scans for conflicts. If a conflict is detected, the bot alerts the owner of the of the google calendar.

Optionally, the bot can notify all participants involved in the conflicting event. By default, only the owner of the google calendar is notified. Choosing to allow the bot to notify all participants should be done with caution.

This instructional document covers the building of most of the automation. However, the final component of the Google Conflict Calendar is where the bot actually performs the analysis for conflicts. The component has many steps and is beyond the scope of this document. Therefore as a prerequisite, please download the file **Google_Calendar_Conflict_Executor.**

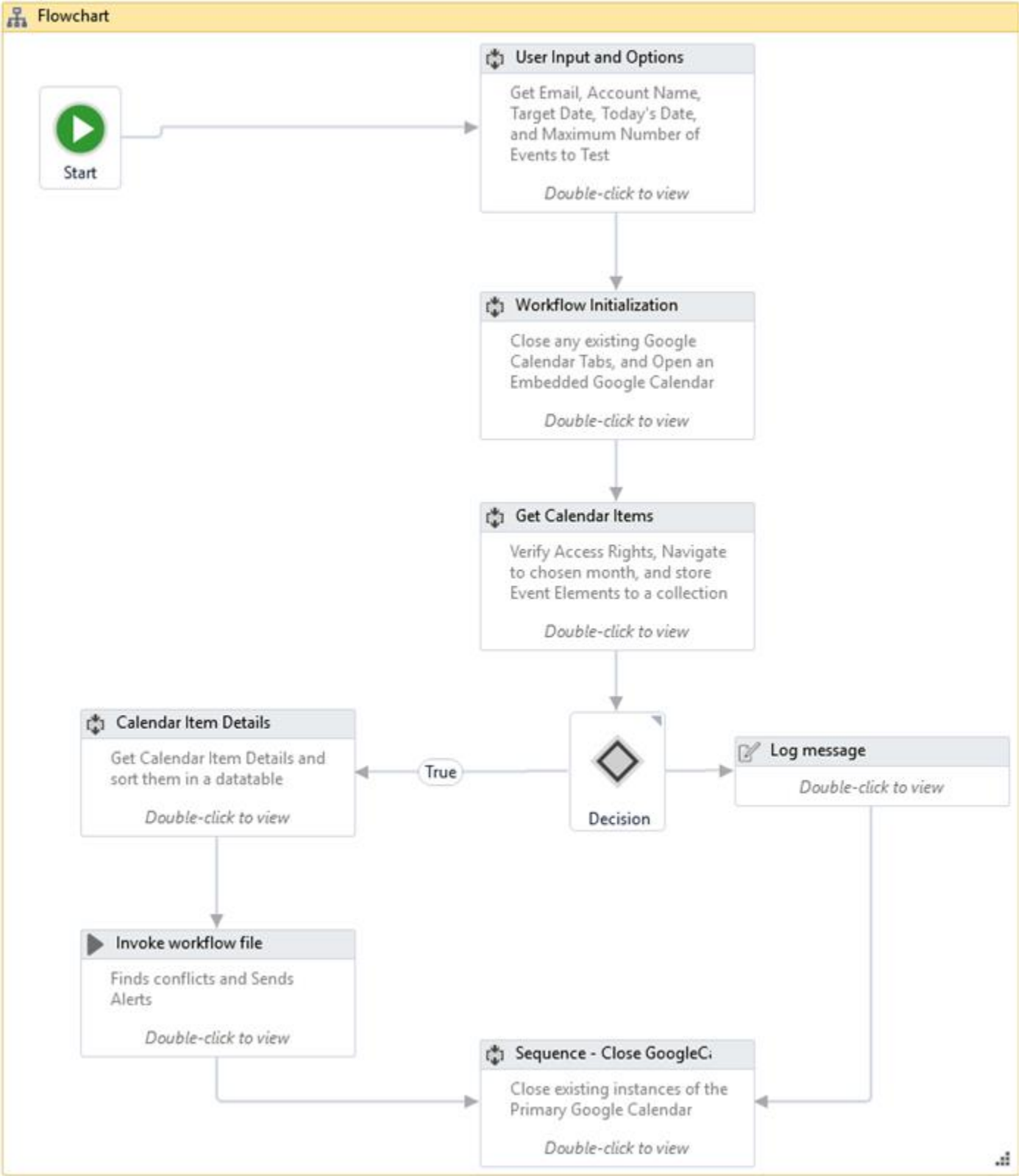
1. Create a New Project (Flowchart)
2. Click on Import Panel below the Workspace, and import `System.Globalization`



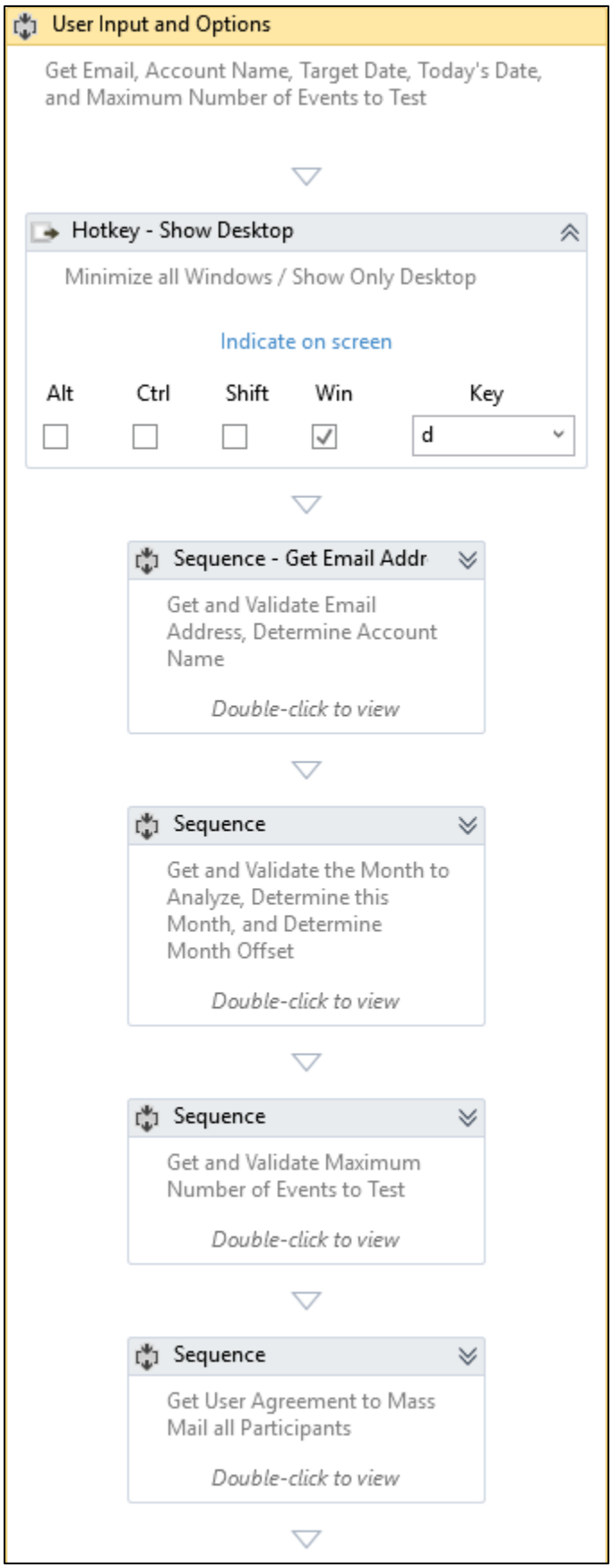
3. Create these Variables within the scope of the Flowchart

Name	Variable type	Scope	Default
bool_Error	Boolean	Flowchart	*Enter a VB expression*
bool_ExistBrowserGoogleCal	Boolean	Flowchart	False
bool_MassEmail	Boolean	Flowchart	False
browser_GoogleCal	Browser	Flowchart	*Enter a VB expression*
datatable_EventInfo	DataTable	Flowchart	*Enter a VB expression*
date_LatestEventEnd	DateTime	Flowchart	DateTime.Parse("December 31, 2000")
date_LatestEventStart	DateTime	Flowchart	DateTime.Parse("December 31, 2000")
date_TargetDate	DateTime	Flowchart	*Enter a VB expression*
date_TodayDate	DateTime	Flowchart	*Enter a VB expression*
elem_eventsElements	IEnumerable<UiElement>	Flowchart	*Enter a VB expression*
int_MaxEvents	Int32	Flowchart	*Enter a VB expression*
int_MonthOffset	Int64	Flowchart	*Enter a VB expression*
int_MyCounter	Int32	Flowchart	*Enter a VB expression*
str_AccountName	String	Flowchart	*Enter a VB expression*
str_email	GenericValue	Flowchart	*Enter a VB expression*
str_LatestEventName	String	Flowchart	*Enter a VB expression*
str_SelectedMonth	String	Flowchart	*Enter a VB expression*
str_Selector	String	Flowchart	*Enter a VB expression*
str_TextInput	String	Flowchart	*Enter a VB expression*
str_TodayYear	String	Flowchart	DateTime.Now.Year.ToString
str_URL_Calendar	GenericValue	Flowchart	*Enter a VB expression*

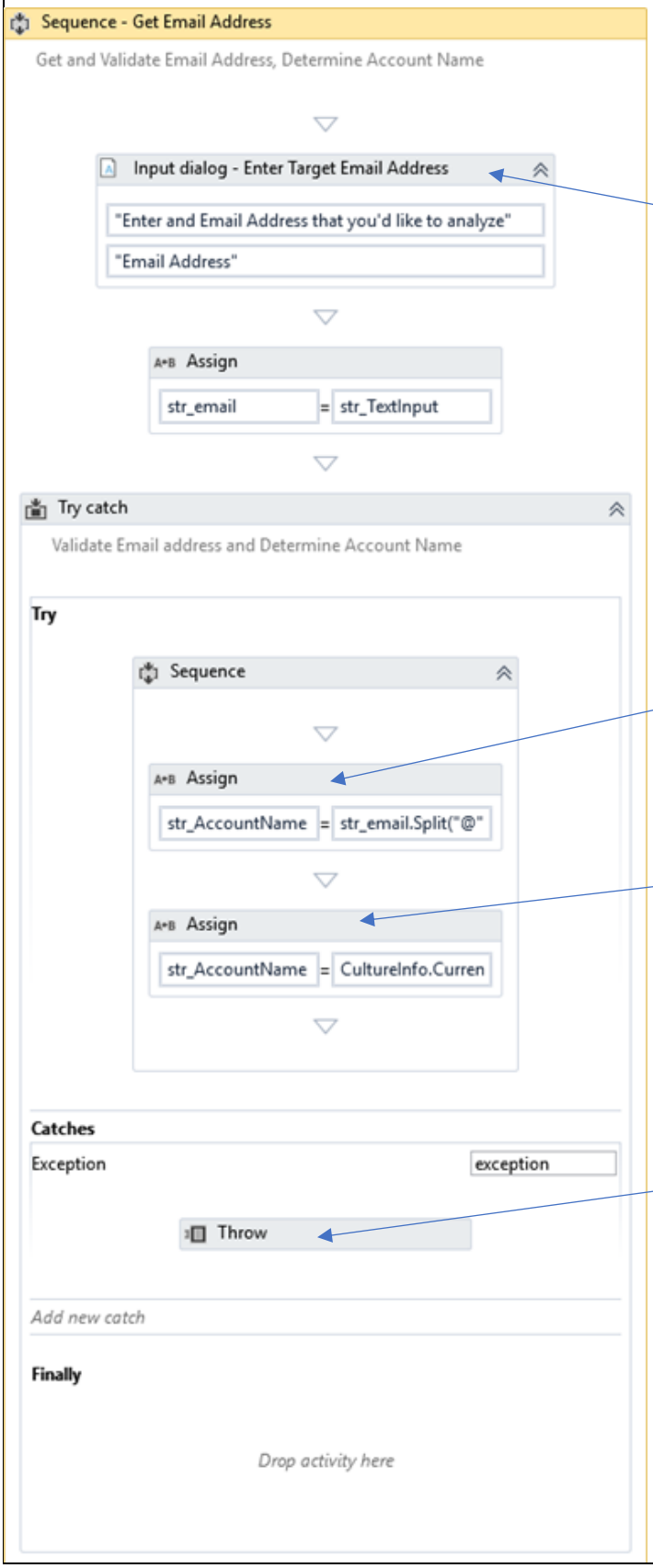
4. Populate the Flowchart with 5 Sequences, 1 Flow Decision, 1 Invoke Workflow Activity, and 1 Log Message as below



5. Enter the 1st Sequence (User Input and Options), and build the following



6. Within the **Get Email Address Sequence** and create the following



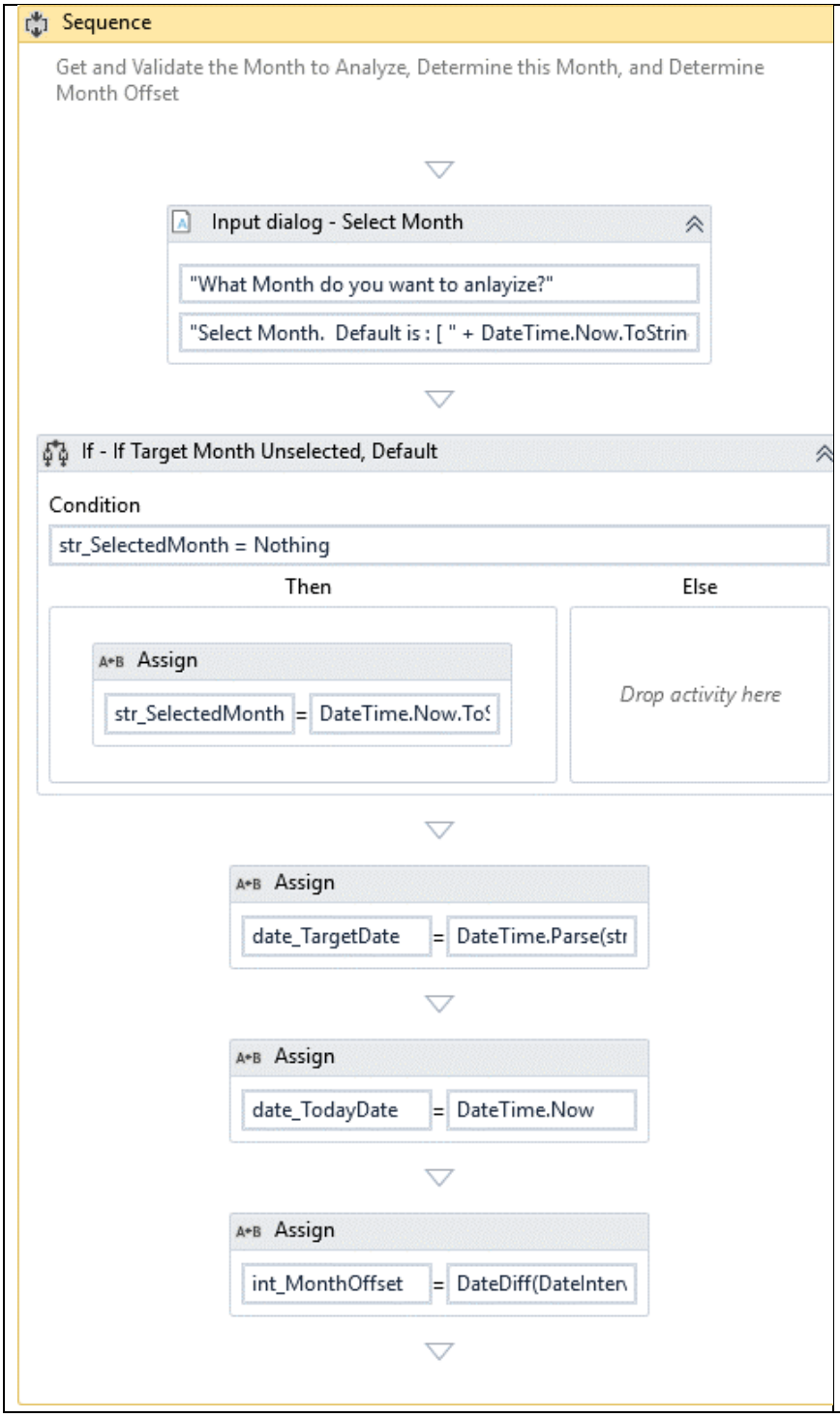
a. Set `str_TextInput` as the OUTPUT for the Input Dialog Box

b. Assign **str_AccountName** to `str_email.Split("@")(0).Replace(".","")`

c. Assign **str_AccountName** to `CultureInfo.CurrentCulture.TextInfo.ToTitleCase(str_AccountName)`

d. In the **Throw Activity**, set the *exception property* to `new businessruleexception("Invalid Account Name")`

7. Within the Date and Time Sequence, create the following



a. Set `str_TextInput` as the OUTPUT for the Input Dialog Activity

b. Set the Label of the Input Dialog Activity as "Select Month. Default is : [" + DateTime.Now.ToString ("MMMM") + "]"

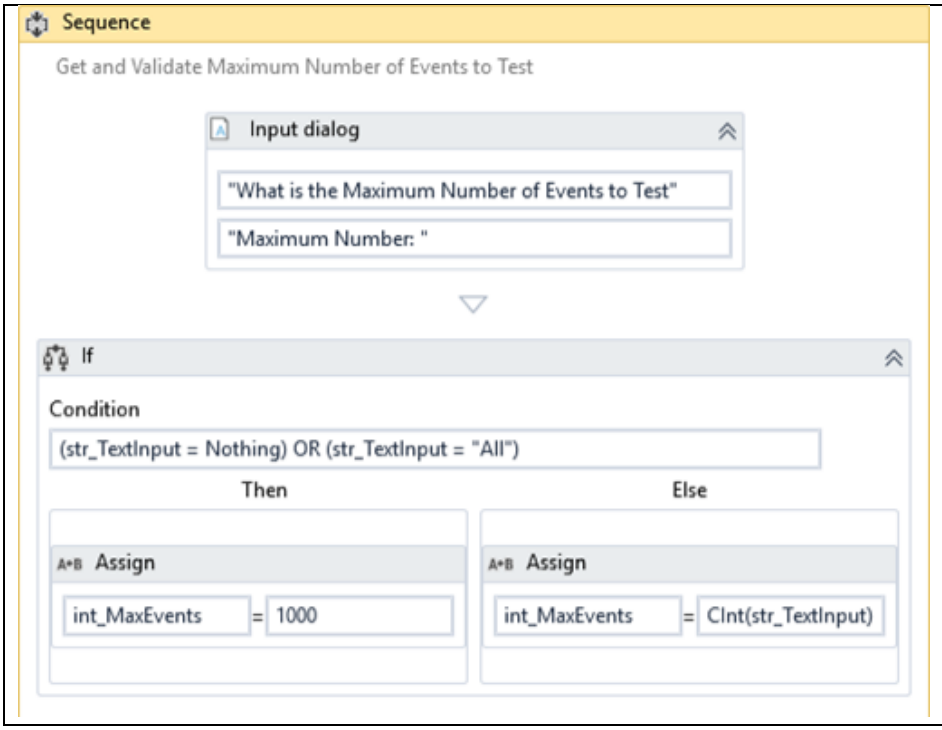
c. Set Assign Activity str_SelectedMonth to DateTime.Now.ToString ("MMMM")

d. Set Assign Activity date_TargetDate to DateTime.Parse (str_SelectedMonth + "1, "+ str_TodayYear)

e. Set Assign Activity date_TodayDate to DateTime.Now

f. Set Assign Activity Int_MonthOffset to DateDiff(DateInterval.Month, date_TodayDate, date_TargetDate)

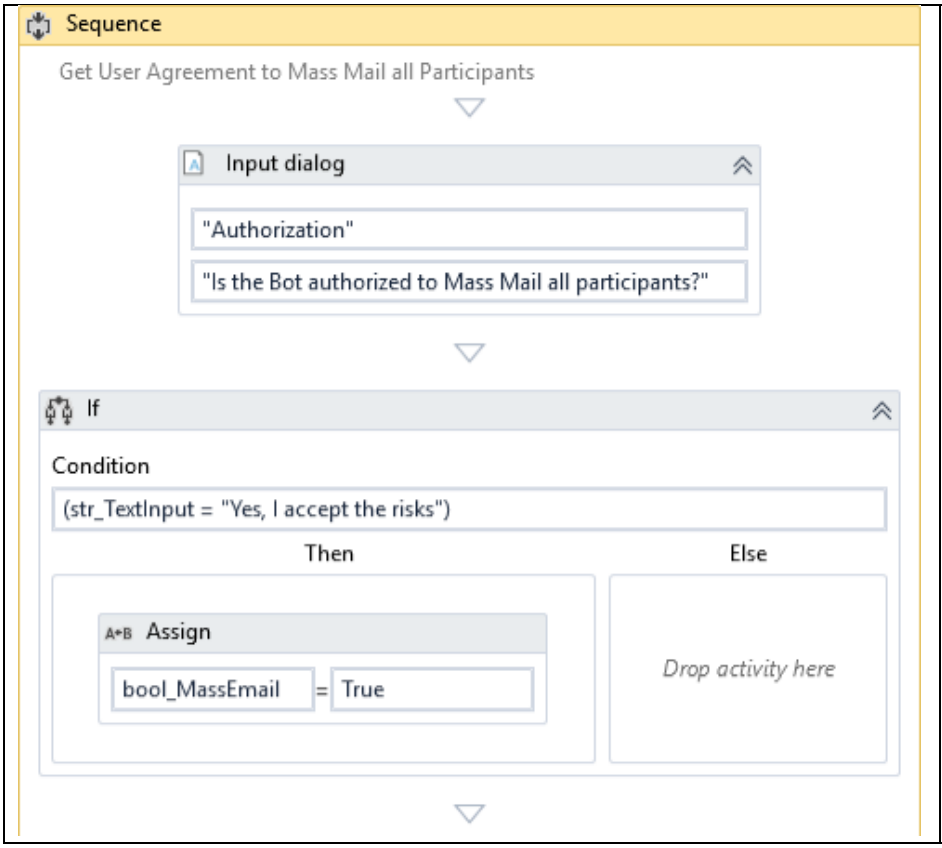
8. Within the Maximum Number of Events Sequence, create the following



Set str_TextInput as the OUTPUT for the Input Dialog Activity

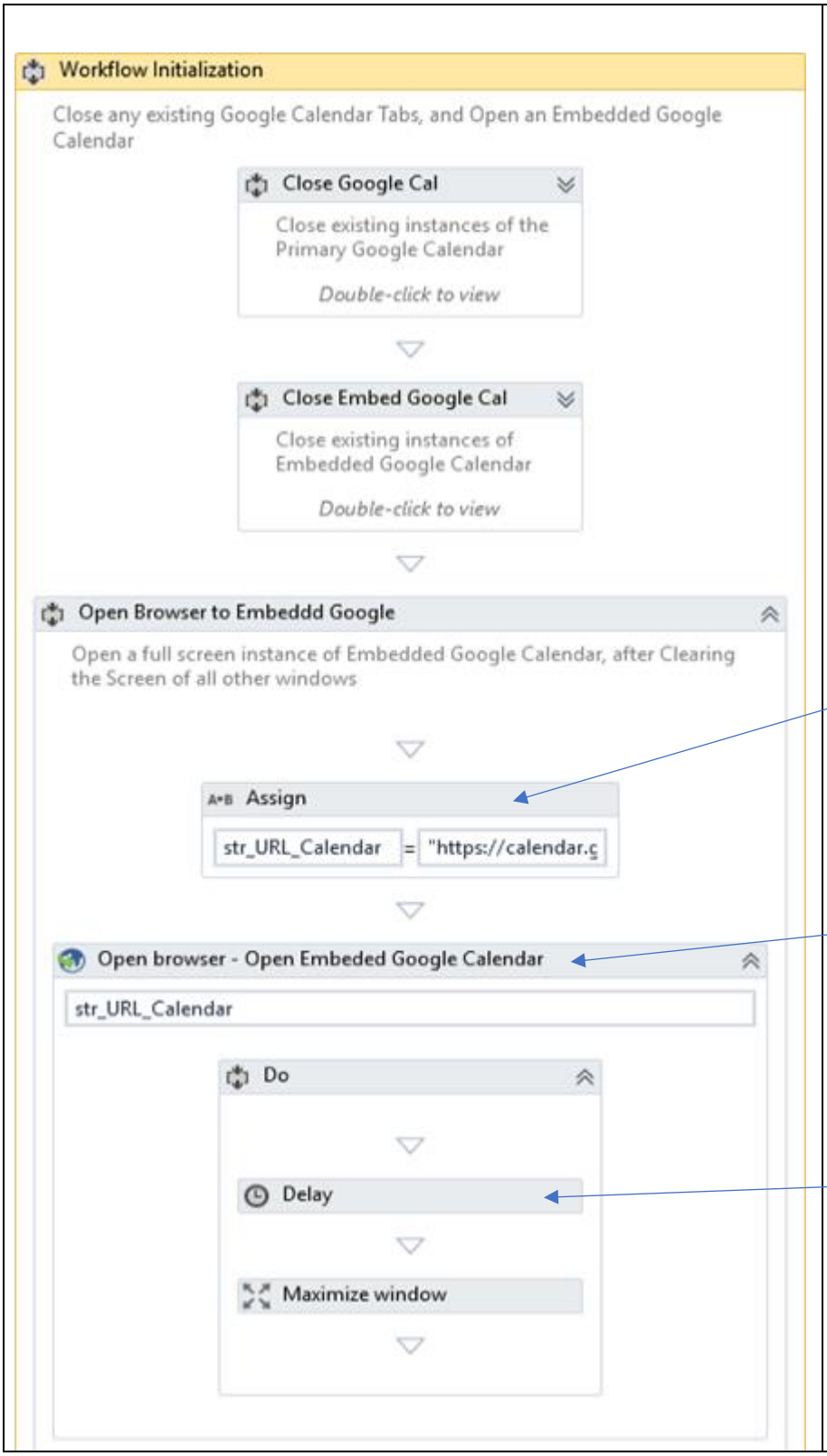
In Else Field, set Assign Activity int_MaxEvents to CInt(str_TextInput)

9. Within the Get User Agreement for Mass Mail Sequence, create the following



Set `str_TextInput` as the OUTPUT for the Input Dialog Activity

10. Go to Flow Chart, and double click into the WorkFlow Initialization and create this.

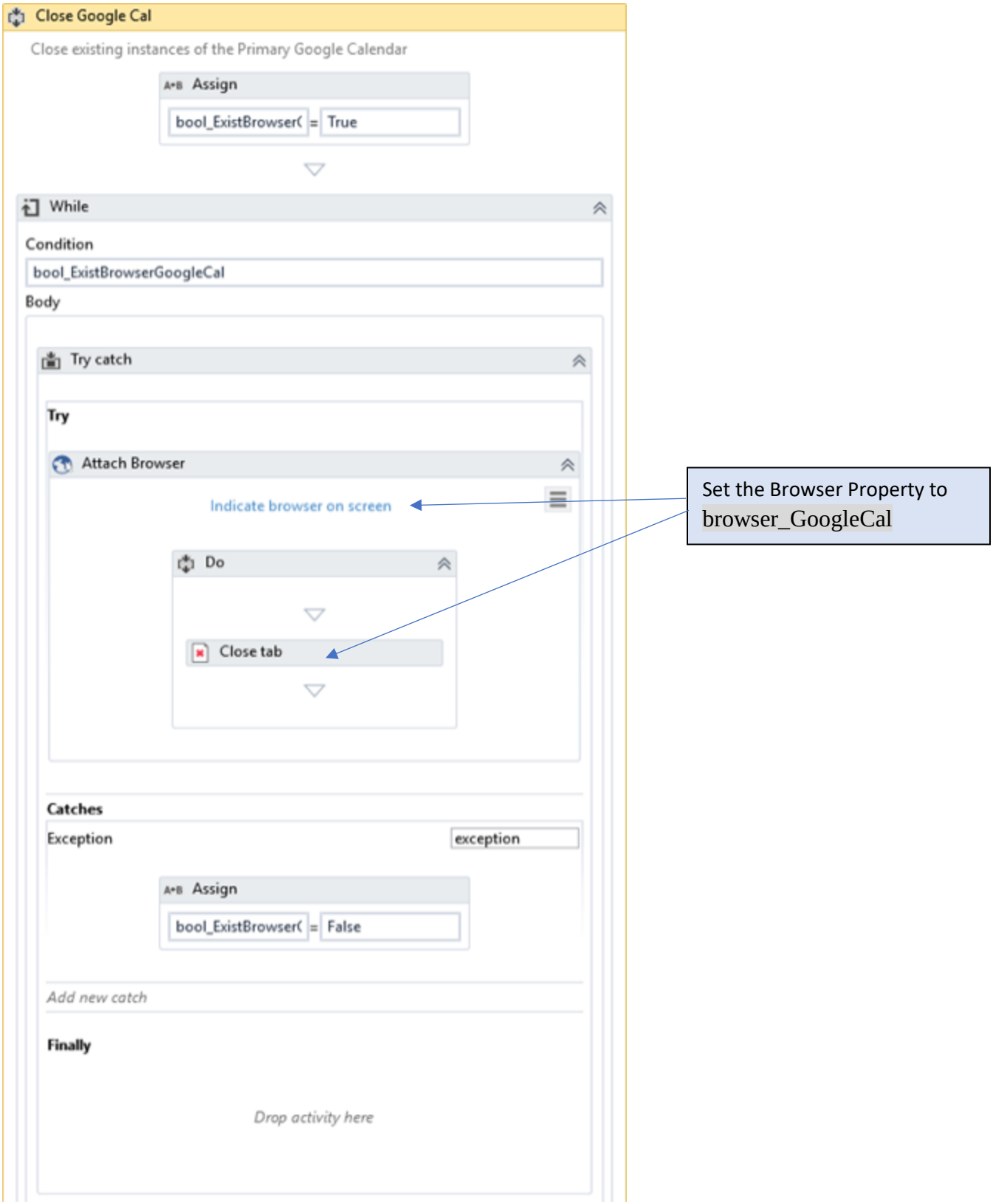


Set the Assign Activity str_URL_Calendar to "https://calendar.google.com /calendar/embed?src="+ str_email.Replace("@","%40")+ "&ctz=America%2FNew_York"

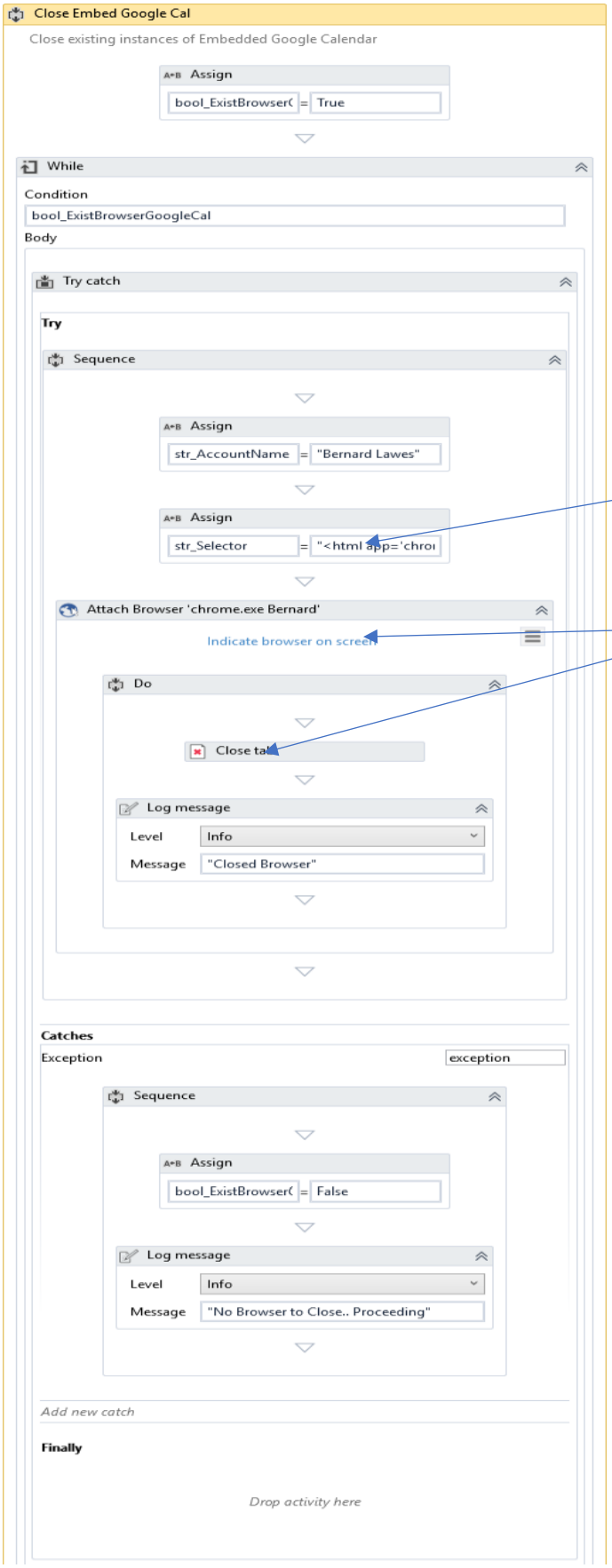
Set the Output Ui Browser Property to browser_GoogleCal

Set Delay to 3 seconds

11. Double click into **Close Google Cal** and create the following



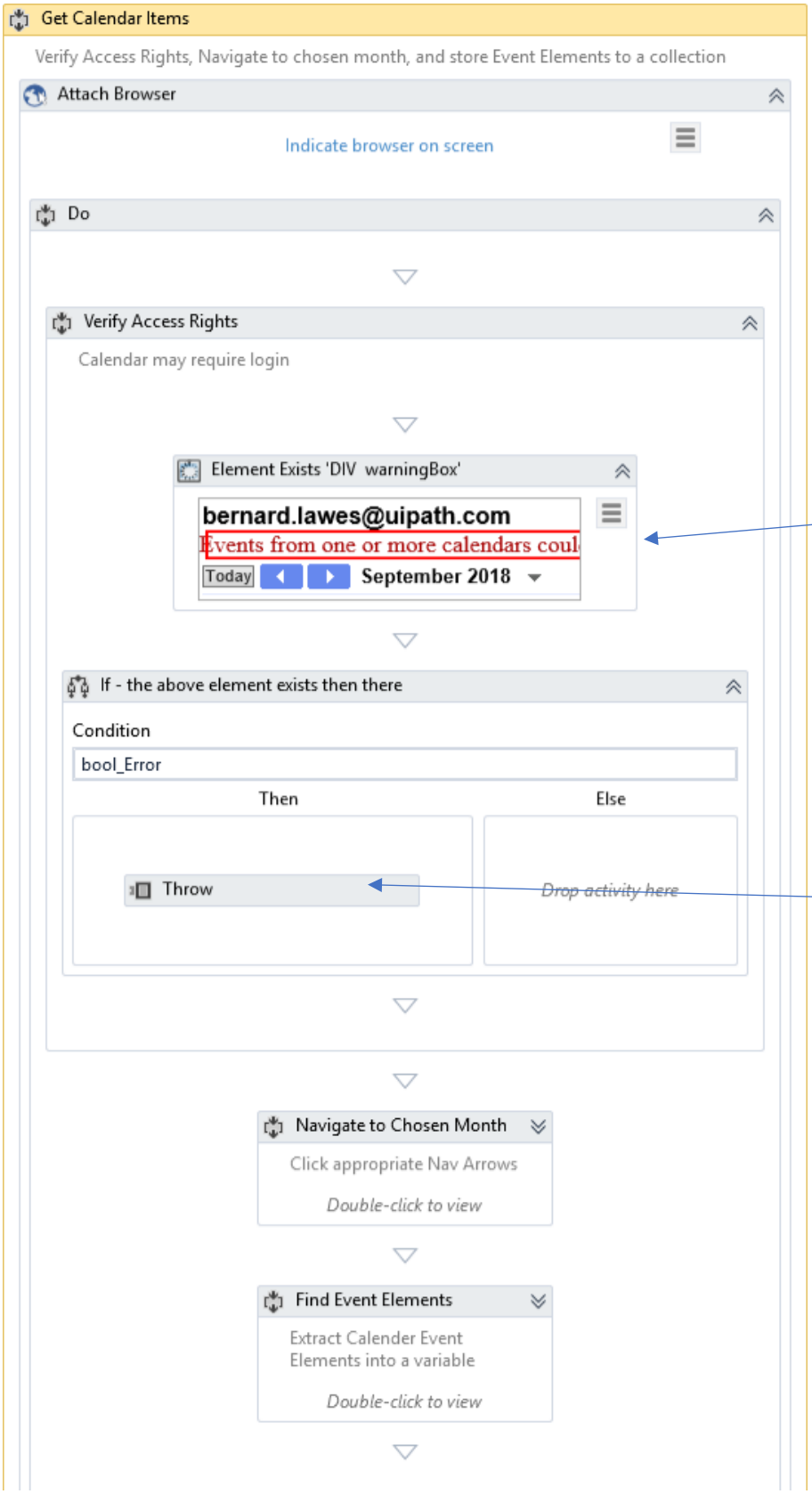
12.Double Click into the **Close Embed Google Cal** and create the following



Set the Assign Activity Value to "<html app='chrome.exe' title='"+str_AccountName+"' />"

Set the Browser Property to browser_GoogleCal

13. Back out to the Flow Chart and double click into **Get Calendar Items Sequence**



Set the Selector Property to

<webctrl id='warningBox' aaname='* permission* ' />

Set the Exception Property to

new businessruleexception("Unauthorized Account. Please Login ")

14. Double Click into Navigate to Chosen Month and Create the following

Navigate to Chosen Month

Click appropriate Nav Arrows

If

Condition

int_MonthOffset <> 0

Then

Else

Sequence

Assign

int_MyCounter = 0

While

Condition

int_MyCounter < math.Abs(int_MonthOffset)

Body

Sequence

If

Condition

int_MonthOffset < 0

Then

Sequence

Highlight

Take informative screenshot

Click 'IMG navBack1'

Image Not Available

Sequence

Highlight

Take informative screenshot

Click 'IMG navBack1'

Image Not Available

Drop activity here

Assign

int_MyCounter = int_MyCounter + 1

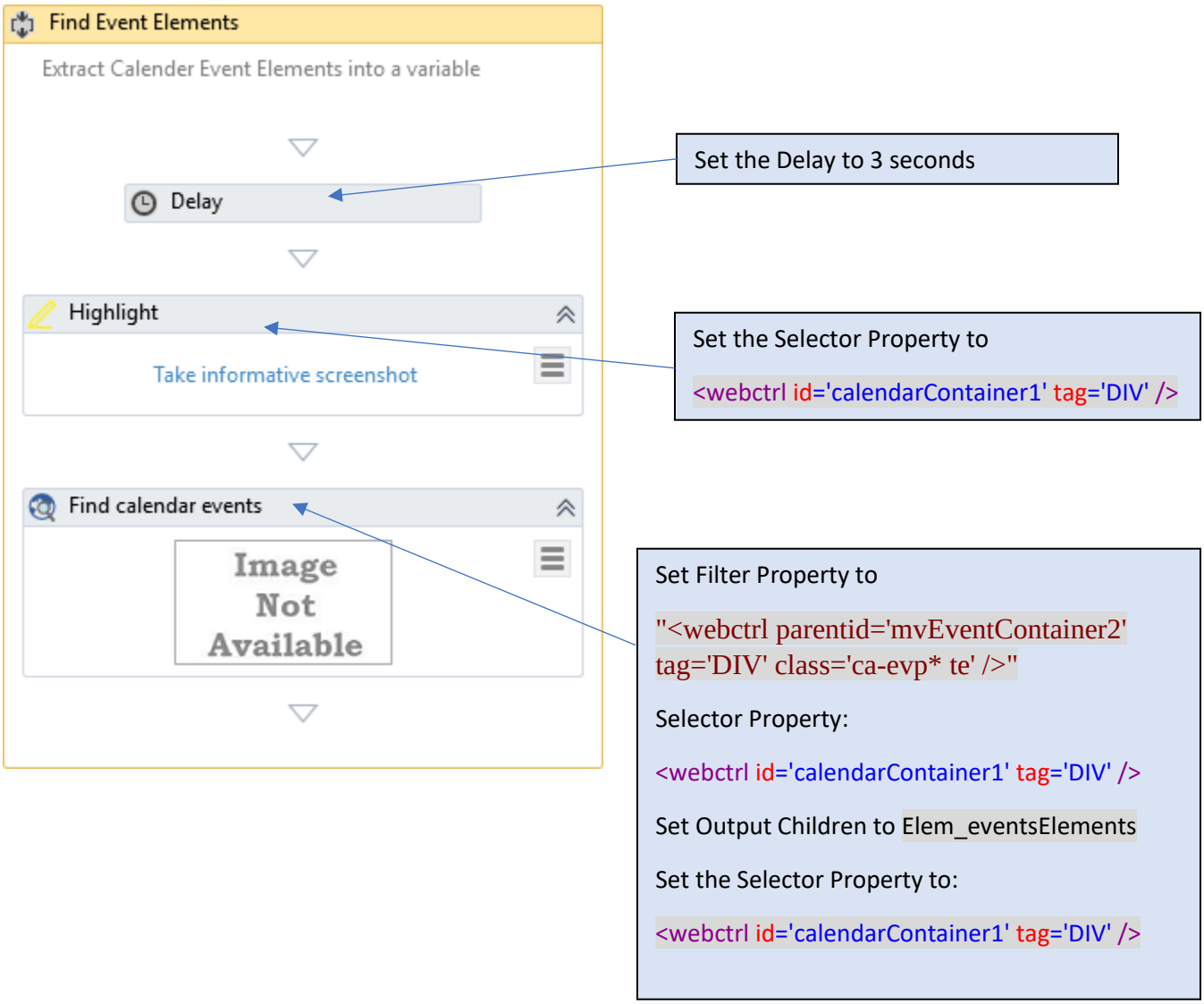
Set the Selector Property to

<webctrl id='navForward1' tag='IMG'

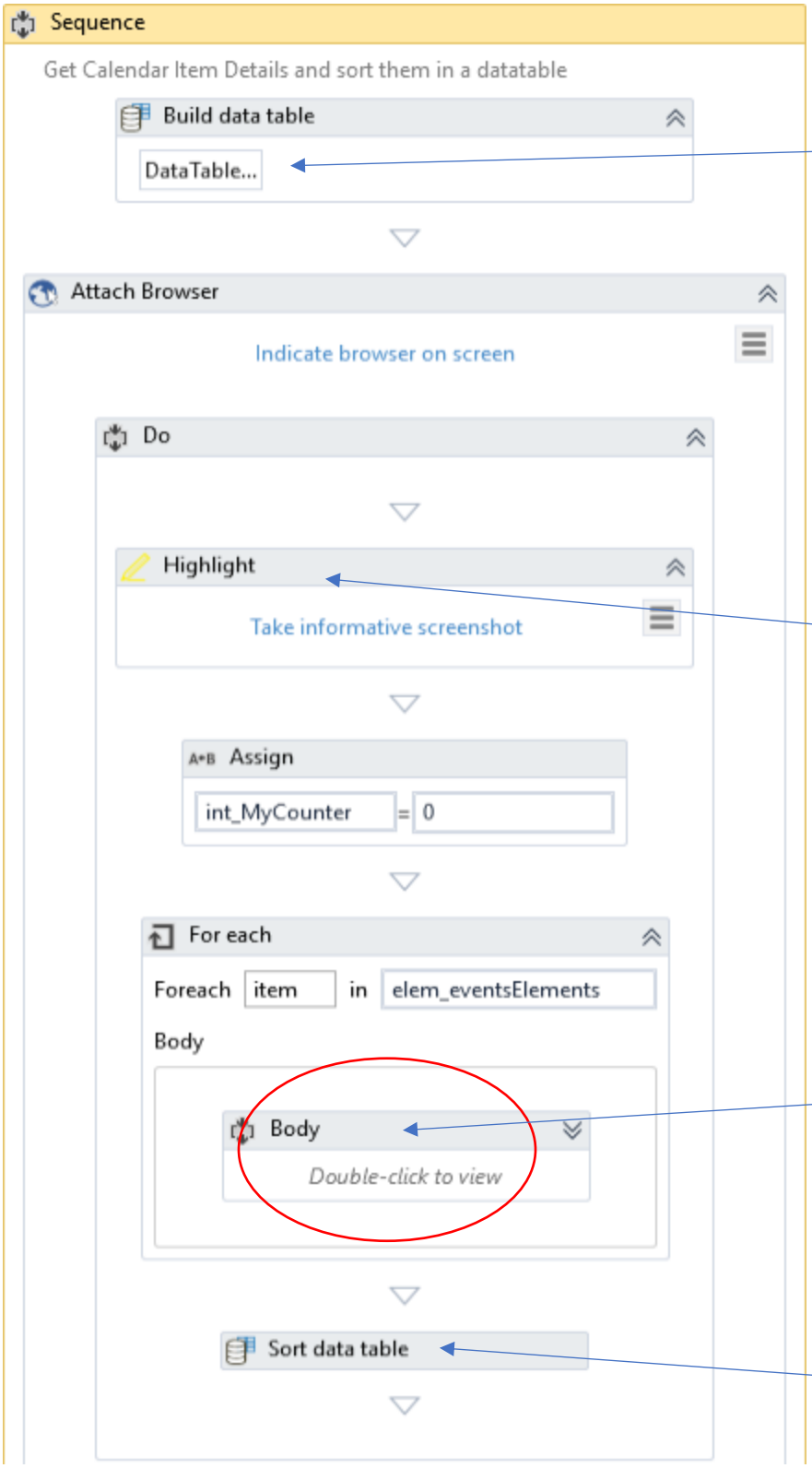
Set the Selector Property to

<webctrl id='navBack1' tag='IMG' />

15. Double click into the Find Elements Sequence and create the following



16. Double click into **Calendar Item Details**



Set Output to datatable_EventInfo

Create a Data Table with the following Fields

EventName (String Type)

EventStart (DateTime Type)

Set the Selector Property to

<webctrl id='calendarContainer1' tag='DIV' />

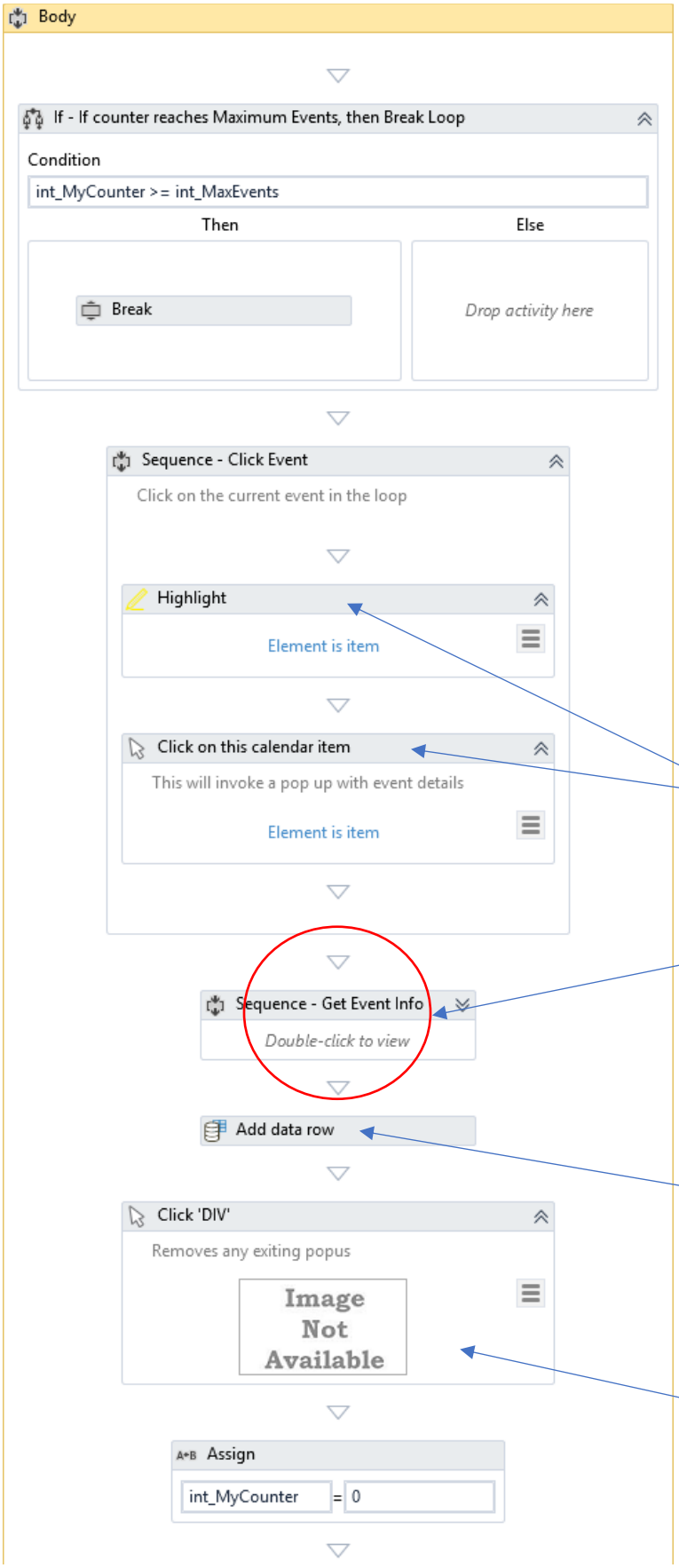
The following page describes what to enter into the body of this For Each Loop

Set Input to: datatable_EventInfo

Set Output to: datatable_EventInfo

Set Sorting Column Index = 1

17. Double Click into the Body of the above For Each loop, and create the following



Set the **Element Property** to Item

The following page describes how to build into this Get Event Info Sequence

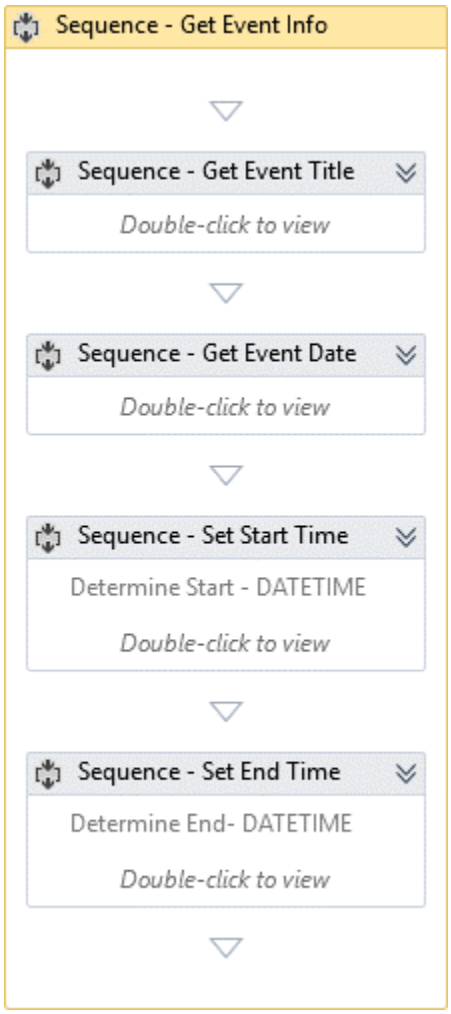
Set ArrayRow: {str_EventTitle, date_StartTime, date_EndTime}

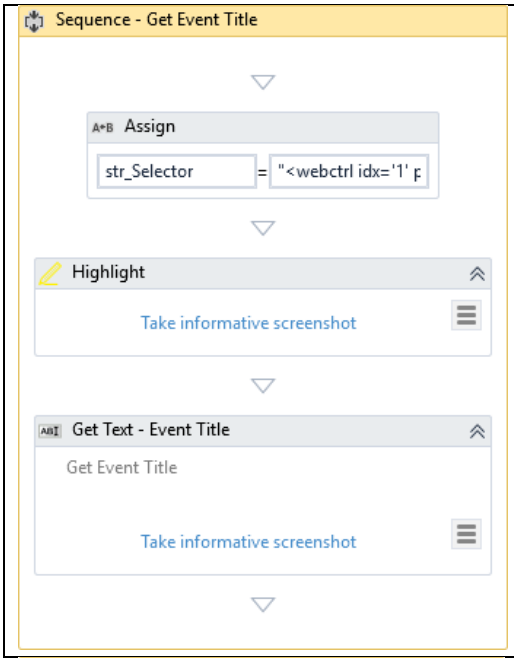
Set Input Datatable to: datatable_EventInfo

Set the Selector Property to

<webctrl tag='BODY' />

18. Double Click into the Get Event Info Sequence and create the following. The following page describes how to build each of the interior sequences



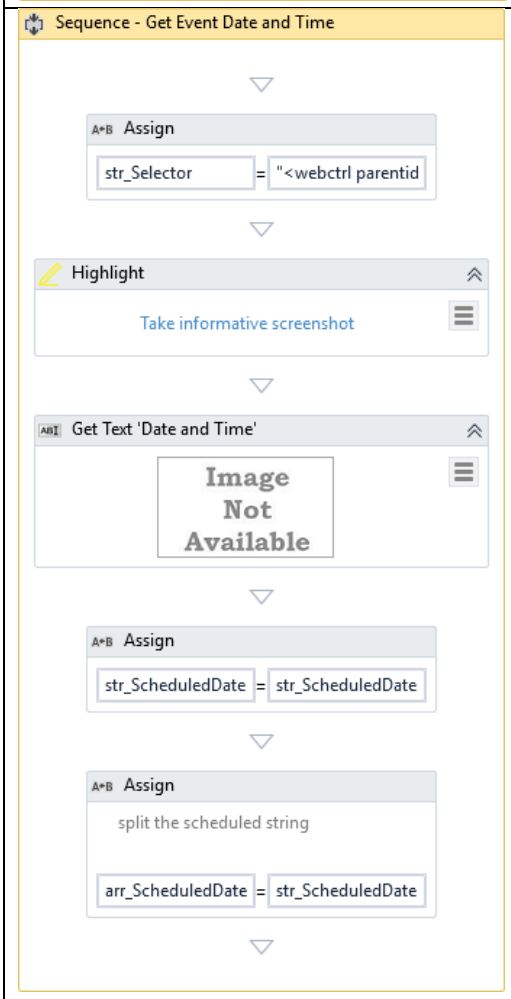


Set Assign Activity value to "<webctrl idx='1' parentid='bubbleContent:1' tag='SPAN' />"

Set Selector Value to str_Selector

Set Selector Value to str_Selector

Set Output Value: str_EventTitle



Set Assign Activity value to ""<webctrl parentid='bubbleContent:1' tag='SPAN' parentclass='detail-item' rowName='"& str_EventTitle.ToString.Replace("""","*").Replace("&","*") &"*' class='event-when' />"

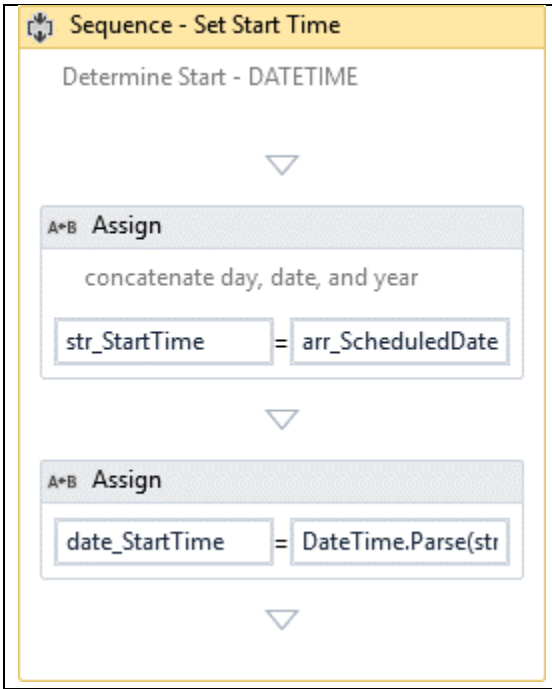
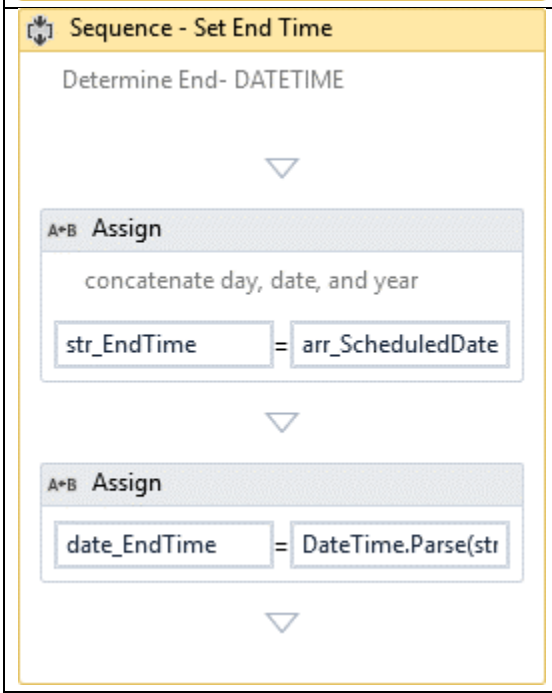
Set Selector Value to str_Selector

Set Selector Value to str_Selector

Set Output Value: str_ScheduledDate

Set Assign Activity Value to str_ScheduledDate.Replace(" – ",",").Replace(str_TodayYear+",","")

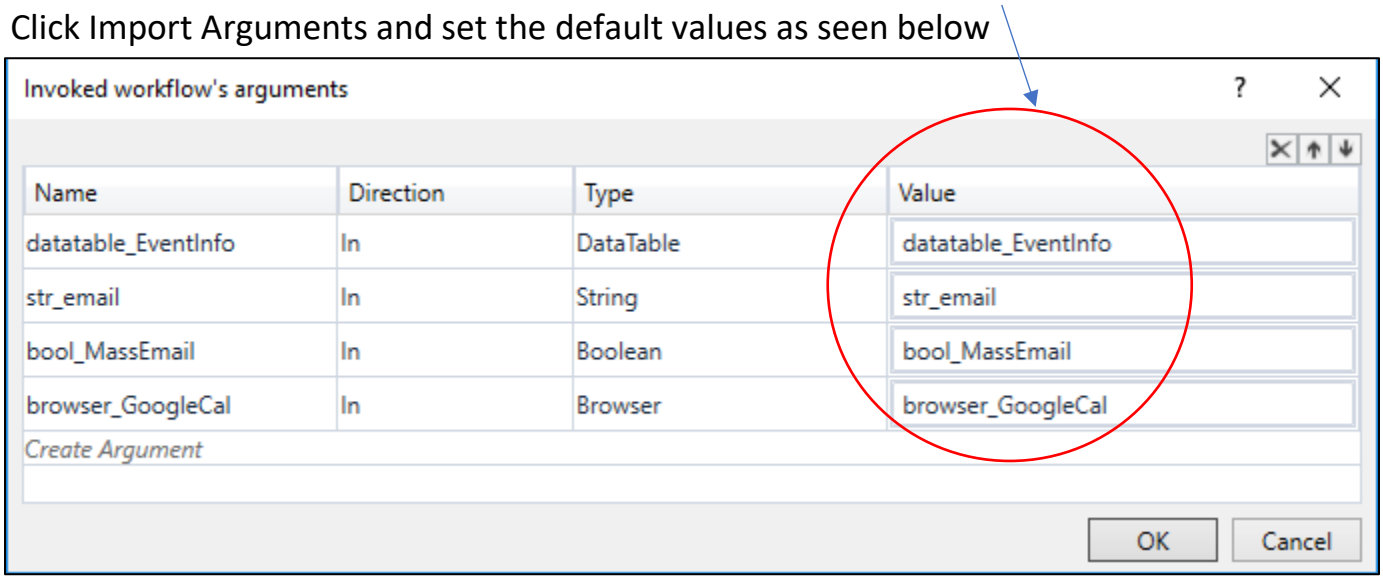
Set Assign Activity Value to str_ScheduledDate.Split(",")

	Set Assign Activity Value to arr_ScheduledDate(0).ToString + "," + arr_ScheduledDate(1).ToString + ", " + str_TodayYear + ", "+ arr_ScheduledDate(2).ToString Set Assign Activity Value to DateTime.Parse(str_StartTime)
	Set Assign Activity Value to arr_ScheduledDate(0).ToString + "," + arr_ScheduledDate(1).ToString + ", " + str_TodayYear + ", "+ arr_ScheduledDate(3).ToString Set Assign Activity Value to DateTime.Parse(str_EndTime)

19. Finally, click the Invoke Workflow File

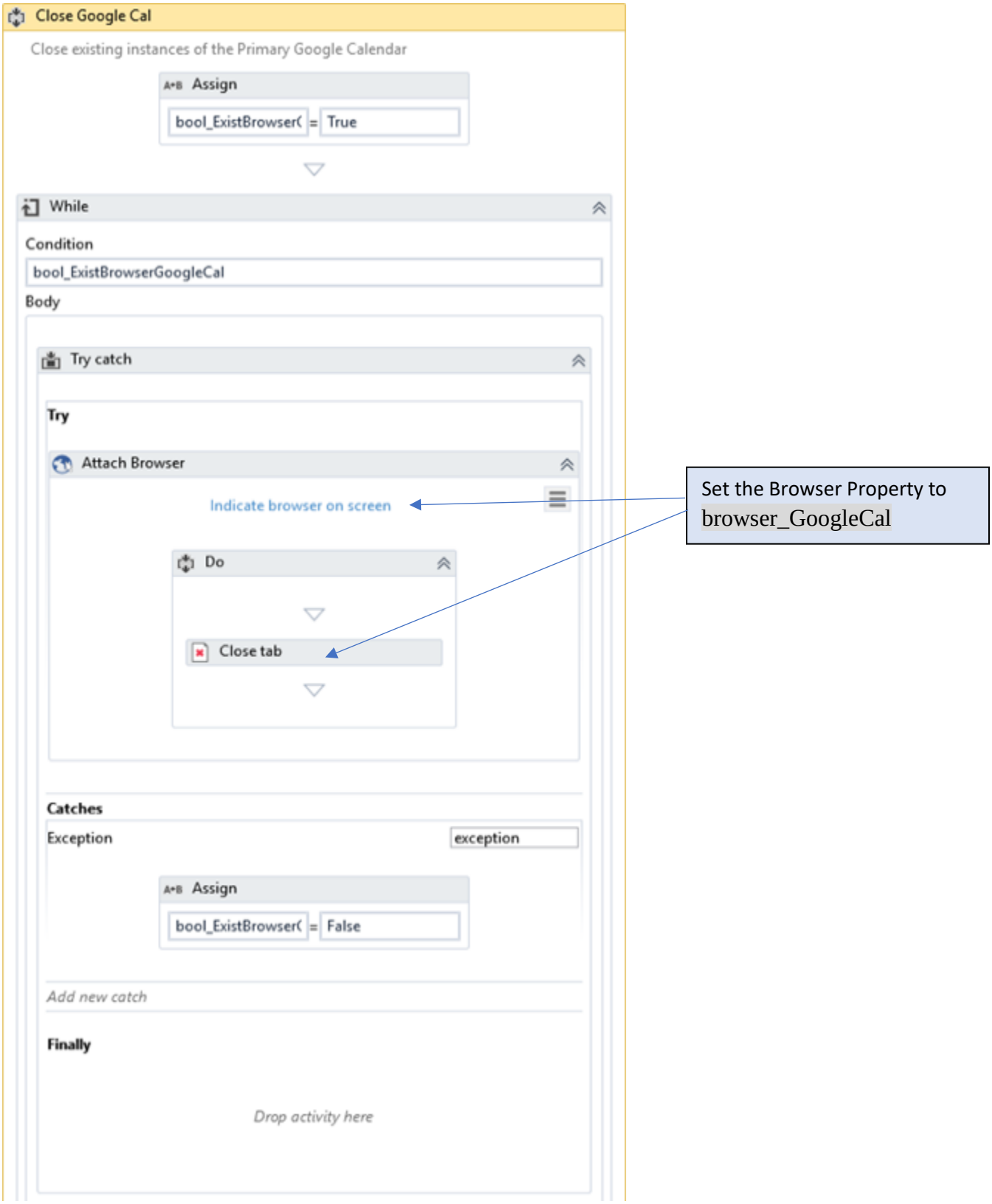
a. Set the WorkflowFileName Property to: "Google_Calendar_Conflict_Executor.xaml"

20. Double click into the Invoke Workflow Activity
21. Click Import Arguments and set the default values as seen below



22. Back out to the Flow Chart, within the Flow Decision Property, set the condition to elem_eventsElements.Count() > 0
23. In the Log Message Activity attached to the False leg of the Flow Decision, set the message to: "No Calendar Conflicts found for specified Period"

24. Again, back out to the Flow Chart Level and Double click into Bottom **Close Google Cal** sequence and create the following



25. Congratulations! You are ready to test drive your bot.

Please note that, when sending to all participants, the SEND button is not actually clicked, it is only highlighted. To actually allow the SEND button to be clicked, you'll need to remove it from the Comment Out Activity which you'll find in the Google "Google_Calendar_Conflict_Executor.xaml"