

#1 TOP RATED ONLINE INSTRUCTOR

LEON PETROU

VB.NET CODING IN

UiPath™

THE ULTIMATE GUIDE
FOR NON-PROGRAMMERS

The Best UiPath RPA Courses

by Leon Petrou

STEP 1: Go from Beginner to Intermediate with UiPath RPA

Complete UiPath RPA Developer Course: Build 7 Robots

[Click here for 95% off Coupon](#)

Free RPA Training / Webinar

[Click here for Free Training](#)

STEP 2: Go from Intermediate to Advanced with UiPath RPA



Complete RPA Bootcamp

[Click here to learn more](#)

Contents

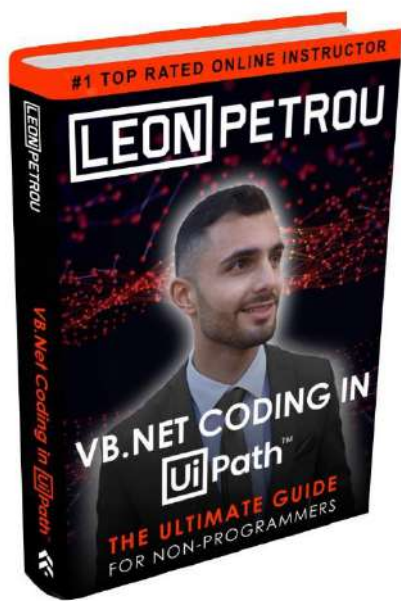
1 Introduction	1
1.1 Why I Wrote this eBook	1
1.2 Why Do We Need to Use VB.NET in UiPath?	1
1.3 What is VB.NET	1
1.4 About the Author	2
1.5 How to Land a High Paying Job as an RPA Developer?	2
1.6 How to Get Free UiPath Support?	2
1.7 Example of Using VB.NET	2
1.8 Terminology	6
1.9 Symbols	6
1.10 VB.NET Reference Cheat Sheet.....	8
2. Excel	9
2.1 How to Count Rows in a DataTable	9
2.2 How to Get a DataTable With Specified Columns	11
2.3 How to Get the Sheet Name From Excel	13
2.4 How to Get Top N Records from a DataTable.....	15
2.5 How to Find the Max and Average Value from a DataTable.....	17
2.6 How to Reorder Columns in a DataTable.....	19
2.7 How to Rename DataTable Columns	22
2.8 How to Convert a DataTable Column to a List.....	24
2.9 How to Only Get the Headers From a Datatable	26
2.10 How to Reverse a DataTable	28
2.11 How to Remove Duplicates from a Datatable	30
2.12 How to Transpose a DataTable	32
2.13 How to Compare Two Excel Files of the Same Format and Get Matched Data	34
2.14 How to Find Common/Uncommon Rows Between Two Databases.....	36
2.15 How to Include a HTML Table in Mail Body with Dynamic Values	39
3 File / Folder Operations	42
3.1 How to Get the Latest File from a Folder.....	42
3.2 How to Delete Multiple Files in a Folder.....	44
3.3 How to Get Specific Types of Files from a Folder	47
4 Text and List Manipulations	49
4.1 How to Read / Write to a Text File.....	49
4.2 How to Define a Dictionary, Add Items To a Dictionary and Access Items From a Dictionary	51
4.3 How to Use Regular Expressions (Regex).....	54

4.4 How to Initialize, Add and Removed Items from a List.....	59
4.5 How to Trim / Clean a String	62
4.6 How to Split a String.....	65
4.7 How to Replace Text in a String	68
4.8 How to Find Common/Uncommon Items Between Two Lists.....	70
4.9 How to Check If a String is Numeric.....	73
4.10 How to Get the Word Count from a String	75
4.11 How to Count the Characters in a String	76
4.12 How to Convert a List to an Array.....	78
4.13 How to Reverse a String.....	80
4.14 How to Convert a SecureString DataType to a String DataType.....	82
5 Email.....	85
5.1 How to Retrieve E-mail Body / Subject / Sender Details	85
5.2 How to Download Specific Files from a Mail	88
5.3 How to Reverse a Mail List.....	91
6 DateTime Operations.....	94
6.1 How to Find the Date Difference (Days, Months and Years).....	94
6.2 How to Compare Time in UiPath	96
6.3 How to Get the Amount of Days in a Month	98
6.4 How to Convert a DateTime to String.....	99
6.5 How to Convert Milliseconds to TimeSpan.....	100
6.6 How to Convert Timestamp to 12/24hrs Format	101
6.7 How to Convert String to DateTime and DateTime Manipulation	102
7 Invoke Code.....	106
7.1 How to Sum a Column Using Invoke Code.....	106
7.2 How to Process and Add a Column Using Invoke Code	109
8 Conclusion.....	112
8.1 Where to Look for VB.NET Code if You Can't Find it in this Guide?.....	112
8.2 How to Take Your UiPath RPA Skills to an Advanced Level?.....	112

1 Introduction

1.1 Why I Wrote This eBook

UiPath is primarily drag-and-drop and does not require much coding. This has attracted many non-programmers to start learning and using UiPath to develop automations. These non-programmers usually reach a hurdle when they need to write some VB.NET code to expand their bots' functionality but have no idea how to write code. Leaving you stuck, confused, and needing to hire expensive developers or learn a new programming language, which not to mention has a steep learning curve for non-programmers. That is why I wrote this book. It is an easy to understand, step-by-step instructional guide on how to use VB.net in UiPath with the most common use-cases of VB.NET in UiPath for industry projects. This book can be used both as a learning resource and as a quick-reference go-to guide for VB.NET while building your automations.



1.2 Why Do We Need to Use VB.NET in UiPath?

Process automation functionality of the UiPath Activities are limited to the Activities provided. However, many processes require additional functionality which are not provided by the suite of activities. Note that the activities in UiPath are simply code packaged into user-friendly activities to speed up the development process. If you need functionality not provided by UiPath Activities, you can most likely do it with VB.NET code. Therefore, we need VB.NET in UiPath to expand the functionality of our automations to build more advanced automations.

1.3 What is VB.NET

VB.NET stands for Visual Basic .NET, and it is a computer programming language developed by Microsoft. VB.NET is an object-oriented programming language (meaning it is made up of data objects, made up of various properties and can perform different functions). Visual Basic runs on the .NET framework, which means that it has full access to the .NET libraries. It is a very productive tool for rapid creation of a wide range of Web, Windows, Office, and Mobile applications that have been built on the .NET framework. We will be using .NET to add additionally functionality to our bots developed in UiPath.

1.4 About the Author

My name is Leon Petrou. I am an engineer and educator. I graduated from the University of Pretoria with a Bachelor of Engineering (BEng) degree with distinction (specialised in Industrial and Systems Engineering). I'm an experienced Robotic Process Automation (RPA) developer, award winning Udemy Instructor and founder at [FutureRPA](#), a smart automation consultancy where we use RPA to streamline our clients' business processes. I'm also the lead instructor of the [Complete RPA Bootcamp](#) where I coach my students to develop their advanced technical skills in RPA so that they can launch their careers as RPA Developers.

1.5 How to Land a High Paying Job as an RPA Developer?

Watch the free training / webinar here: <https://www.completerpabootcamp.com/rpatraining>

1.6 How to Get Free UiPath Support?

Join our private Facebook group to ask any of your RPA / UiPath / VB.NET questions:

<https://www.facebook.com/groups/2481744772043143/>

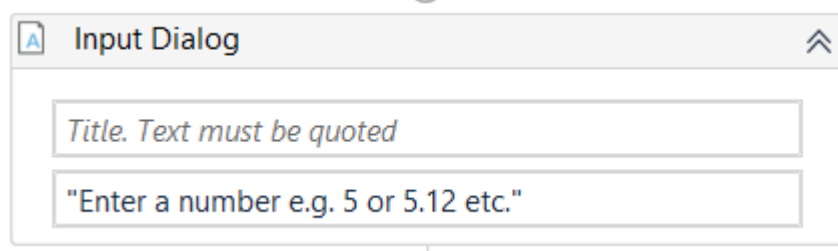
1.7 Example of Using VB.NET

Let's say a user inputs a number, then we want to find the square root of that number and display in a message box like so, "The answer is X". You can download this example here:

<https://drive.google.com/drive/folders/1EI3J8Ej-7Fe6pDiCQyKSNXVVcQW2iq54?usp=sharing>

Let's see how to build this workflow...

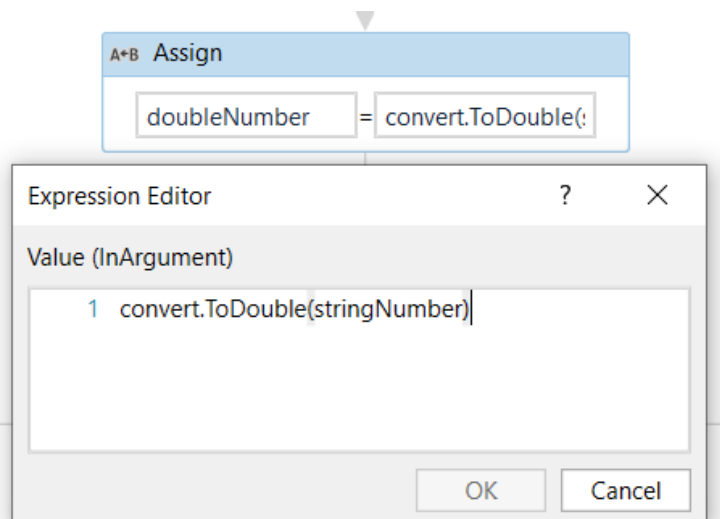
Step 1: We drag in an "Input Dialog" activity for the user to enter a number. We save the number in a string variable.



Step 2: We convert the string to a double (which is a number which can contain decimals and negative values) using an "Assign" activity. We do the conversion because we can perform mathematical calculations with a double but we can't with a text string. We use this code:

```
convert.ToDouble(stringNumber)
```

Note: Replace stringNumber with your string variable



Remember to set the doubleNumber variable to a System.Double type in the variable panel.

Name	Variable type	Scope	Default
stringNumber	String	Sequence	Enter a VB expression
doubleNumber	Double	Sequence	Enter a VB expression
Create Variable			

Step 3: Write this code in a “Message box” activity. Note that a “Message box” activity requires a String input.

"The answer is " + Math.Sqrt(doubleNumber).ToString

In VB.NET we write a string in double quotation marks like this "The answer is " or we can use the .ToString method to convert a variable or expression to a string as we use above.

To concatenate (or combine) two strings together into one, we use a + like we do above.

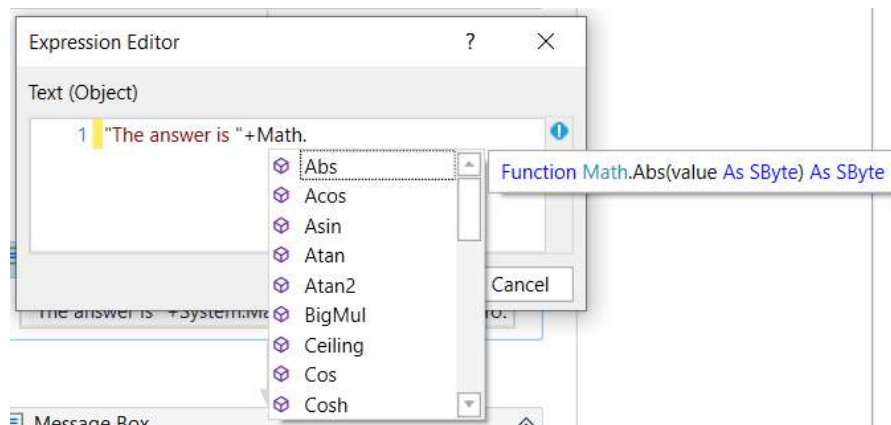
To calculate the square root, we can use the Math function from the “System” namespace.



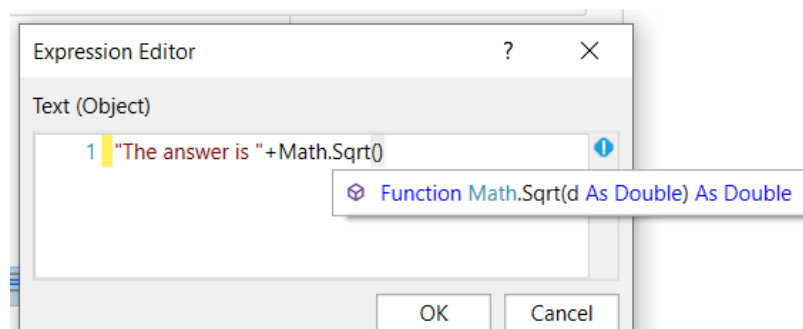
If the “System” namespace is already imported in your Imports panel, then you can simply use the Math function directly. Else, if the System namespace is not already imported you will have to write System.Math

You can import a namespace by simply typing it into the search bar in the 'Imports' panel where it says "Enter or select namespace".

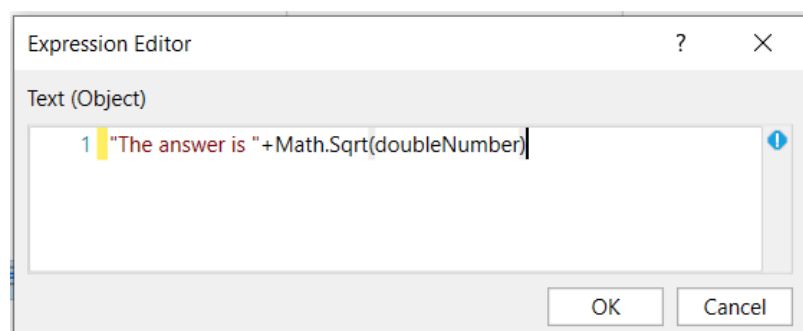
If you enter a period after the math function you will see a list of methods in the math function. Like so:



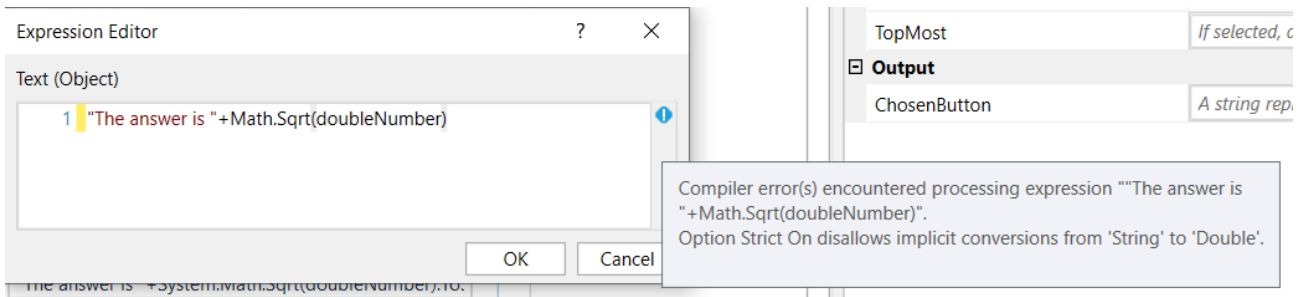
Since we are calculating the square root, we want to select the `Sqrt()` method.



You can see the `Sqrt()` method requires a parameter as a 'double' data type. So, inside the brackets, we add in our `doubleNumber` variable:

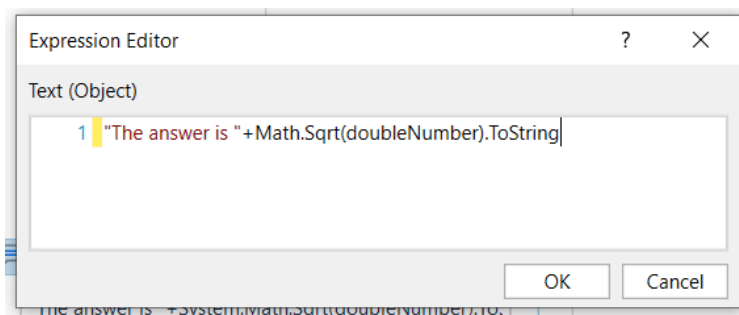


Note that we are getting an error as can be seen by the blue exclamation mark. If we hover our pointer over the exclamation mark, we can read the error message.



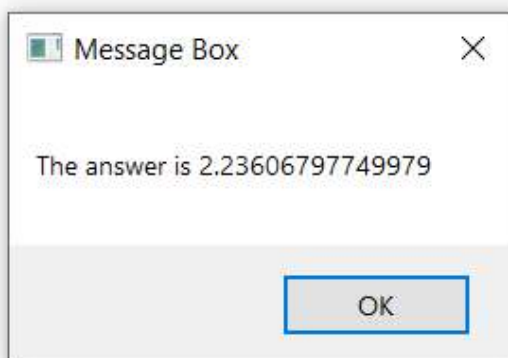
The error above is because the expression `Math.Sqrt(doubleNumber)` is of data type 'Double' but the "Message Box" requires a 'String' data type, so we can simply convert the double to a string using the `.ToString` method like so:

`Math.Sqrt(doubleNumber).ToString`



The error is now gone!

Step 4: We run the workflow, enter a number into the input dialog and see the output in a message box popup. In this case I entered the number 5.



1.8 Terminology

Using the example from above:

```
System.Math.Sqrt(doubleNumber).ToString
```

System is the namespace

Math is the function

Sqrt() is the method of the Math function

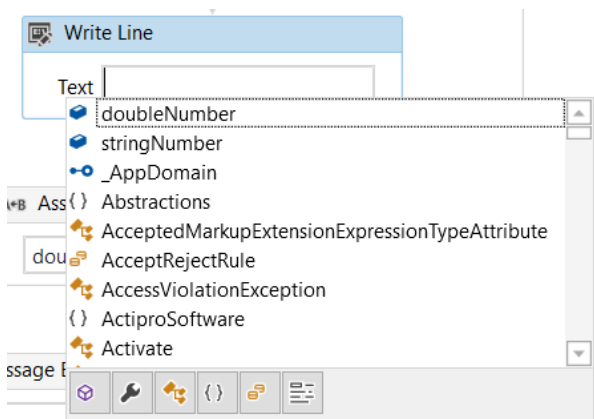
doubleNumber is the parameter (here we entered our doubleNumber local variable)

ToString is also a method

You can learn more about the Math function and methods [here](#) and you can learn more about the .ToString method [here](#).

1.9 Symbols

If you click CTRL+SPACE while in an expression editor in UiPath and intellisense will popup like below, to show all available options. The symbols on the left of the names represent the type of object. It is often useful to know what those symbols represent when coding in VB.NET...



Here are some symbols and their object types:

{ } Namespace

For example

```
.{ } System
```



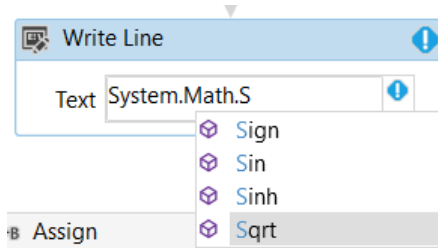
Class (also called a function)

For example



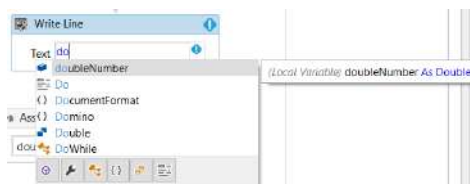
Method

For example



Variable

For example:



There are a few others but those are the most used ones.

1.10 VB.NET Reference Cheat Sheet

Namespace <i>Imports Namespace</i>
Data Types <i>Boolean, Byte, Char, DateTime, Decimal, Double, Int16, Int32, Int64, Integer, Long, Object, Short, Single, String, IntPtr, UInteger, UIntPtr</i>
Variable Declaration <i>Dim Public Private <variable_name> As <type></i>
Type Declaration <i>Dim Public <variable><suffix></i>
Suffixes <i>% - Integer, \$ - String, @ - Decimal, & - Long, # - Double, ! - Single</i>
Arrays <i>Dim <arrayname>(<MaxIndexVal>) As <type></i> <i>Dim <arrayname>(<LowerBound> To <UpperBound>) As <type></i>
Initialize Array <i>Dim <arrayname>() As <type> = {<value1>, <value2>, ... , <valueN>}</i>
Change Size of Array <i>ReDim <arrayname>(<MaxIndexVal>)</i>
Comments <i>'Comment text</i> <i>'No multi-line comments at this time</i>
XML Comments <i>Press the ' (apostrophe) key thrice.</i>
Line Continuation <i>strtext = "To break a long string across multiple lines, " & _</i> <i>"end the string, add the line continuation character " & _</i> <i>"and continue the string on the next line."</i>
Arithmetic Operators <i>+ (Addition), - (Subtraction), * (Multiplication), / (Division), % (Modulus)</i>
String Concatenation <i>+, &</i>
Relational Operators <i>< (Less Than), <= (Less Than or Equal To), > (Greater Than), >= (Greater Than or Equal To), = (Equal To), <> (Not Equal To)</i>
Logical Operators <i>OR, NOT, AND, AndAlso, OrElse, Xor</i>
String Manipulation <i>.Substring(<start>,[<length>])</i> <i>.Trim() <trims from beginning & end of string></i> <i>.TrimEnd([<char array>])</i> <i>.TrimStart([<char array>])</i> <i>.ToLower() <to lower case></i> <i>.ToUpper() <to upper case></i> <i>.Replace(<find>,<replace>)</i> <i>.Equals(<expression>) <6 available overloads></i> <i>.Contains(<string>)</i> <i>.Join(<separator>,<value>,[<count>])</i> <i>.Compare(<string1>,<string2>,[<ignore case>]) <7 overloads available></i> <i>.Copy(<string>)</i>

If Else <i>If(<expression>) Then</i> <i><statement 1></i> <i>Else</i> <i><statement 2></i> <i>End If</i>
Inline If <i>variable = If(condition, value_if_false, value_if_true)</i>
For Loop <i>For <initialize> (Relational Operator) <condition></i> <i><statement></i> <i>Next</i>
For Each Loop <i>For Each <variable> In <object></i> <i>[Exit For]</i> <i><statements></i> <i>[Continue For]</i> <i><statements></i> <i>Next</i>
While Loop <i>While <expression></i> <i><statement></i> <i>End While</i>
Do-While Loop <i>Do</i> <i><statement></i> <i>Loop While <expression></i>
Select Case Statement <i>Select Case <expression></i> <i>Case <expression1></i> <i><statement sequence 1></i> <i>Case <expression2></i> <i><statement sequence 2></i> <i>Case <expressionN></i> <i><statement sequence N-1></i> <i>Case Else</i> <i><statement sequence N></i> <i>End Select</i>
Function Structure <i><Private, Public> <Function_Name>([Parameters])</i> <i>body of the function</i> <i>End Function</i>
Class Structure <i>Public Class <Class_Name></i> <i>body of class</i> <i>End Class</i>
Public <i>'method_prototypes</i> <i>'data_attributes</i> Private <i>'method_prototypes</i> <i>'data_attributes</i> Friend <i>'method_prototypes</i> Shared <i>'method_prototypes</i> <i>'data_attributes</i>
Error Handling <i>Try</i> <i><statements that may cause an error></i> <i>Catch</i> <i><statements to use when an error occurs></i> <i>Finally</i> <i><statements to use no matter what happens></i> <i>End Try</i>

For another VB.NET Cheat Sheet visit this link:

<https://drive.google.com/file/d/1fZbQBSnZq9lwfJNE-5-kQiCbr4H304f9/view?usp=sharing>

Let's now get into the fun and do some examples...

2. Excel

2.1 How to Count Rows in a DataTable

While working with Excel, sometimes it is necessary to find the number of records / rows available in it.

```
total_Count = my_DataTable.Rows.Count
```

Note: my_DataTable is the DataTable you are counting the rows of.

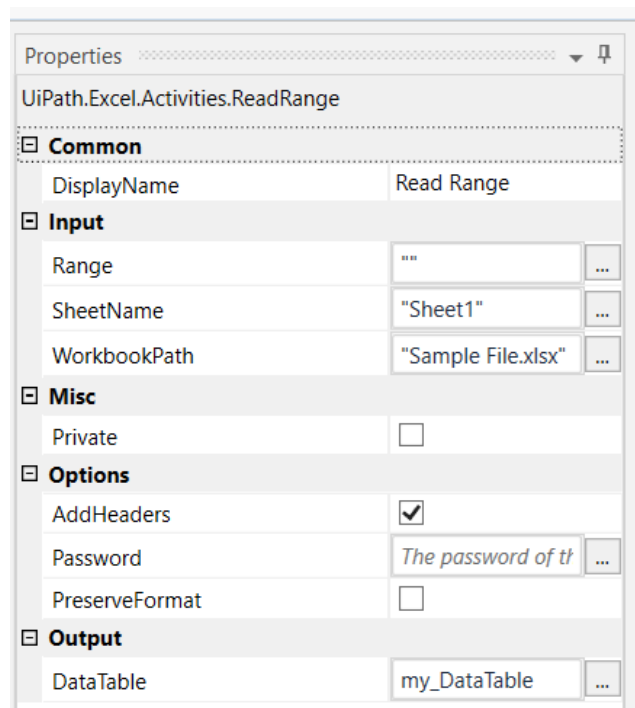
Let us implement a workflow to count the total number of rows in Excel.

Implementation with UiPath

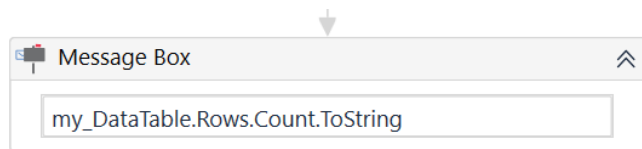
Step 1: Drag the “Read Range” activity from the Workbook library into the design panel and supply the “File Path” to it.



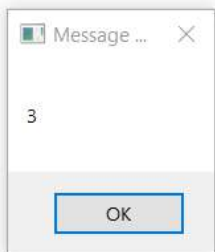
Step 2: Ensure the ‘Add Headers’ property is checked and you have created a new output DataTable variable. In this case we called it my_DataTable



Step 3: Drag a “*Message Box*” activity to display the rows count.



Step 4: Now, run the code. It returns 3 because there are three rows in the Excel Sheet (excluding the header).



Click the link below to download the example project:

https://drive.google.com/open?id=1OgRsFEypWz1G3nTP5f_IFrxXUHtAcpPn

2.2 How to Get a DataTable With Specified Columns

While dealing with data table operations, sometimes we might want to retrieve only specific columns from it. You can do this using the following code:

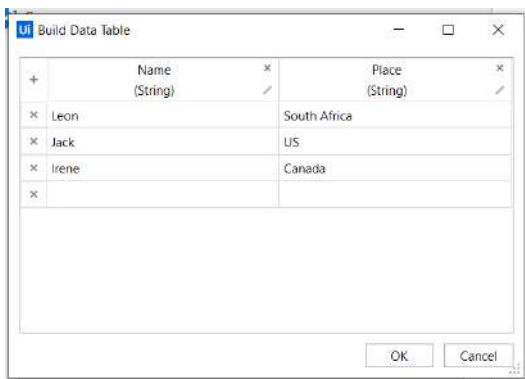
```
dt_New=dt_Original.DefaultView.ToTable(false,"Column1","Column2",...)
```

Note: Simply replace the "Column1","Column2" and so on with your required Columns

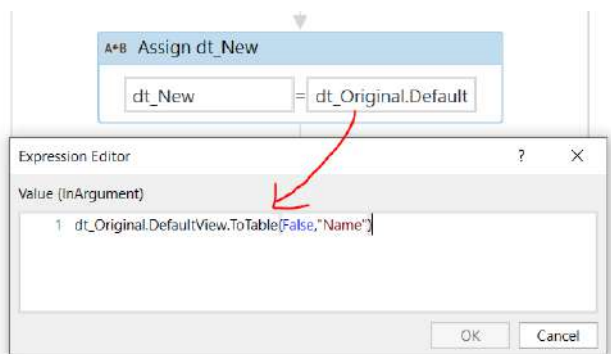
Implementation with UiPath

Let us implement a workflow which takes a "DataTable" and outputs the "DataTable with specified columns". In this case, only the "Name" column.

Step 1: Drag a "Build Data Table" activity into the design panel and populate it with some sample data.



Step 2: Drag an "Assign" activity into the design panel and fill it with above-mentioned code.

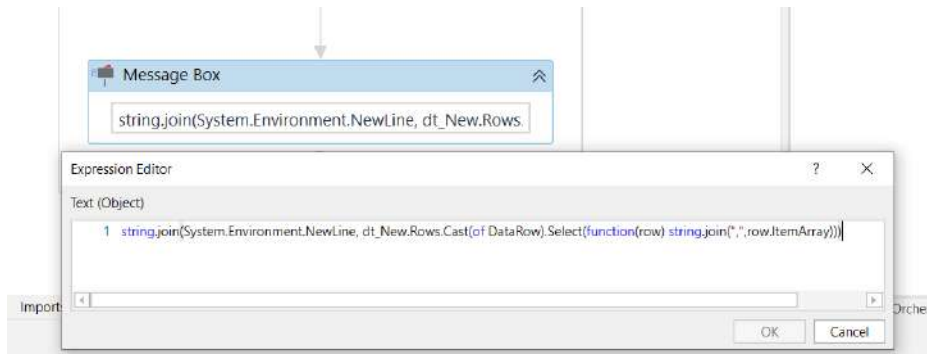


Step 3: Drag a "Message Box" activity to display the result

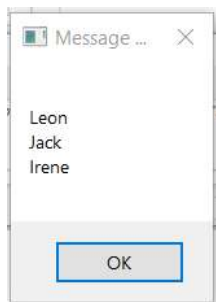
Note: Here we have used LINQ to display the DataTable Instead of Output DataTable. LINQ is Language Integrated Query is a Microsoft .NET Framework component that adds native data querying capabilities to .NET languages.

This expression is equivalent to using an "Output DataTable" Activity (used to convert a DataTable data type to a String DataType).

```
string.join(System.Environment.NewLine, dt_YourDataTable.Rows.Cast(of DataRow).Select(function(row)
string.join(", ",row.ItemArray))))
```



Step 3: Finally, the output:



Click the link below to download the example project:

https://drive.google.com/open?id=1LLC_7bsfnpUQWDVnHHQWZo7tLcfElux

2.3 How to Get the Sheet Name From Excel

Excel is the most used application in the business process automation.

To work with it, we might sometimes want to know the sheets available in it, let us see how to find all the sheet names available in a workbook.

`wb as WorkbookApplication`
`wb.GetSheets`

Implementation with UiPath

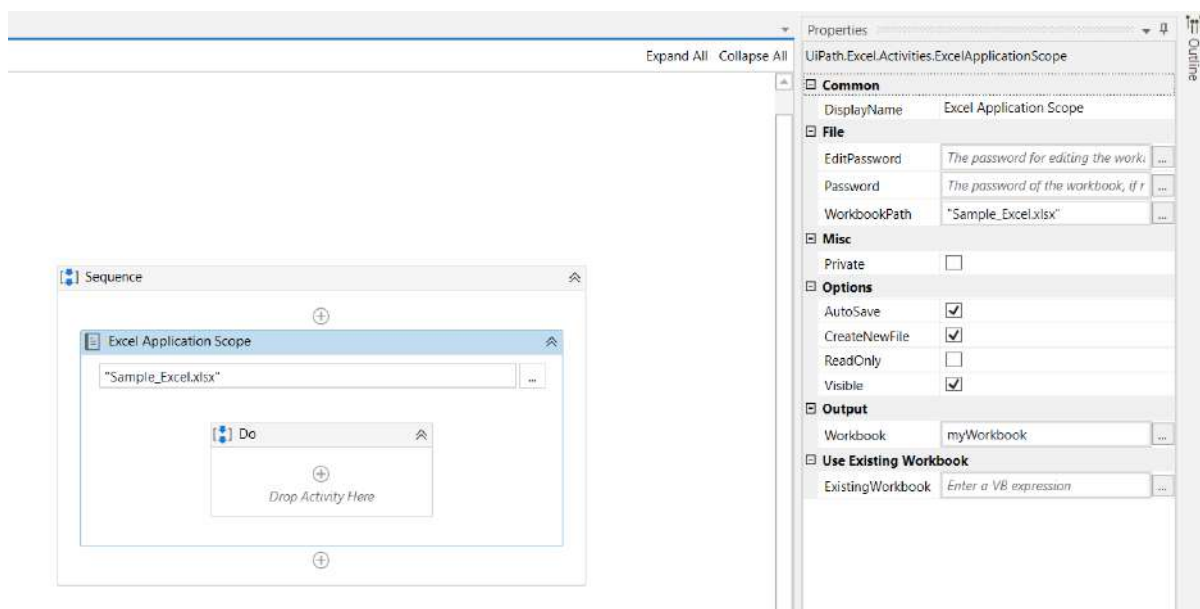
Let us implement a workflow which takes a sample excel file and displays all the sheets available in it.

Step 1: Drag “Excel Application Scope” activity into the designer panel and supply the following values into the activity’s properties panel. Excel needs to be installed on your machine to use this.

Note:

WorkbookPath: “Sample_Excel.xlsx” (String type)

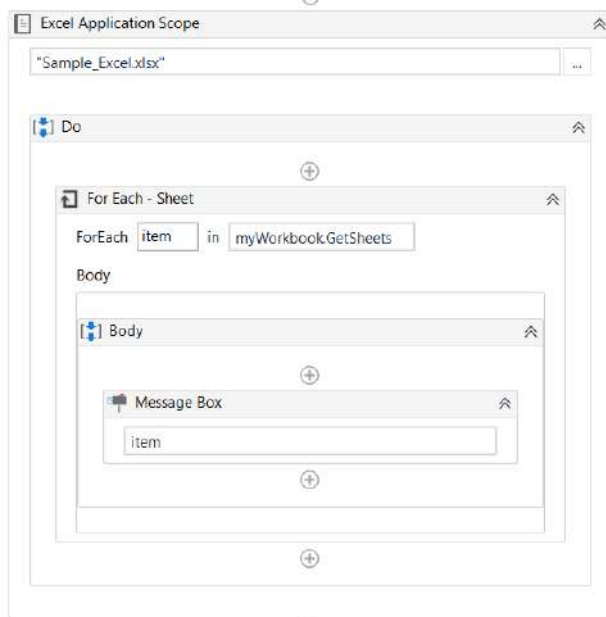
Workbook: myWorkbook (WorkbookApplication type)



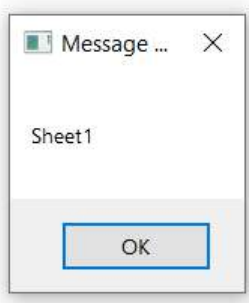
Step 2: Drag “For Each” activity into the designer panel and supply the values to it as shown below.

`myWorkbook.GetSheets`

Step 3: Use “*Message Box*” to display the sheets.



Step 4: Now, execute.



Click the link below to download the example project:

<https://drive.google.com/open?id=1ON0JghAo07tljrjN8R7kaUWN7aRwPXR>

2.4 How to Get Top N Records from a DataTable

Sometimes we might be asked to process only the top N records out of the total available records in the DataTable. Let us see how to implement it...

```
new_DataTable=old_DataTable.AsEnumerable().Take(N).CopyToDataTable()
```

Note:

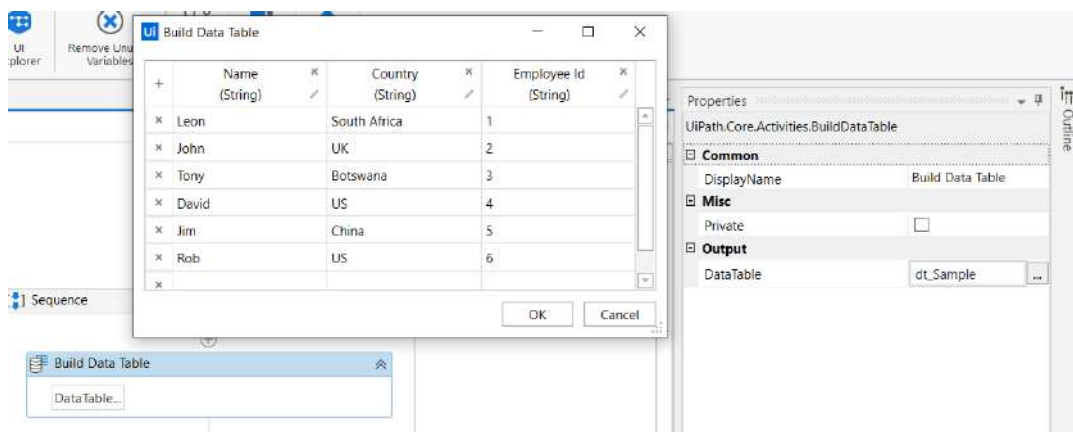
Replace old_DataTable with your DataTable

And replace N with a number e.g. N= 1,2,3....

Implementation with UiPath

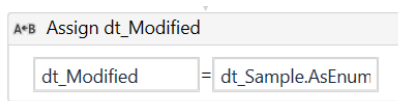
Let us implement a workflow which takes a sample DataTable and displays the top N records of it, according to our supplied input.

Step 1: Drag “Build Data Table” activity into the designer panel and insert some sample data into it.

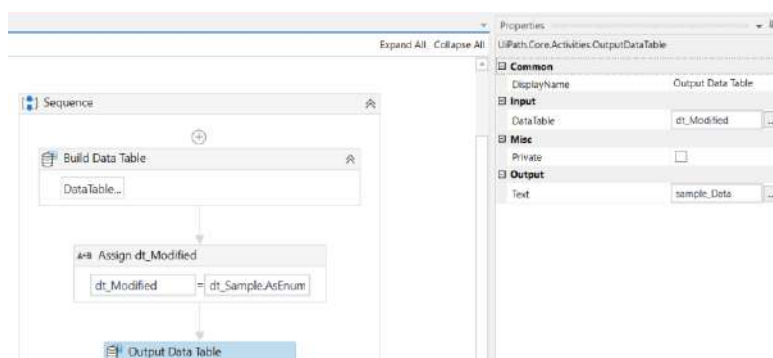


Step 2: Drag “Assign” activity into the designer panel and populate it with the below code to get the top 3 records.

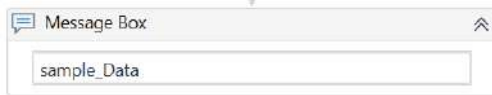
```
dt_Sample.AsEnumerable().Take(3).CopyToDataTable()
```



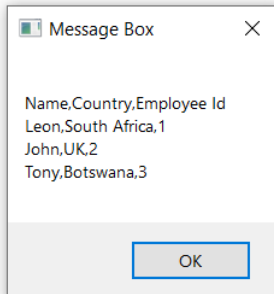
Step 3: Drag “Output Data Table” activity into the designer panel to convert DataTable to String type.



Step 4: Drag “*Message Box*” activity into the designer panel to display the modified data table.



Step 5: Finally, let's run the project.



Click the link below to download the example project:

https://drive.google.com/open?id=1_3sevd7NpAgY2huoNMKWphuQ3amZRYnR

2.5 How to Find the Max and Average Value from a DataTable

Assuming we have a data set and we need to calculate the max and average of a particular column. How do we do it?

MAX Value:

```
myMax_Value = dt_Sample.AsEnumerable().Max(Function(row) cint(row("ColumnName")))
```

AVERAGE Value:

```
myAverageValue = dt_Sample.AsEnumerable().Average(Function(row) cint(row("ColumnName")))
```

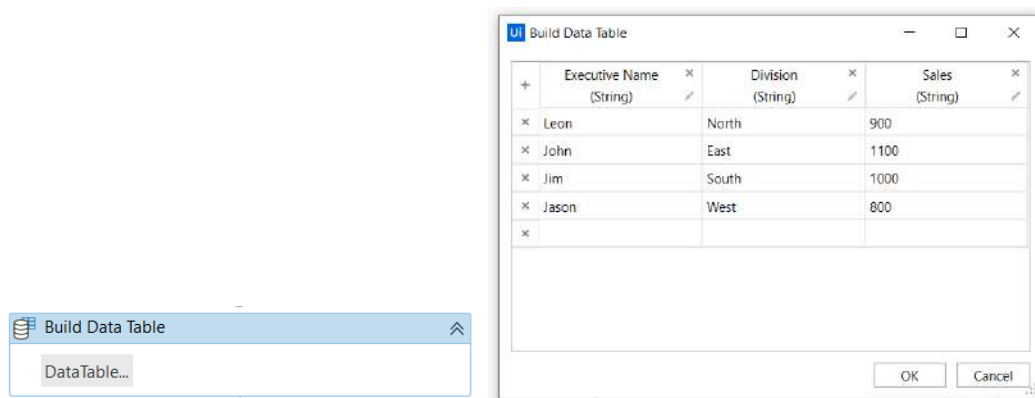
Note:

1. Replace dt_Sample with your DataTable
2. Replace ColumnName with your required column name

Implementation with UiPath

Let us implement a workflow which takes a sample DataTable and displays the Max and Average of a particular column of that DataTable

Step 1: Drag "Build Data Table" activity into the design panel and populate it with some sample data.



Step 2: Drag "Message Box" activity into the design panel to display the Max value of Sales Column and populate it with the below code.

```
dt_Sample.AsEnumerable().Max(Function(row) cint(row("Sales")))
```



Step 3:

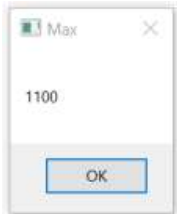
Drag "Message Box" activity into the design panel to display the Average value of Sales column and populate it with the below code.

```
dt_Sample.AsEnumerable().Average(Function(row) cint(row("Sales")))
```

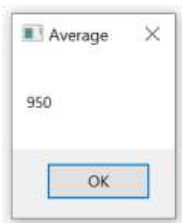


Step 4: Finally, let's run the project.

Max Value:



Average Value:



Click the link below to download the example project:

<https://drive.google.com/open?id=1JgHyueNIR59U8tJVUfMr31w67G2cKKw>

2.6 How to Reorder Columns in a DataTable

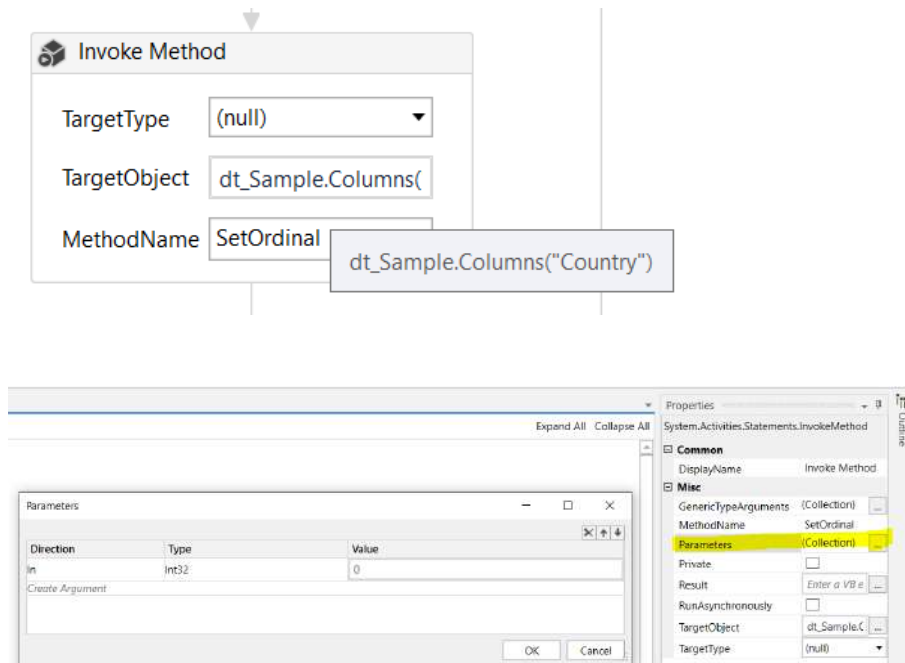
Sometimes we need to change the order of the columns in excel and then process the data.

```
dt_Sample.Columns("ColumnName").SetOrdinal(Parameter)
```

Note:

1. *ColumnName* – The name of the column for which we want to change the position.
2. *Parameter* – The position or index to which we want to move the column to (Indexing starts from zero).

SetOrdinal method can be used to achieve the above using an “Invoke Method” activity.



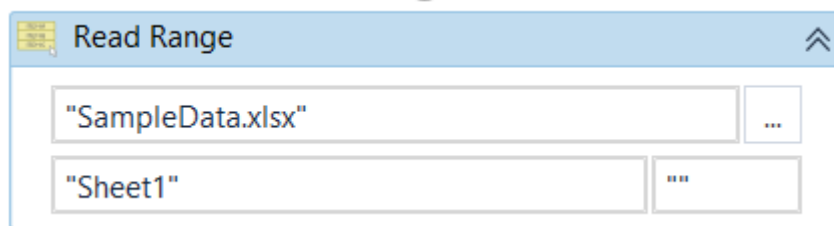
This is the same thing as:

```
dt_Sample = dt_Sample.Columns("Country").SetOrdinal(0)
```

Implementation with UiPath

Let us implement a workflow which reads an excel file and then we will try to reorder the columns.

Step 1: Drag a Workbook “*Read Range*” activity into the design panel to read the excel file into a Data Table.



Step 2: Drag “*Invoke Method*” activity into the design panel and populate it with the below-mentioned code.

Note:

TargetObject – dt_Sample.Columns(“Country”)

MethodName – SetOrdinal

In Parameter – 0 (0 indicates first column position)

Invoke Method Country to Idx 0

TargetType: (null)

TargetObject: dt_Sample.Columns(

MethodName: SetOrdinal

Parameters

Direction	Type	Value
In	Int32	0

Create Argument

OK Cancel

Drag another “*Invoke Method*” activity into the design panel and populate it with the below-mentioned code.

Note:

TargetObject – dt_Sample.Columns(“City”)

MethodName – SetOrdinal

In Parameter – 1 (1 indicates second column position)

Invoke Method City to Idx 1

TargetType: (null)

TargetObject: dt_Sample.Columns(

MethodName: SetOrdinal

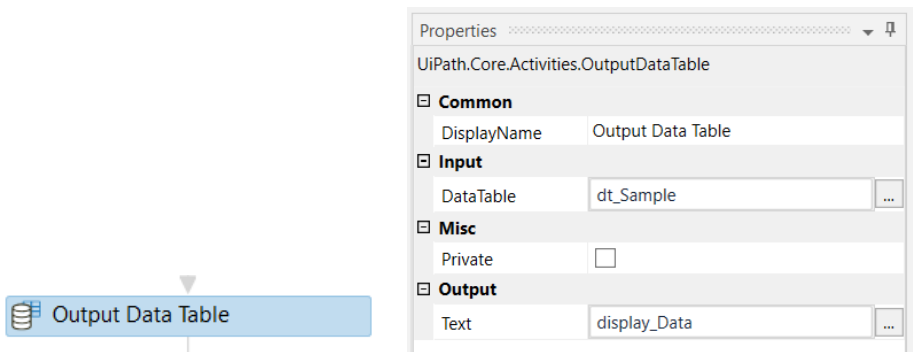
Parameters

Direction	Type	Value
In	Int32	1

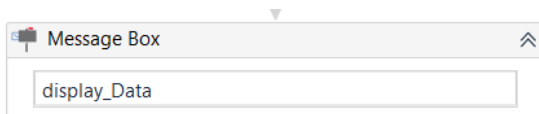
Create Argument

OK Cancel

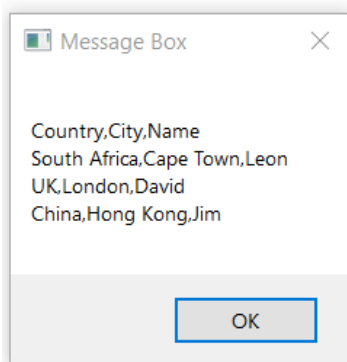
Step 3: Drag “Output Data Table” activity into the design panel and convert the DataTable to String type.



Step 4: Drag “Message Box” activity into the design panel to display the modified data table.



Step 5: Finally, we execute the workflow. We can see the modified DataTable with reordered column names. The original order was Name, Country, City. Now the new order is Country, City, Name



Click the link below to download the example project:

https://drive.google.com/drive/folders/136dTOAt9GMI3e8ycxISufwXYuR_zQfzH?usp=sharing

2.7 How to Rename DataTable Columns

In RPA as we often work with excel, there are so many operations that we usually perform on it.

What if we are asked to rename a column of a DataTable?

Let us see how to do it!!!

```
dataTable.Columns("old_ColumnName").ColumnName = "new_ColumnName"
```

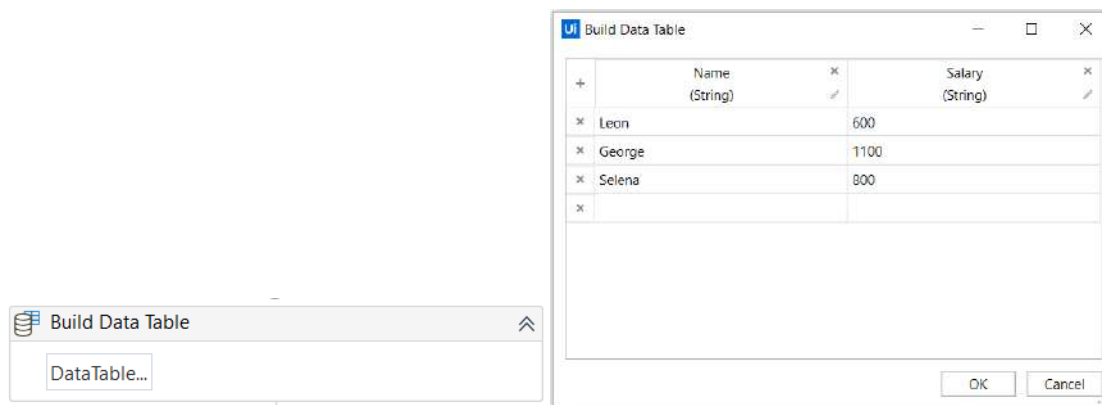
Note:

1. old_ColumnName is the name of the specific column that you want to rename
2. new_ColumnName is the column name with which you want to rename the above old column name with.

Implementation with UiPath

Let us implement a workflow which takes a sample DataTable as input and we will try to *rename an existing column* of it from "Name" to "FirstName"

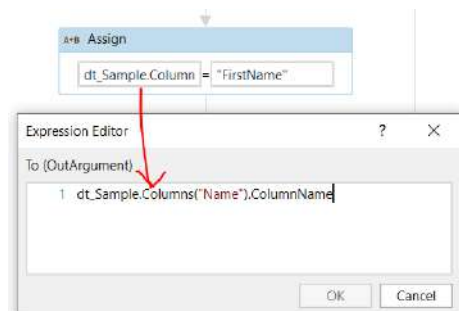
Step 1: Drag "Build Data Table" activity into the design panel and populate it with some sample data.



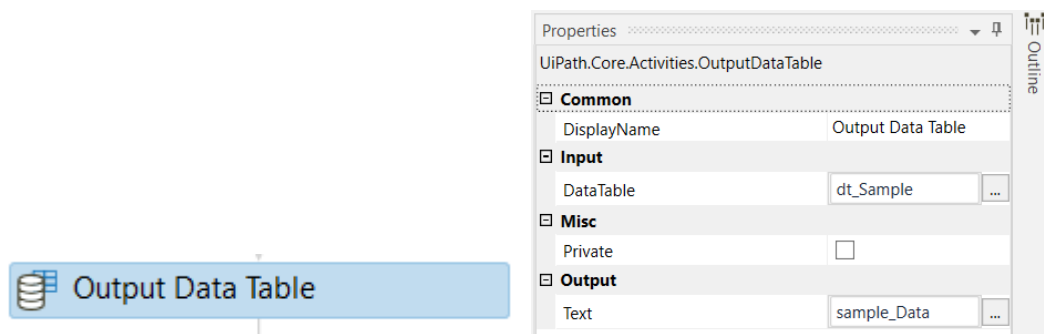
Step 2: Drag an "Assign" activity into the design panel and fill it with above-mentioned code.

Note:

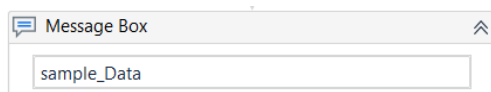
We are renaming "Name" column of above build DataTable to "FirstName"



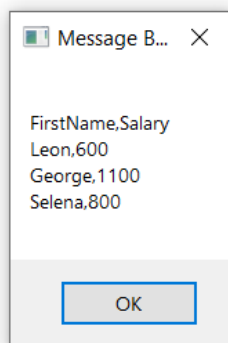
Step 3: Drag “Output Data Table” activity into the design panel and convert DataTable to String type.



Step 4: Drag “Message Box” activity into the design panel to display the modified data table.



Step 5: Finally, let’s run the project.



Click the link below to download the example project:

<https://drive.google.com/open?id=1K26DHdDYFgRffuyq3qqIOQ4-D9vQ1czw>

2.8 How to Convert a DataTable Column to a List

We may be required to take the values from a column of a DataTable and then convert it to a list so that we can perform further actions or decision flows.

Let us see how to do it in UiPath?

```
MyList = ( From row in dt_Sample.AsEnumerable() Select Convert.ToString(row("ColumnName"))) ).ToList()
```

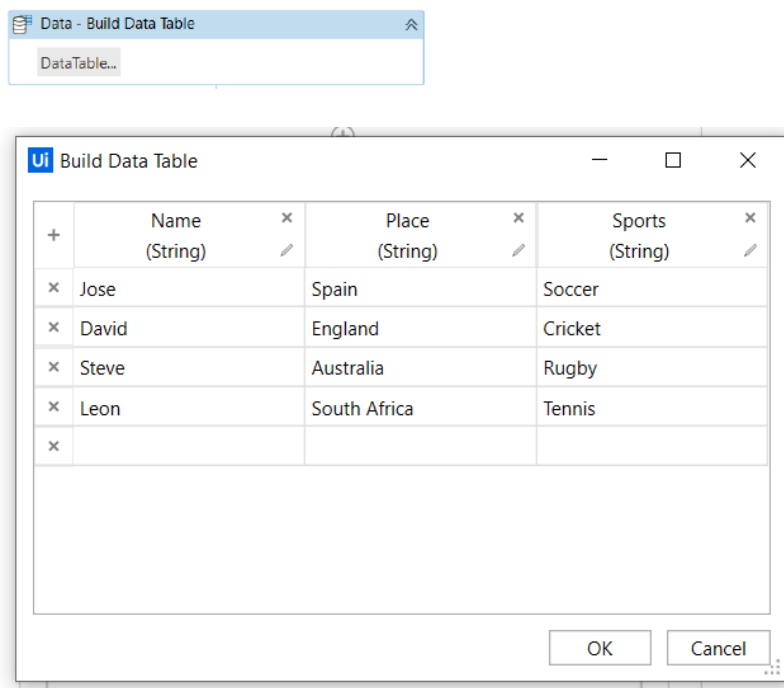
Note:

1. ColumnName is the specific column name which you want to convert to list
2. MyList is the variable containing your values

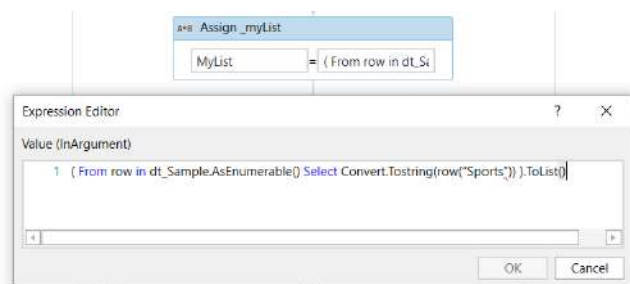
Implementation with UiPath

Let us implement a workflow which takes a sample data table as input and converts a particular Column's values into a list of string.

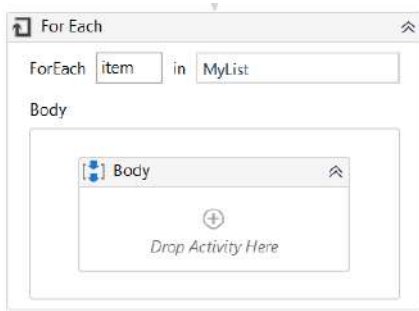
Step 1: Drag "Build Data Table" activity into the design panel and populate it with some sample data.



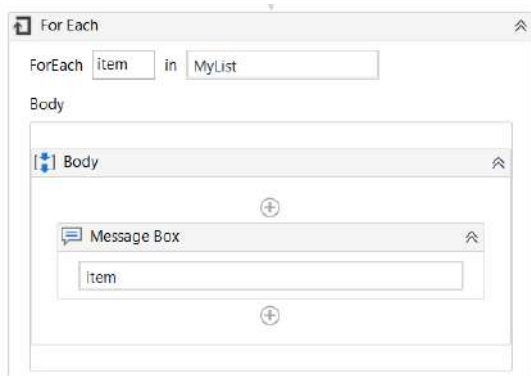
Step 2: Drag "Assign" activity into the design panel and fill it with above-mentioned code.



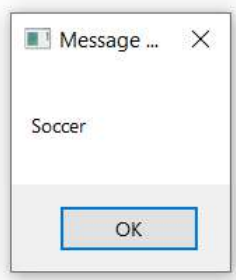
Step 3: Drag "For Each" activity into the design panel and pass the above "MyList" variable to it.



Step 4: Drag “*Message Box*” activity into the For Each activity and pass the “*item*” value into it.



Step 5: Finally, let's run the bot.



Now, we can see all the values in the columns being displayed by clicking OK.

Click the link below to download the example project:

<https://drive.google.com/open?id=1iOaeCwh18p0EJ8td6kudYIBnPNIM67Vj>

2.9 How to Only Get the Headers From a Datatable

Sometimes you might want to copy the schema of one excel to another excel.

```
dt_New = dt_Original.Clone
```

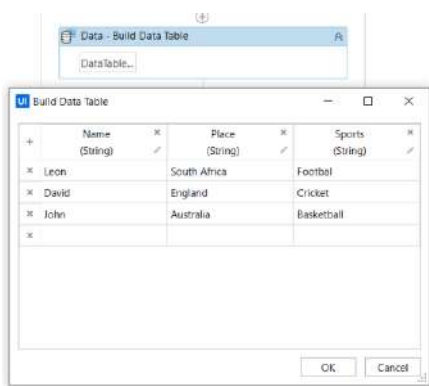
This only clones the structure of the DataTable, i.e. the headers. It doesn't copy the content of the DataTable, to do that you would use the Copy Method like so:

```
dt_New = dt_Original.Copy
```

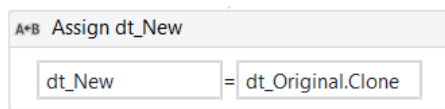
Implementation with UiPath

Let us implement a workflow which takes a “*Data Table*” and displays the “*Headers*” of it.

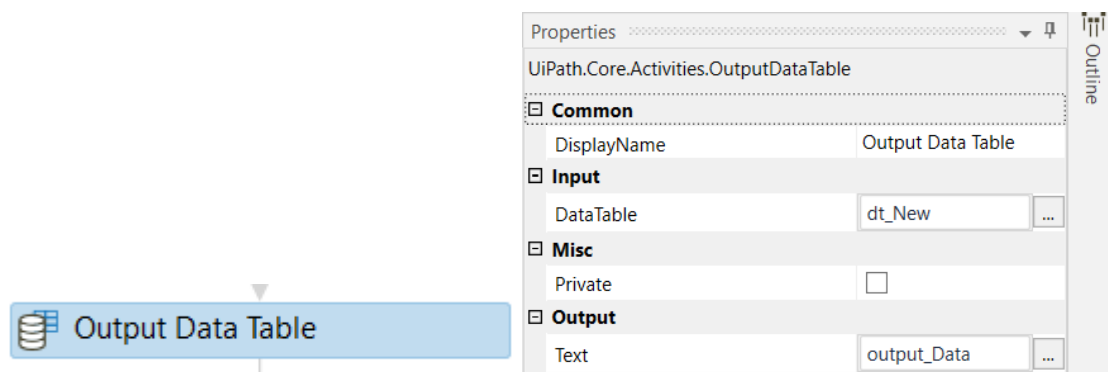
Step 1: Drag “Build Data Table” activity into the design panel and populate it with some sample data.



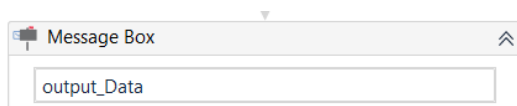
Step 2: Drag an “Assign” activity into the design panel and fill it with the above-mentioned code.



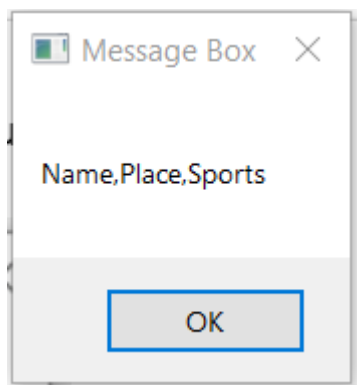
Step 3: Drag “Output Data Table” activity into the design panel and convert the DataTable type to String type.



Step 4: Drag a “Message Box” activity into the design panel to display the headers.



Step 5: Finally, let's run the project.



Click the link below to download the example project:

<https://drive.google.com/open?id=1uRdAHkqWojMstGQ562FWF3oqaBtdcmLJ>

2.10 How to Reverse a DataTable

We may come across situations where we need to process the records available in excel or a data table in reverse.

So let's explore how to do that in UiPath!!!

```
dt_Reverse = dt_Original.AsEnumerable.Reverse.CopyToDataTable
```

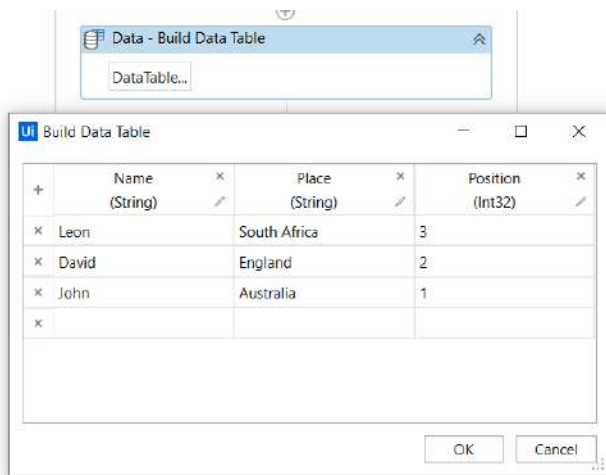
Note:

1. Simply replace the “dt_Original” with your required “DataTable”

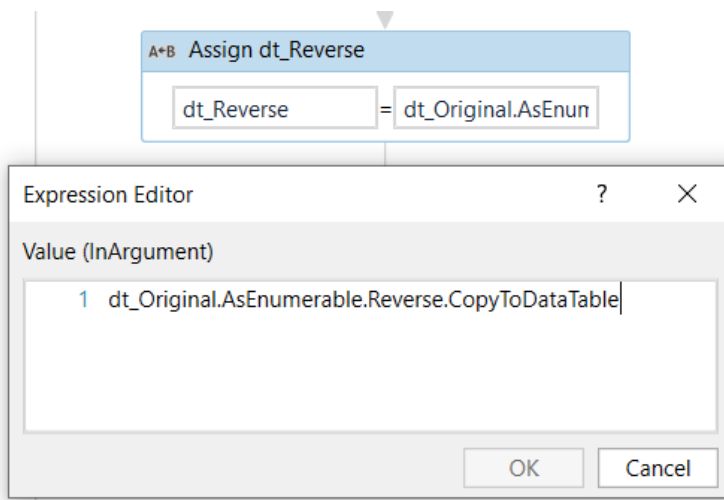
Implementation with UiPath

Let us implement a workflow which takes a sample data table as input and outputs the reverse of it.

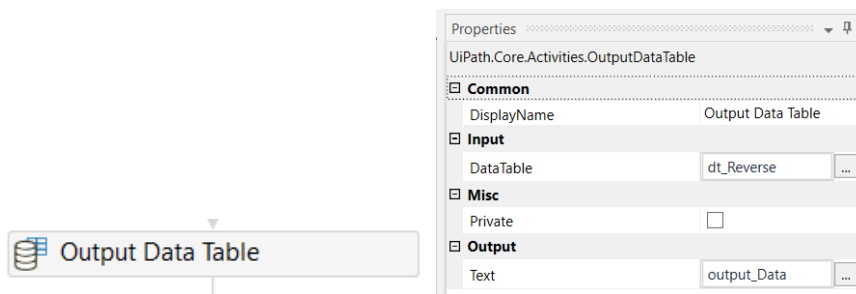
Step 1: Drag a “Build Data Table” activity into the design panel and populate it with some sample data.



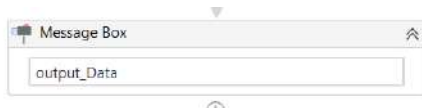
Step 2: Drag “Assign” activity into the design panel and fill it with above-mentioned code.



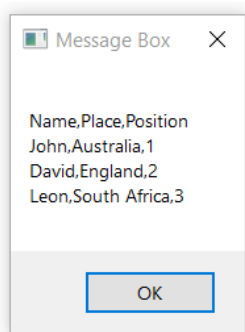
Step 3: Drag “*Output Data Table*” activity into the design panel and convert Datatable to String type.



Step 4: Drag “*Message Box*” activity into the design panel to display the reversed data table.



Step 5: Finally, we run the project.



Click the link below to download the example project:

<https://drive.google.com/open?id=1V4KvRcwonDDzUd6lhcnS-DP-uyREIFMh>

2.11 How to Remove Duplicates from a Datatable

Sometimes a data table will contain duplicate records which you want to remove to only keep unique records.

We can use the below LINQ query to achieve it.

```
dataTable new_dt =  
old_dt.AsEnumerable().GroupBy(Function(i) i.Field(Of String)("ColumnWithDuplicate")).Select(Function(g)  
g.First()).CopyToDataTable
```

Note:

This is useful when we are removing duplicates with respect to some "Column" name. Simply replace the "ColumnWithDuplicate" with your required column name

OR

We can use the simple DefaultView Code too...

```
dataTable new_dt = old_dt.DefaultView.ToTable(True)
```

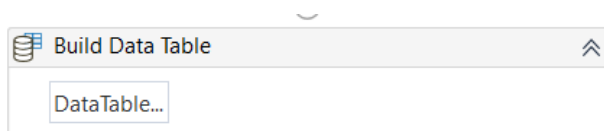
Note:

Useful when we are removing duplicates with respect to every "Column"

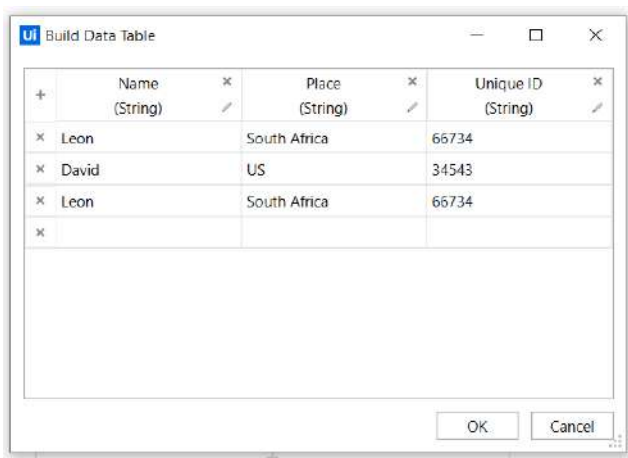
Implementation with UiPath

Assume we have sample data, which includes information about 3 customers.

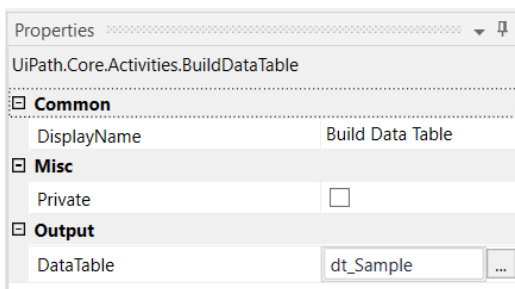
Step 1: Drag a "Build Data Table" activity into the design panel



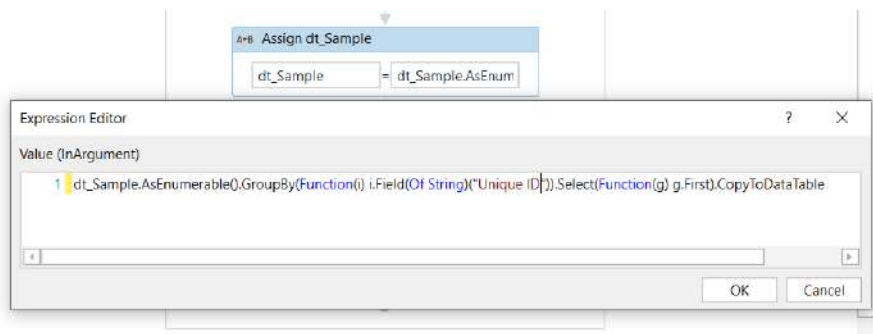
Step 2: Populate the "Build Data Table" activity with sample data



Step 3: Create a variable of DataTable type with some name and supply it to the Output property of “Build Data Table” activity

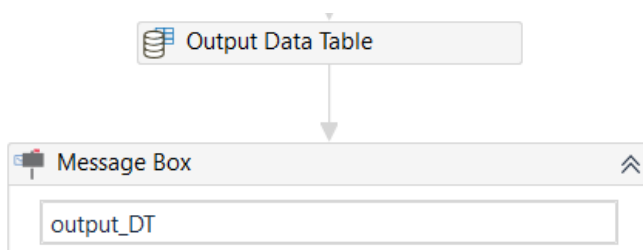


Step 4: Drag “Assign” activity into the design panel and in the ‘Value’ property of that activity use the above code with the assigned column name.

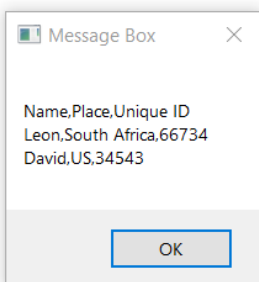


Note: Here we are using the “Name” column to remove the duplicates from the data table.

Step 5: Drag an “Output Data Table” and “Message Box” activity into the design panel and supply the appropriate parameters to display the data.



Step 6: Run the workflow.



Click the link below to download the example project:

https://drive.google.com/open?id=1eN9M1_usqzO_k5d_FNJQga2JBZtGRTWC

2.12 How to Transpose a DataTable

Sometimes while working with DataTables, we might need to convert rows to column and vice versa.

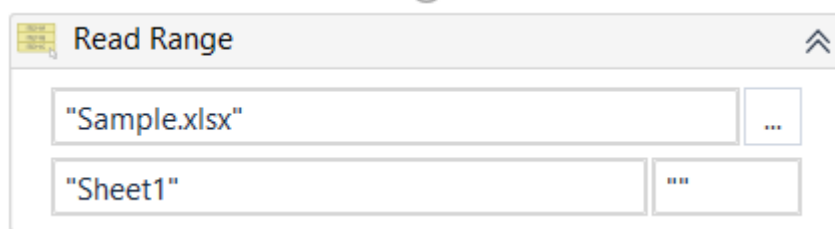
Implementation with UiPath

Let us implement a workflow to convert rows to columns in a DataTable.

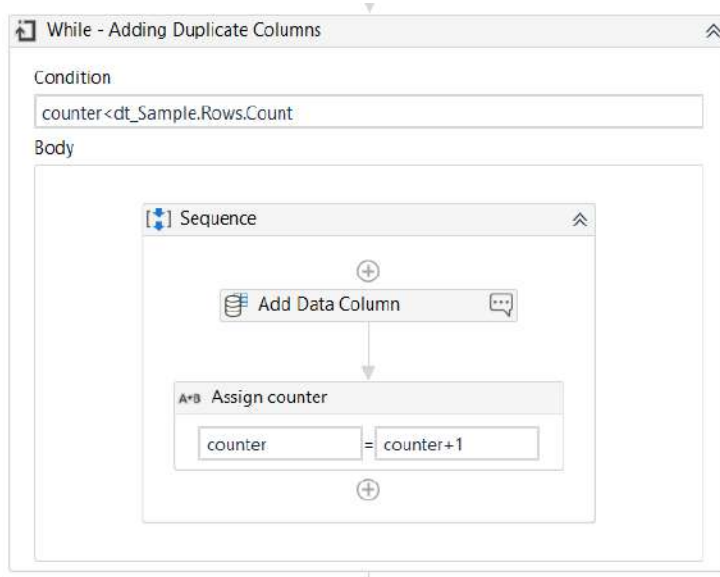
Sample File:

	A	B
1	Name	Job
2	Leon	Engineer
3	Jack	Lawyer
4	Sam	Doctor
5	Jennifer	Accountant

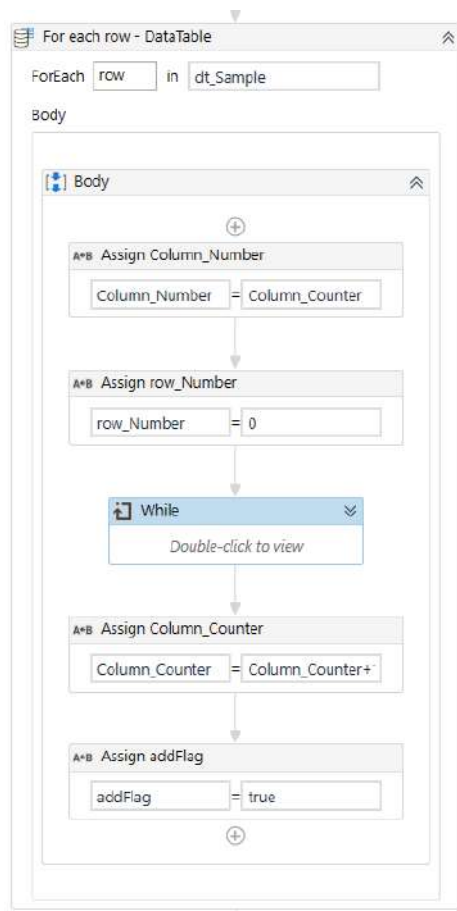
Step 1: Drag a Workbook “Read Range” activity into the design panel to read the input excel



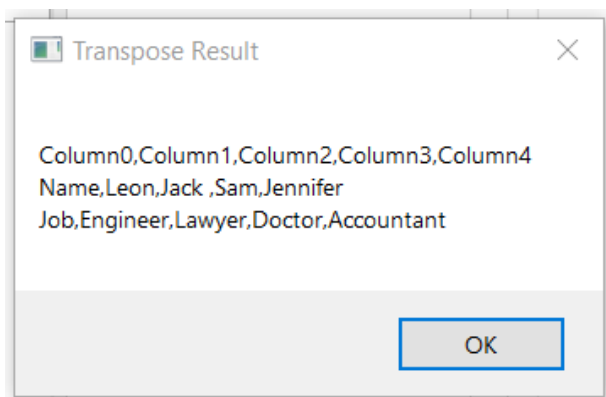
Step 2: Drag a “While”, “Add Data Column” and “Assign” activities into the design panel to create dummy columns (*Because here we are transposing the headers too*)



Step 3: Drag a “For Each Row”, “While”, “Add Data Row” and “Assign” activities into the design panel and write the logic to convert rows to columns.



Step 4: Finally, execute the workflow to see the results.



Click the link below to download the example project:

<https://drive.google.com/drive/folders/1ZmAYAl32LZlnFnz08JA-ZFlsGKza3ira?usp=sharing>

2.13 How to Compare Two Excel Files of the Same Format and Get Matched Data

While dealing with data table operations, sometimes we might want to compare two data tables and get matching data based on a single column. Use this expression:

```
(From a In SourceDT.AsEnumerable Join b In Destination_Table.AsEnumerable On  
a("Column_Name").ToString Equals b("Column_Name").ToString  
Select a).CopyToDataTable.Select().CopyToDataTable().DefaultView.ToTable(False,"  
Column_Name_To_Output")
```

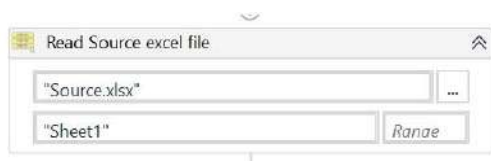
Note:

Replace SourceDT and DestinationDT with your required data tables and Column_Name with your required column name that you want to find a match with and Column_Name_To_Output for the column you want to output which is the matched data.

Implementation with UiPath

Let us implement a workflow which takes two excel files as input and gives us matched data between those excel files as output

Step 1: Drag a “Read Range” (here I have used workbook activity) activity and read excel file 1 (Source.xlsx) and set SourceDT as the output variable.

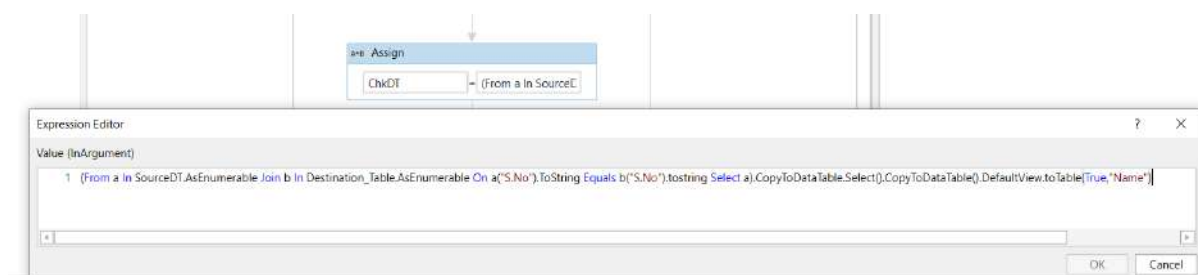


Drag a “Read Range” (here I have used workbook activity) activity and read excel file 2 (Destination.xlsx) and set Destination_Table as the output variable.



Step 2: Drag an “Assign” activity into the design panel and enter the above-mentioned code.

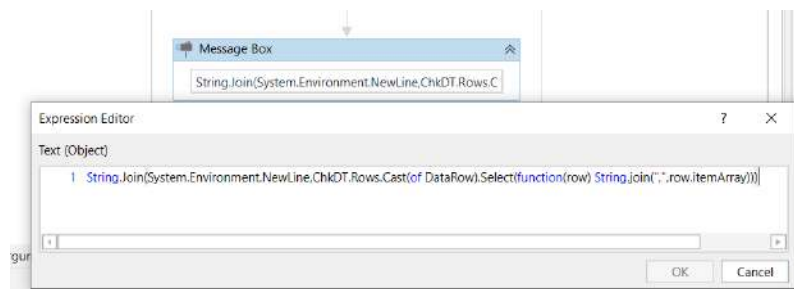
Note that we are looking for a match of the serial number “S.No” then we output the person’s “Name” in the Column_Name_To_Output property.



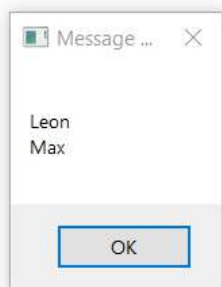
Step 3: Drag a “Message Box” activity to display the result.

Note:

Here I have used LINQ to display the DataTable instead of output Datatable



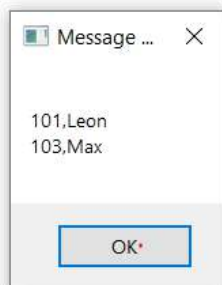
Step 4: Execute the workflow.



If you want to see the entire row as output, then don't enter a value for the Column_Name_To_Output like so:

(From a In SourceDT.AsEnumerable Join b In Destination_Table.AsEnumerable On a("S.No").ToString Equals b("S.No").ToString Select a).CopyToDataTable.Select().CopyToDataTable().DefaultView.ToTable(True)

Then the output will look like this:



Click the link below to download the example project:

<https://drive.google.com/drive/folders/1DXCDIlwAboByIG-UUOPw4CfojkNYemiu?usp=sharing>

2.14 How to Find Common/Uncommon Rows Between Two Datasets

Datasets in programming are commonly used to store the collection of data in rows and columns. Sometimes we may need to find the common or uncommon records between the two *DataTables*.

Common Values:

```
dt_CommonRows =  
dt1.AsEnumerable().Intersect(dt2.AsEnumerable(),System.Data.DataRowComparer.Default).CopyToDataTable
```

UnCommon Values:

```
dt_UnCommonRows =  
dt1.AsEnumerable().Except(dt2.AsEnumerable(),System.Data.DataRowComparer.Default).CopyToDataTable
```

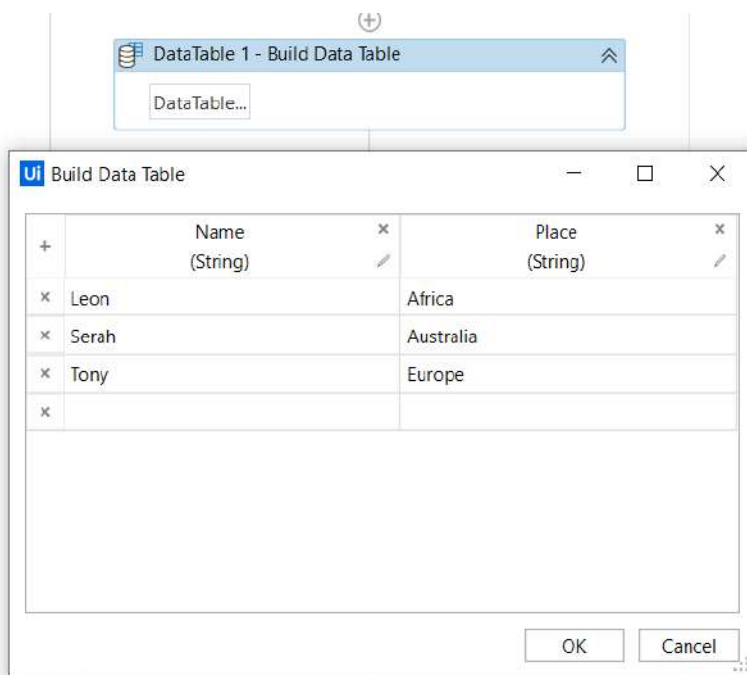
Note:

Replace dt1 and dt2 with your appropriate data tables.

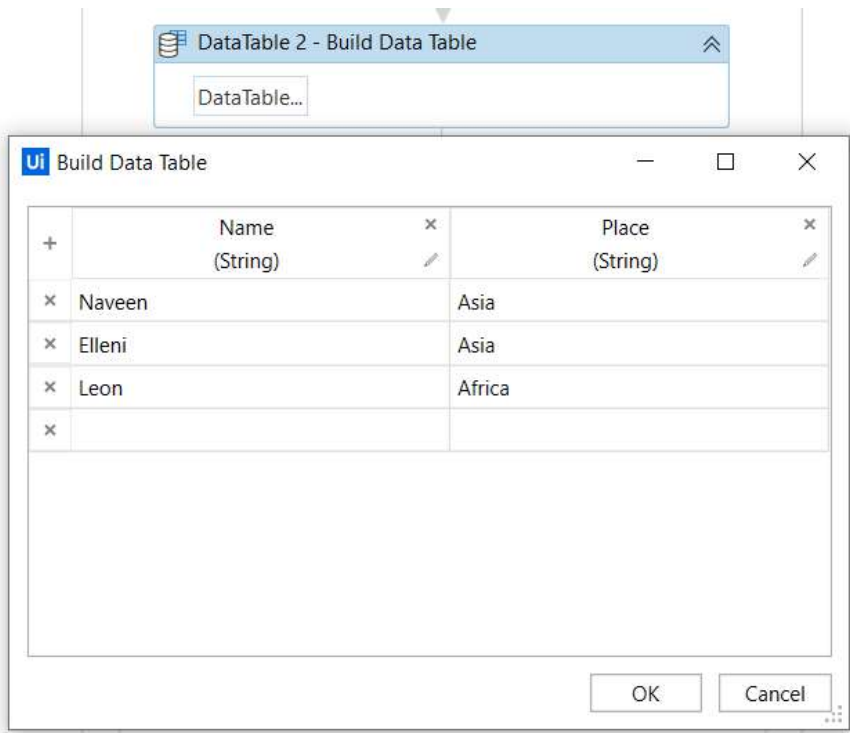
Implementation with UiPath

Let us implement a workflow that takes a two-sample DataTables and displays the common and uncommon rows between them.

Step 1: Drag “*Build DataTable*” activity into the design panel to build Sample DataTable 1 and initialize it with some sample data as shown below.



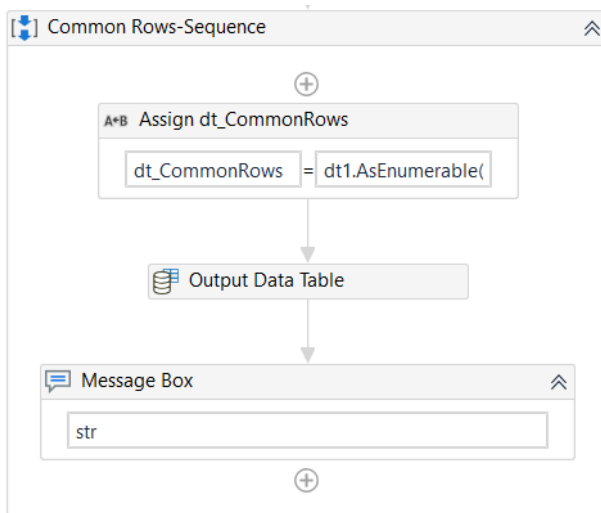
Step 2: Drag “Build DataTable” activity into the design panel to build Sample DataTable 2 and initialize it with some sample data as shown below.



Step 3: Drag an “Assign” activity into the design panel and use the below-mentioned code to find the COMMON rows between two DataTables.

`dt1.AsEnumerable().Intersect(dt2.AsEnumerable(),System.Data.DataRowComparer.Default).CopyToDataTable`

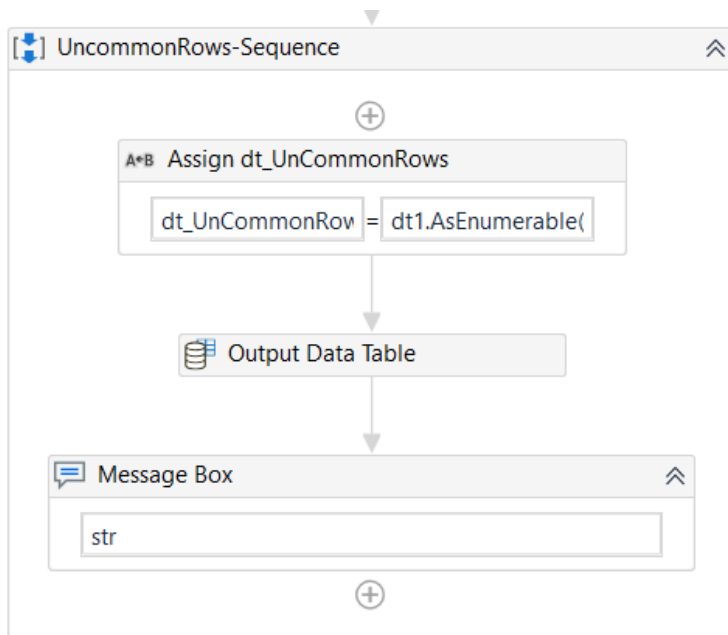
And use “Output DataTable” and the “Message box” to display the resulting DataTable.



Step 4: Drag another “Assign” activity into the design panel and use the below-mentioned code to find the UNCOMMON rows between two DataTables.

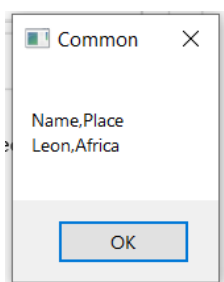
`dt1.AsEnumerable().Except(dt2.AsEnumerable(),System.Data.DataRowComparer.Default).CopyToDataTable`

And use “Output DataTable” and the “Message box” activities to display the resulting DataTable.

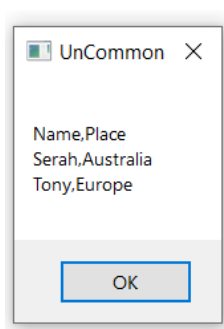


Step 4: Finally, execute the workflow to see the results.

Common Rows:



Uncommon Rows:



Click the link below to download the example project:

https://drive.google.com/drive/folders/1imgyq_0upuBbqbrAAOzpqmTg3er3kaSp?usp=sharing

2.15 How to Include a HTML Table in Mail Body with Dynamic Values

Sometimes we might need to send emails which contain tabular data. This HTML template can help you in inserting tabular structure inside the mail body.

Implementation with UiPath

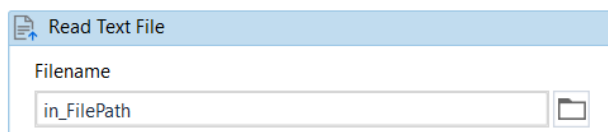
In the Html Template file provided, Add the required number of “tags” for the required number of rows inside the table.

```
MailBodyHtmlContent - Notepad
File Edit Format View Help
<html>
<head>
<style>
table, th, td {
border: 1px solid black;
}
</style>
</head>
<body>

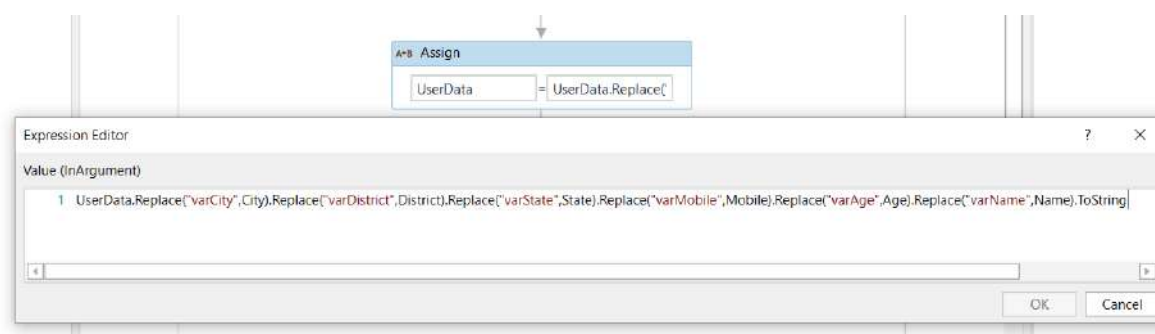
<table>
<tr>
<td>Name </td>
<td>varName </td>
</tr>
<tr>
<td>Age </td>
<td>varAge </td>
</tr>
<tr>
<td>Mobile </td>
<td>varMobile </td>
</tr>
<tr>
<td>State </td>
<td>varState </td>
</tr>
<tr>
<td>District </td>
<td>varDistrict </td>
</tr>
<tr>
<td>City </td>
<td>varCity </td>
</tr>

</table>
```


Step 1: Drag a “Read Text File” activity into the design panel and provide the html file path to it. Read the Html Content into a String Variable called UserData.



Step 2: Drag an “Assign” activity into the design panel and use the Replace method to replace the “varXXX” values with the respective variables to pass the data dynamically.



Step 3: Drag a “Send Outlook Mail Message” activity into the design panel and try to provide the required property fields i.e. SUBJECT, TOMAIL etc. Along with the “UserData” Variable that we have with us from the above steps performed and check the property “IsBodyHtml”.

 **Send Outlook Mail Message**

To

in_ToMailAddress



Subject

in_Subject


Body

"Hi team,

"+"Please find the below
[Attach Files](#)

Properties  

UiPath.Mail.Outlook.Activities.SendOutlookMail

 **Attachments**

Attachments


(Collection)

...

AttachmentsCollection

Allows specifying a

...

 **Common**

DisplayName


Send Outlook Mail Mess

...

TimeoutMS

Specifies the amou

...

 **Email**

Body


"Hi team,

...

Subject

in_Subject


...

 **Forward**

MailMessage

The message to be

...

 **Input**

Account


The account used to

...

SentOnBehalfOfName

Indicates the displa


...

 **Misc**

Private

☐

...

 **Options**

IsBodyHtml


☒

...

IsDraft

☐

...

 **Receiver**

Bcc

The hidden recipier

...

Cc

The secondary reci

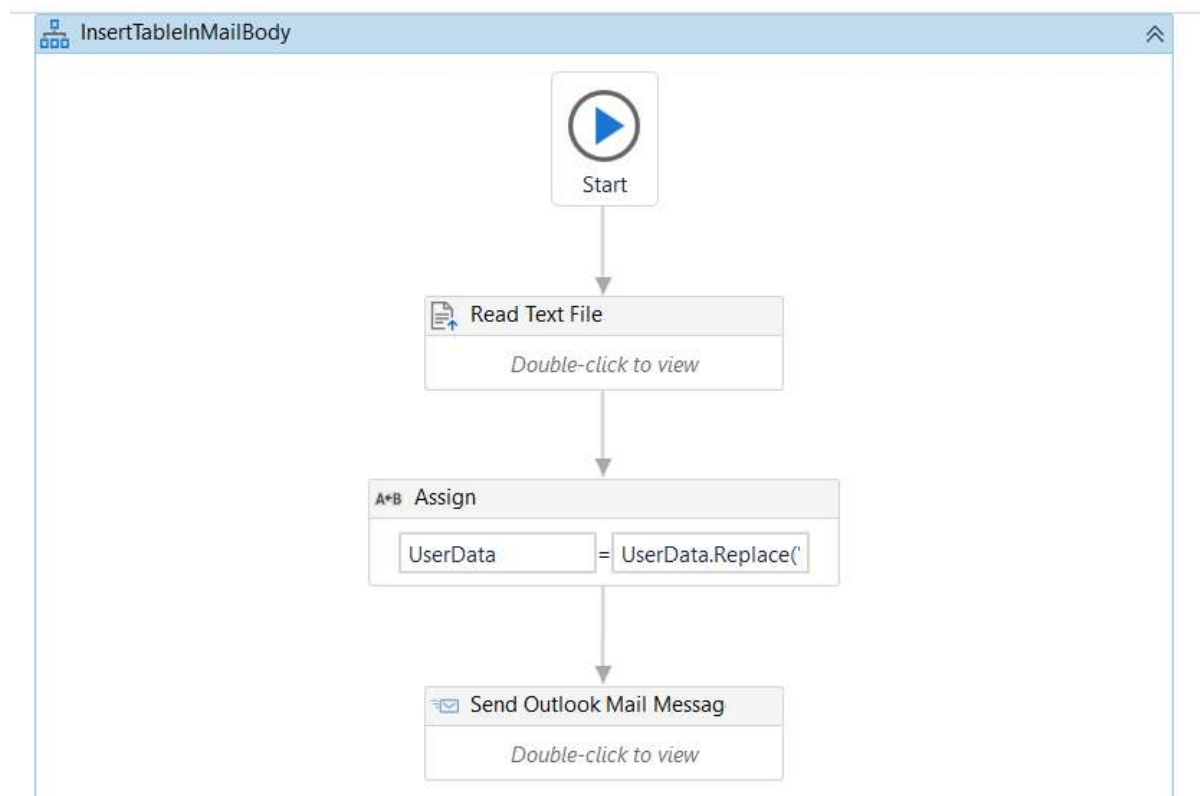
...

To

in_ToMailAddress

...

This is what the final flow looks like:



Step 4: Run the bot and see the output (sent mail):

Hi team,

Please find the below the user details.

Name	Leon Petrou
Age	26
Mobile	06544154545
State	Eastern Cape
District	East
City	Cape Town

Regards,
Leon's Bot

Click the link below to download the example project:

https://drive.google.com/drive/folders/11SCFzAuCOZGhmPYYUztrj7tttFNZ3jQ_?usp=sharing

3 File / Folder Operations

3.1 How to Get the Latest File from a Folder

When we download a file from the browser or some other application, we might need to use the most recently downloaded file to proceed further. So, how to get the latest file from a folder?

```
String.Join("",  
Directory.GetFiles(our_FolderPath, "*", SearchOption.AllDirectories).OrderByDescending(Function(d) New  
FileInfo(d).CreationTime).Take(1) )
```

Note:

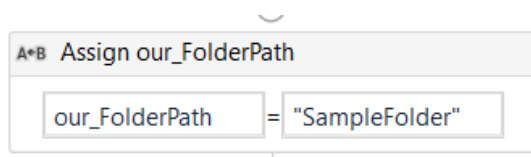
1. Replace our_FolderPath with your desired folder path
2. Replace "*" with your required file format "*.xlsx" or "*.pdf" etc. If set to "*" it will read all file types.

Implementation with UiPath

Let us implement a project to get the latest file from a folder.

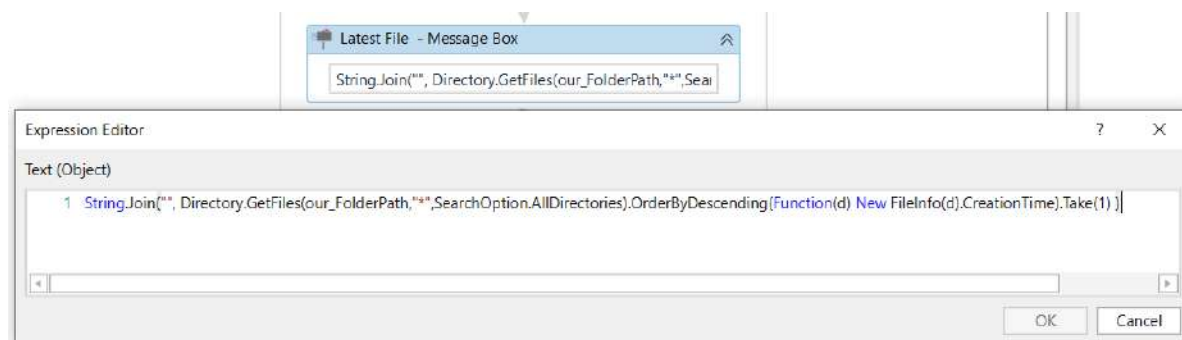
Step 1: Drag an "Assign" activity into the design panel and supply the "Folder Path" to it. Here we use the relative path instead of the full path because the SampleFolder is in the same folder as our Main.xaml file.

```
our_FolderPath = "SampleFolder"
```

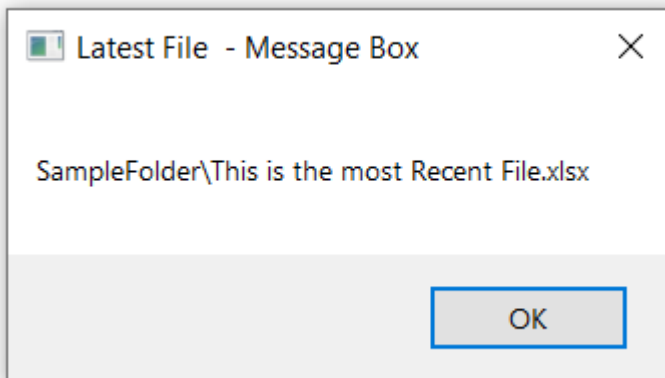


Step 2: Drag a "Message Box" activity into the design panel and supply the below-mentioned code to it.

```
String.Join("",  
Directory.GetFiles(our_FolderPath, "*", SearchOption.AllDirectories).OrderByDescending(Function(d) New  
FileInfo(d).CreationTime).Take(1) )
```



Step 3: Finally, run the workflow to display the path of the latest file



Click the link below to download the example project:

https://drive.google.com/drive/folders/1O44zFQNLP5q1O06Q_pXJu_rgiVKwaZMt?usp=sharing

3.2 How to Delete Multiple Files in a Folder

While automating business processes, we deal with many files and might perform few activities on them like deleting, moving or renaming. Let us see how to delete multiple files from a folder.

```
Array.ForEach(Directory.GetFiles(our_DirectoryPath),Sub(x) File.Delete(x))
```

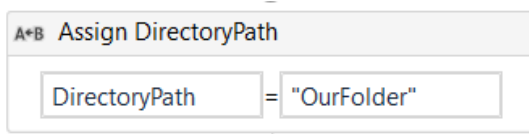
Note:

Replace our_DirectoryPath with your desired folder path. This can be a full or relative path.

Implementation with UiPath

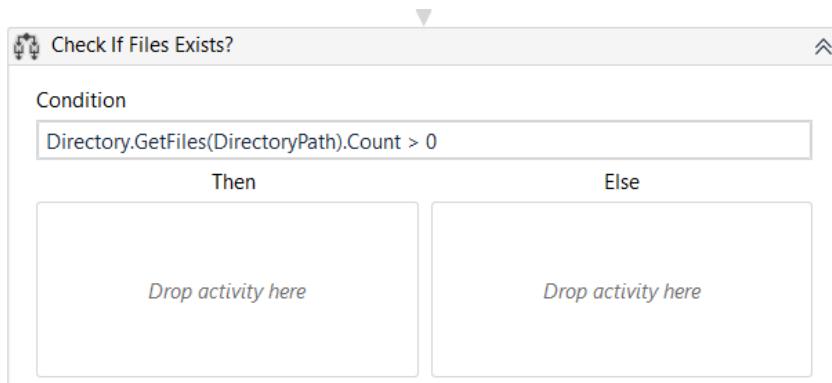
Let us implement a project to delete multiple files from a given folder.

Step 1: Drag an “Assign” activity into the design panel and supply the “Folder Path” to it.



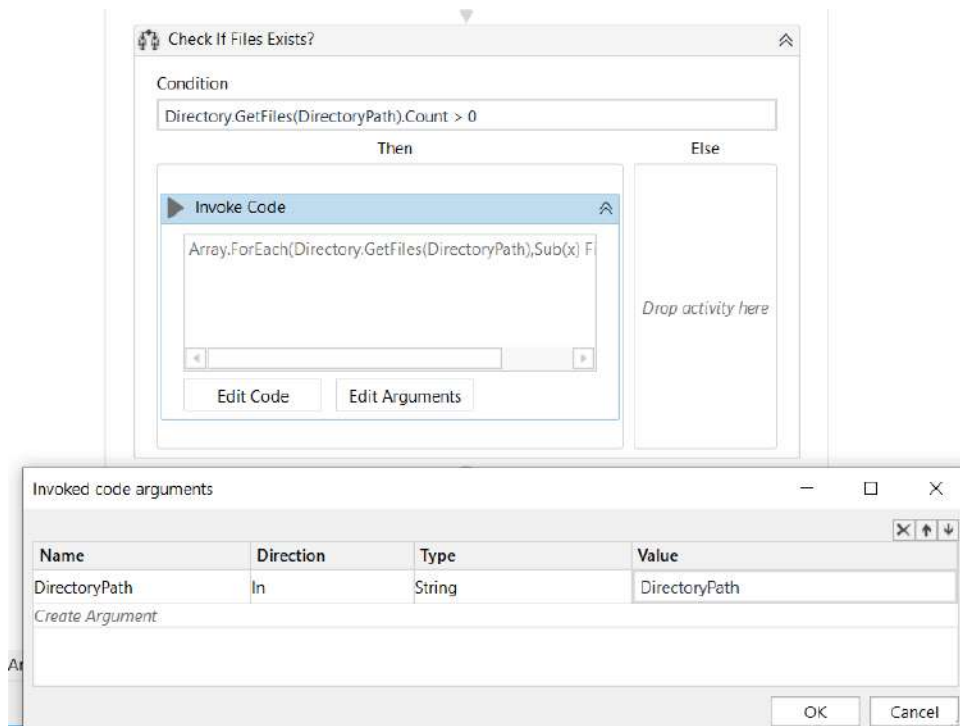
Step 2: Drag “If” activity to check if the folder contains any files.

```
Directory.GetFiles(DirectoryPath).Count > 0
```

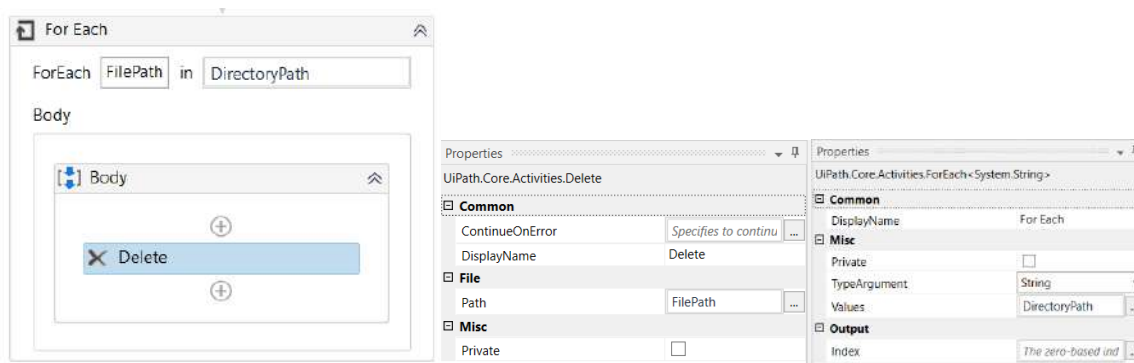


Step 3: Drag an “Invoke Code” activity, add the below-mentioned code to it to and pass the folder’s Path from which we have to delete the files.

`Array.ForEach(Directory.GetFiles(DirectoryPath),Sub(x) File.Delete(x))`



Note this above expression in the Invoke Code activity is the same as doing this:



Step 4: Now, execute the workflow to find the result. The OurFolder folder will be empty.

Before:

Name	Date modified	Type	Size
New Bitmap Image.bmp	2019/06/21 11:05	BMP File	0 KB
New Compressed (zipped) Folder.zip	2019/06/21 11:05	Compressed (zipp...	1 KB
New Microsoft Excel Worksheet (2).xlsx	2019/06/21 11:05	Microsoft Excel W...	7 KB
New Microsoft Excel Worksheet.xlsx	2019/06/21 11:04	Microsoft Excel W...	7 KB
New Text Document.txt	2019/06/21 11:05	Text Document	0 KB

After:

Name



Date modified

Type

Size

This folder is empty.

Click the link below to download the example project:

<https://drive.google.com/drive/folders/1Bov66E67ZQs-7Y8h1dOrGtjUhyWbchor?usp=sharing>



3.3 How to Get Specific Types of Files from a Folder

File management is inevitable in business processes automations. We might want to copy, move, delete or process files based on few conditions. Let's see how to get specific type of files from a folder

```
myList = Directory.GetFiles(folderPath, "*.xlsx", System.IO.SearchOption.AllDirectories)
```

Note:

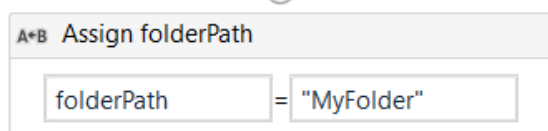
1. folderPath is the folder path where you have your files
2. myList is the variable containing your required files (this is a string list of file paths)
3. Replace "*.xlsx" with your desired extension. Another example "*.pdf". You can Even set it to contain a part of a file name. For example, "Invoice-2020-09-*.pdf". This will get all pdf files for all days in September 2020.

Implementation with UiPath

Let us implement a workflow which displays the names of prespecified files (i.e. .xlsx extension files) from a folder.

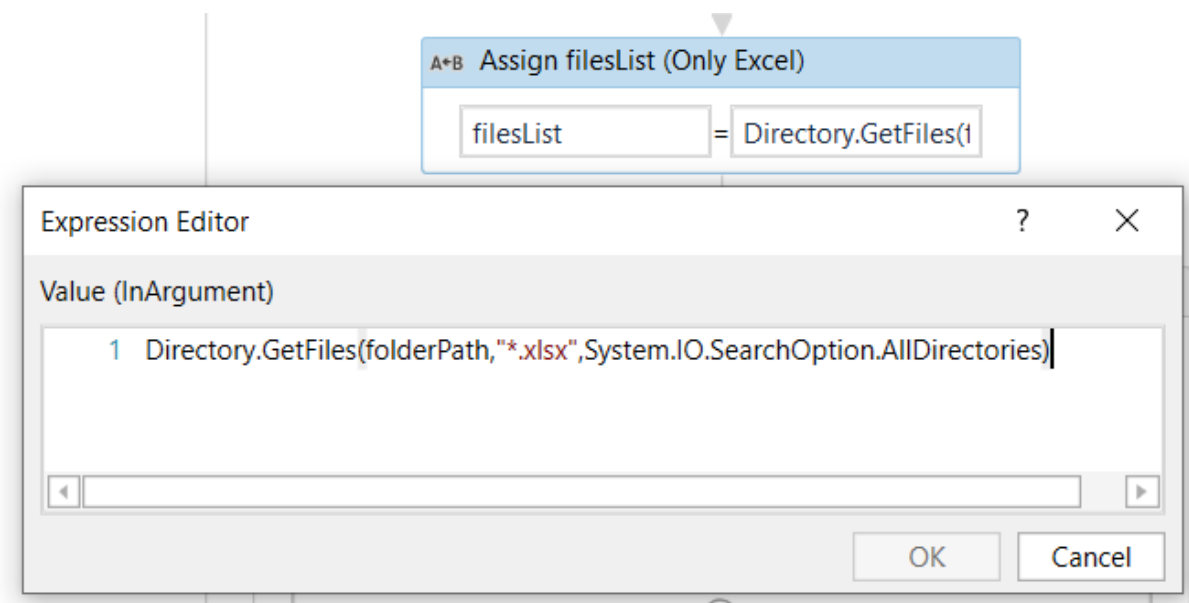
Step 1: Drag "Assign" activity into the design panel and populate it with your folder path, here we use the relative file path to MyFolder:

"MyFolder"

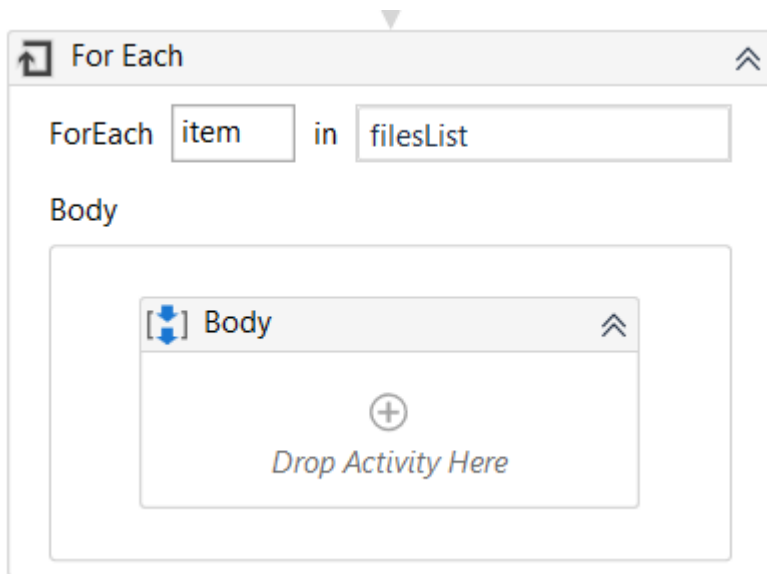


Step 2: Drag an "Assign" activity into the design panel and populate it with above-mentioned code

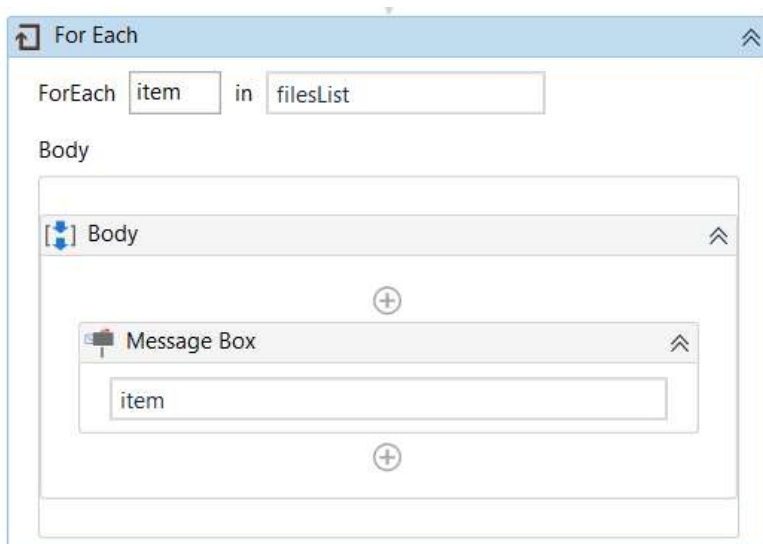
"Directory.GetFiles(folderPath, "*.xlsx", System.IO.SearchOption.AllDirectories)"



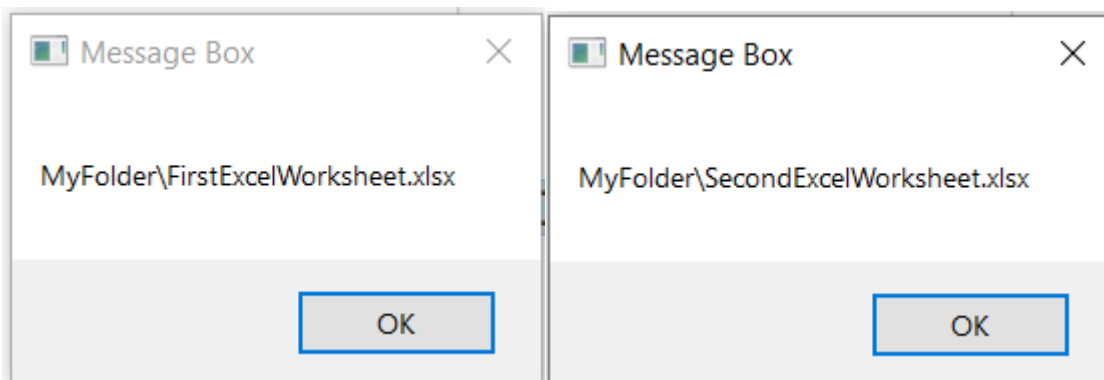
Step 3: Drag a "For Each" activity into the design panel and pass the above "filesList" variable to it. Set the type argument in the properties panel to a String.



Step 4: Drag “Message Box” activity into the For Each activity and pass the “item” value into it.



Step 5: When we run it, we can see the file names of .xlsx extension being displayed.



Click the link below to download the example project:

<https://drive.google.com/drive/folders/1OhaftBIK1Z-6Eigp256cachh1Jw4CpvU?usp=sharing>

4 Text and List Manipulations

4.1 How to Read / Write to a Text File

Let's see how to read and write to Text file. This is useful for storing and editing text.

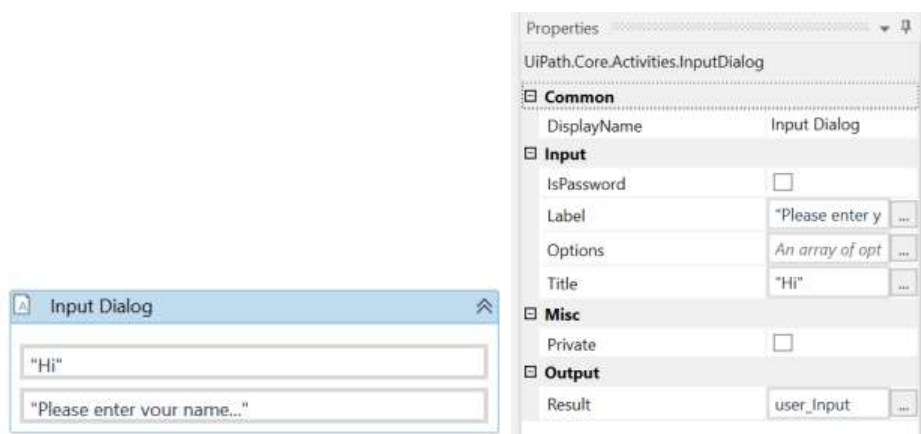
Implementation with UiPath

Let us implement a workflow to write the data to a text file, then read data from it and then again append some text to it.

Different Text file Operations:

1. Writing data to a text file
2. Reading data from a text file
3. Appending data to an existing text file

Step 1: Drag “*Input Dialog*” activity into the design panel and create a variable to store the user-provided value.



Step 2: Drag “*Write Text File*” activity to write the data provided by the user to a text file.
(If the supplied .txt file is not available in the directory, the activity will create a new file with that name)



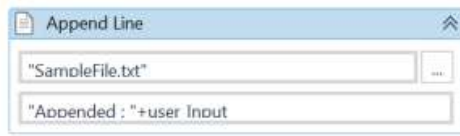
Step 3: Drag “*Read Text File*” activity to read the data from the provided text file.



Step 4: Drag “*Message Box*” activity to display the text obtained from the above step.



Step 4: Drag “*Append Line*” activity into the design panel to append the text to an existing text file and pass the inputs i.e. data that is to be appended and text file’s path. When we use “Append Line” activity, it will not overwrite the text file data but instead, it will write the new data from the next new line unlike “Write Text File”.



Step 5: Finally, execute the workflow and observe the output in the SampleFile.txt

Click the link below to download the example project:

https://drive.google.com/drive/folders/1jbl6Ztcwx_JEcdJZoRb1V-zCmWS7M9aQ?usp=sharing

4.2 How to Define a Dictionary, Add Items To a Dictionary and Access Items From a Dictionary

What is a Dictionary? In general, a Dictionary is a collection of key and value pairs, where “Key” is equivalent to Words and “Value” is equivalent to Meanings/Definitions.

The Dictionary<TKey, TValue> is a generic collection class available in the System.Collection.Generics namespace. Here “TKey” denotes the type of key and “TValue” is the type of TValue. For example, a Country Capital dictionary in VB.NET would look like this:

```
{{“South Africa”, “Pretoria”},{“USA”, “Washington DC”}}
```

Implementation with UiPath

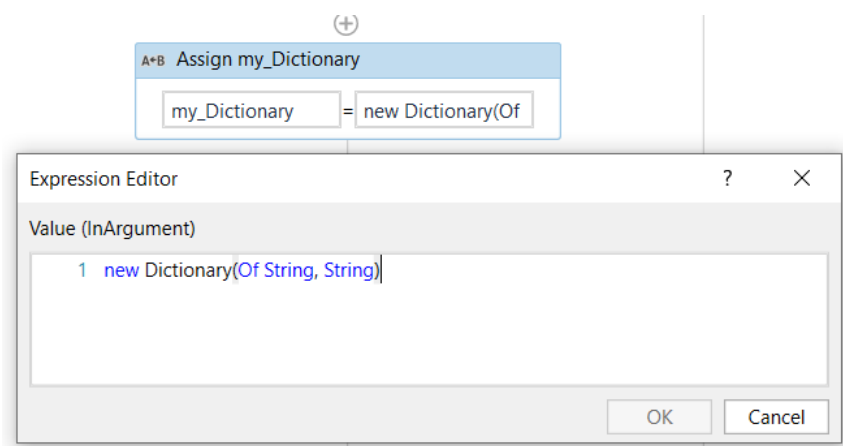
Major Steps That We Will Implement:

1. Defining a Dictionary
2. Adding items to the Dictionary
3. Accessing items from the Dictionary

Step 1: Drag “Assign” activity into the design panel and define the dictionary.

```
my_Dictionary = new Dictionary(Of String, String)
```

Here my_Dictionary is of System.Collection.Generics.Dictionary<String, String> type

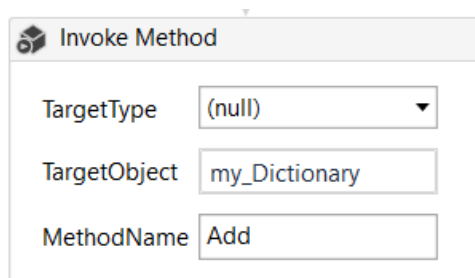


Step 2: Drag “Invoke Method” activity into the design panel to add items to the Dictionary.

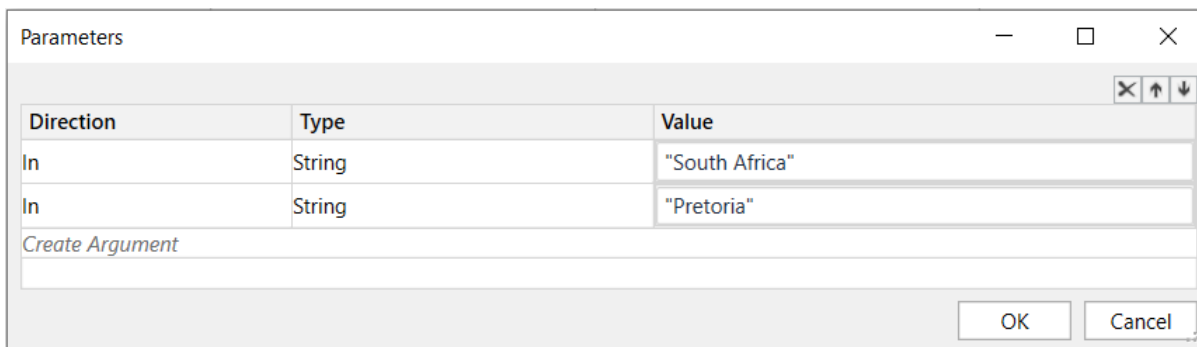
Example:

Key: South Africa

Value: Pretoria



Click “Parameters” in the properties panel of the Invoke Method activity.

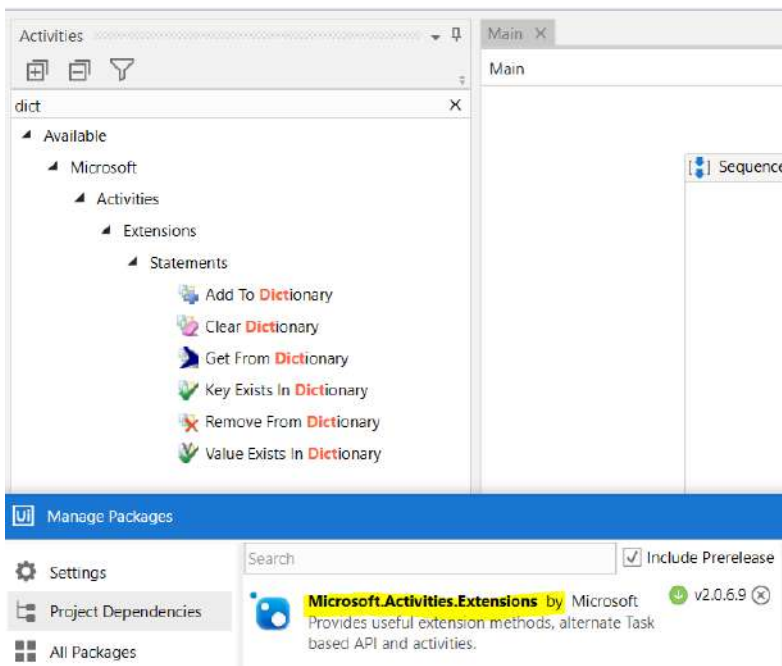


Direction	Type	Value
In	String	"South Africa"
In	String	"Pretoria"

Create Argument

OK Cancel

Step 3: Another way of adding Dictionary items is by using Microsoft.Activities.Extensions package.

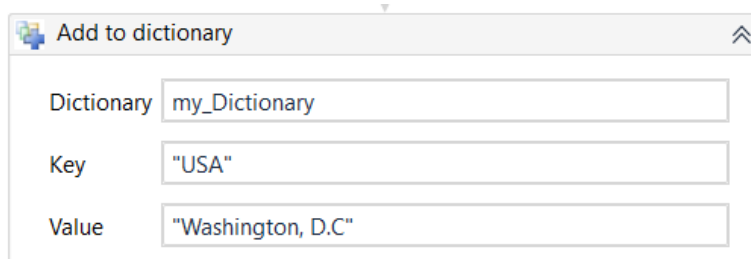


Drag “Add to dictionary” activity into the design panel to add items to the Dictionary.

Example:

Key: USA

Value: Washington, D.C



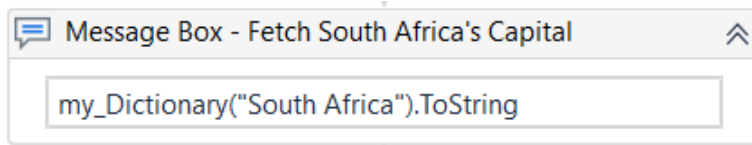
Add to dictionary

Dictionary: my_Dictionary

Key: "USA"

Value: "Washington, D.C"

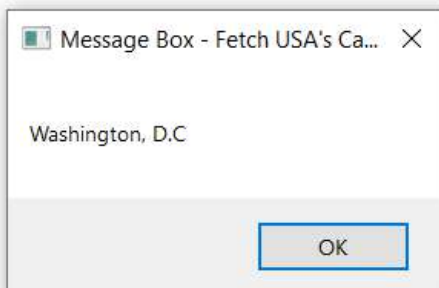
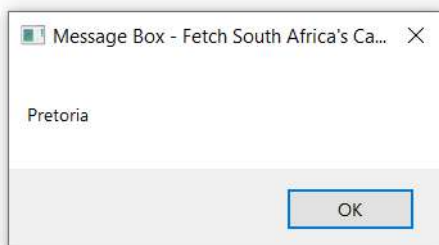
Step 4: Drag “Message Box” activity to display the first added Key and Value pair
i.e. Find the Value of the Key “South Africa”



Step 5: Drag “Message Box” activity to display the second added Key and Value pair.
i.e. Find the Value of the Key “USA”



Step 6: Finally, execute the workflow.



Click the link below to download the example project:

https://drive.google.com/drive/folders/1jbl6Ztcwx_JEcdJZoRb1V-zCmWS7M9aQ?usp=sharing

4.3 How to Use Regular Expressions (Regex)

A regular expression is a sequence of characters that define a search pattern. Usually such patterns are used by string searching algorithms for "find" or "find and replace" operations on strings, or for input validation for online forms. We write a pattern that will match our requirement in a body of text. For example we can write a Regex expression for an email address, phone number, social security number, zip code or any other piece string which follows a pattern.

Implementation with UiPath

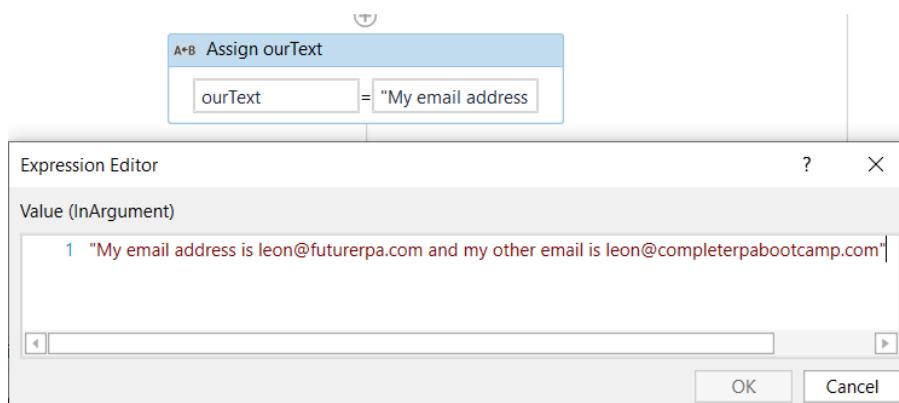
Let us implement a workflow to extract a "email address" from a given text using *Regular Expressions*.

Major steps that we will implement:

1. Check if the given text contains any email id?
2. Find all the matches
3. Extract the URLs from the email id.

Step 1: Drag an "Assign" activity into the design panel and supply sample data to it.

i.e. "My email address is **leon@futererpa.com** and my other email is **leon@completerpabootcamp.com**"



Step 2: Drag "Is Match" activity into the design panel to find if the sample data contains any email addresses in it and supply the required parameters. The output of this activity is a Boolean.

Note:

The regular expression for an email address is "[a-zA-Z0-9_\\-\\.]+)@([a-zA-Z0-9_\\-\\.]+)\\.([a-zA-Z]{2,5})"

I found this on Google but you can also generate Regex expressions yourself from scratch using the RegEx

RegEx Builder

Test Text

RegEx

Value

Quantifiers

Advanced

Value

Quantifiers

Advanced

Exactly

1

Full Expression

RegEx Option

Save

Cancel

Is Match

Properties ▼ 📌

UiPath.Core.Activities.IsMatch

Common

DisplayName	Is Match
-------------	----------

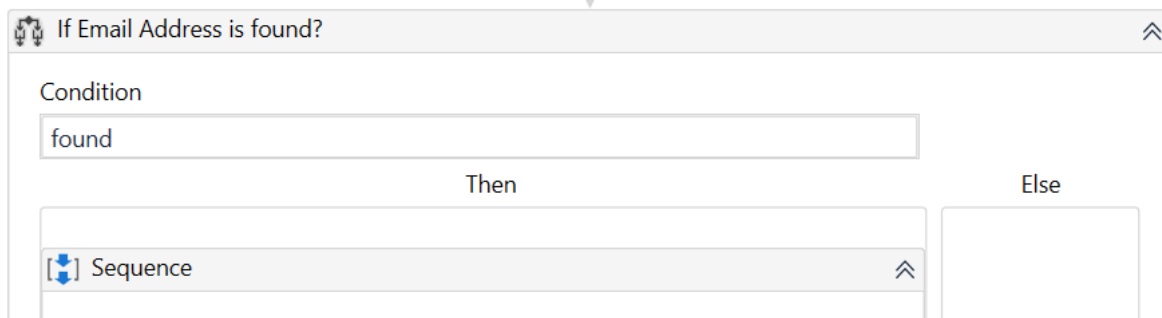
Input

Input	ourText ...
Pattern	"([a-zA-Z0-9_\\-\\.]+)@([a-zA-Z0-9_\\-\\.]+)\\.([a-zA-Z]{2,5})" ...
RegexOption	IgnoreCase ▼

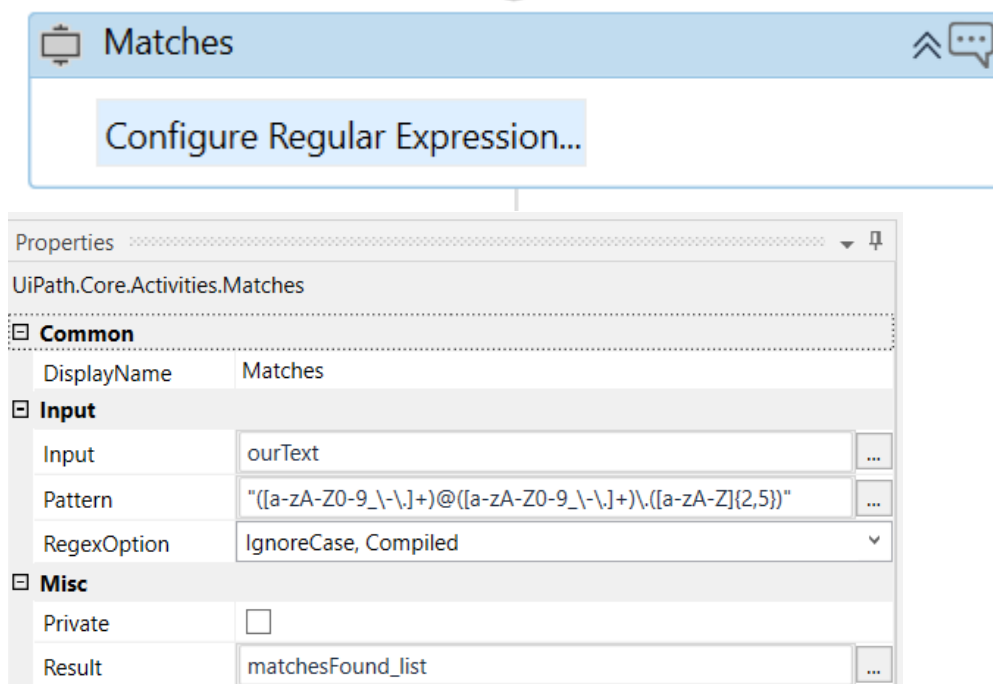
Misc

Private	<input type="checkbox"/>
Result	found ...

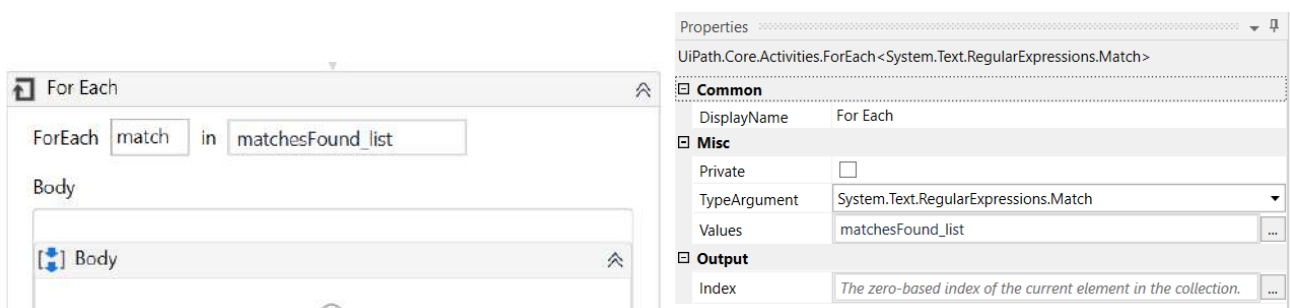
Step 3: Drag an “If” activity into the design panel to check if we found any matches through the above activity



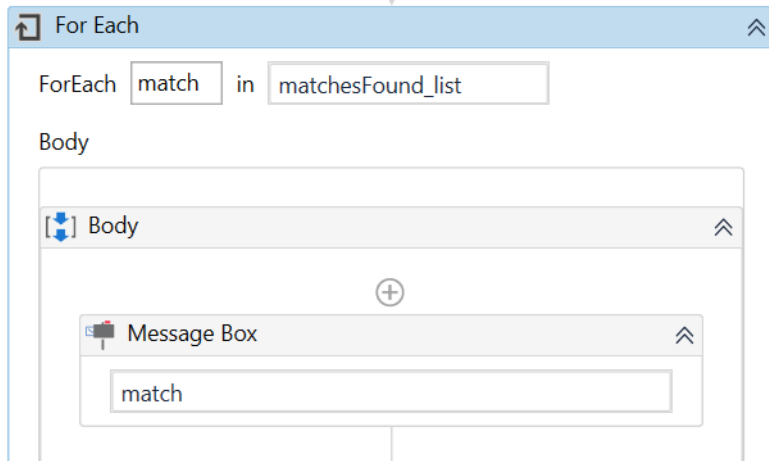
Step 4: Drag a “Matches” activity into the design panel to find all the email matches and supply the required parameters. The Result is a list of matches.



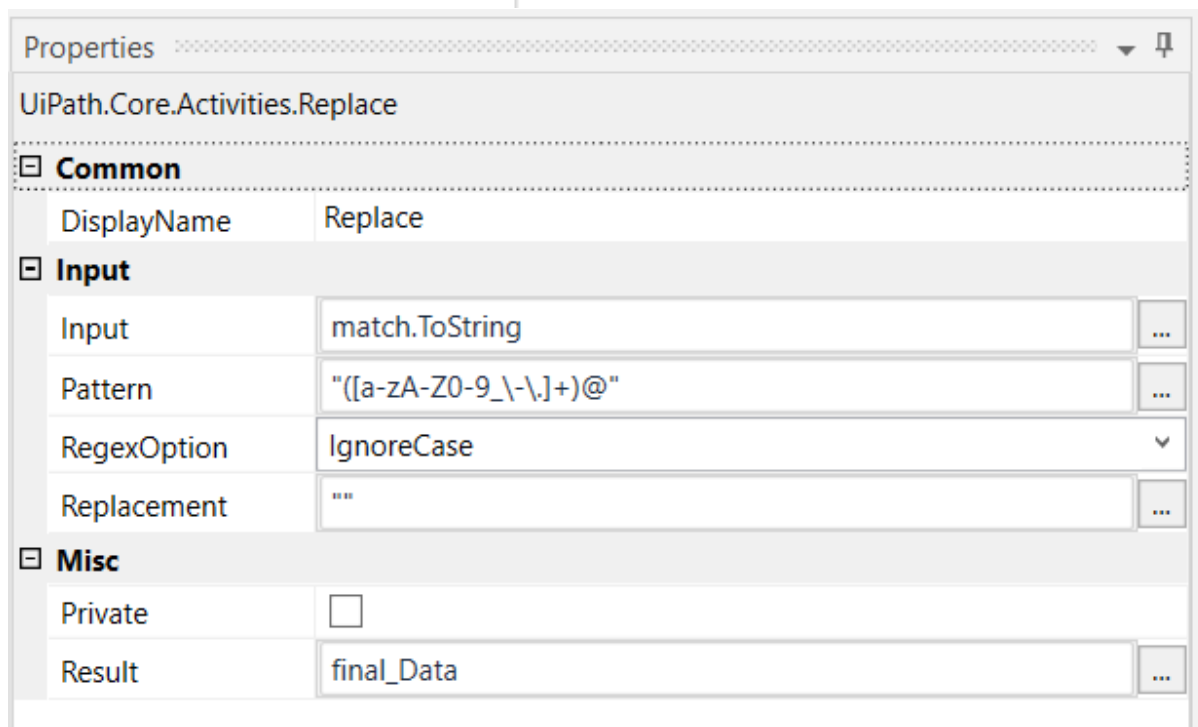
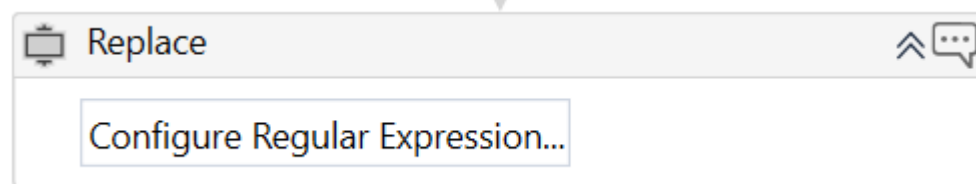
Step 5: Drag a “For Each” activity to loop through each match. Set the Type Argument to System.Text.RegularExpressions.Match



Step 6: Drag a “*Message Box*” activity into the body of the For Each activity to display each match in the list.



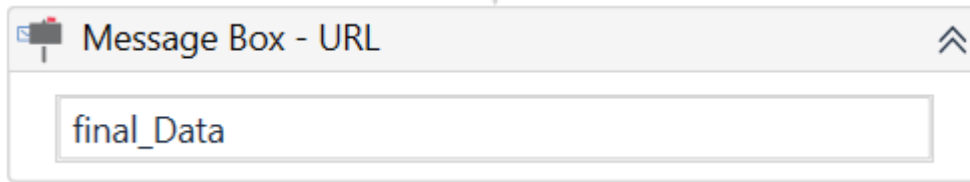
Step 7: Drag a “*Replace*” activity to extract the URL from the above-fetched email id and pass the mandatory fields



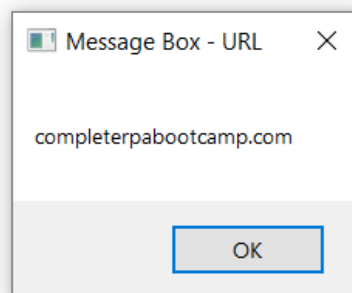
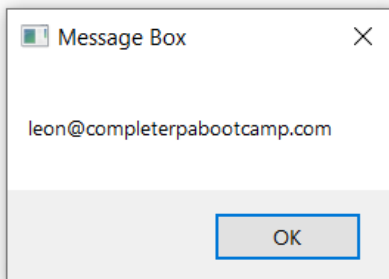
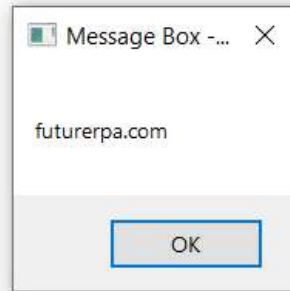
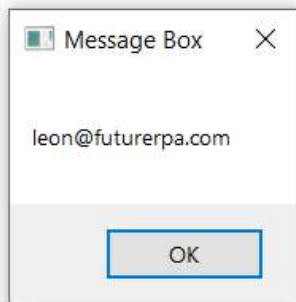
Note that we replace the username part of the email which follow the Regex "([a-zA-Z0-9_-\\.]+)@" with a blank string ""

This will leave just the URL part of the email address.

Step 7: Drag another “*Message Box*” activity to display the URL.



Step 8: Finally, execute the workflow and see the output of email and URL.



Confused about writing a Regular Expression?

Click [here](#) for a Regex playlist on YouTube (highly recommend) and click [here](#) for a great website to experiment with Regex expressions.

Click the link below to download the example project:

https://drive.google.com/drive/folders/1jbl6Ztcwx_JEcdJZoRb1V-zCmWS7M9aQ?usp=sharing

4.4 How to Initialize, Add and Removed Items from a List

Sometimes when we work with a List, we may need to initialize them. We use lists when the number of items in the list is undefined (as opposed to an array which has a fixed length). We are able to add and remove items from a list. Major steps that we will implement:

1. Initialize list
2. Add item to list
3. Remove item from list

Implementation with UiPath

Step 1: Initialize List

```
my_List = new List(of string)(new string[]{"value1", "value2"})
```

Note:

Create a List with your required DataType, i.e. string, Int, boolean....

First we initialise the list and add the three primary colours to the list

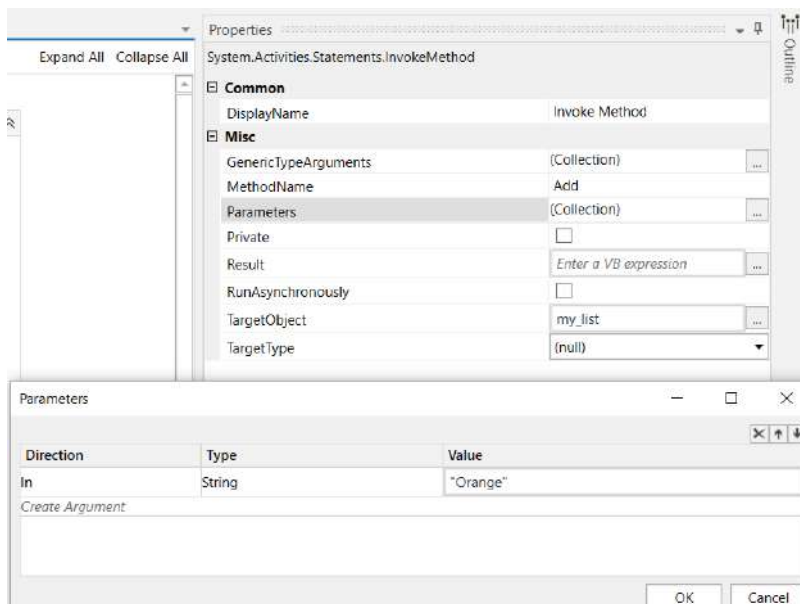
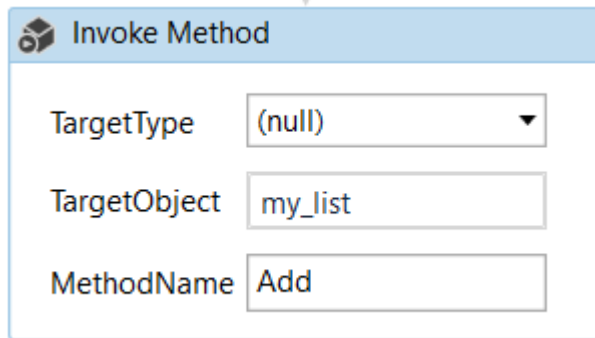


Set the Variable Type to an ICollection<string> which is basically a list of strings.

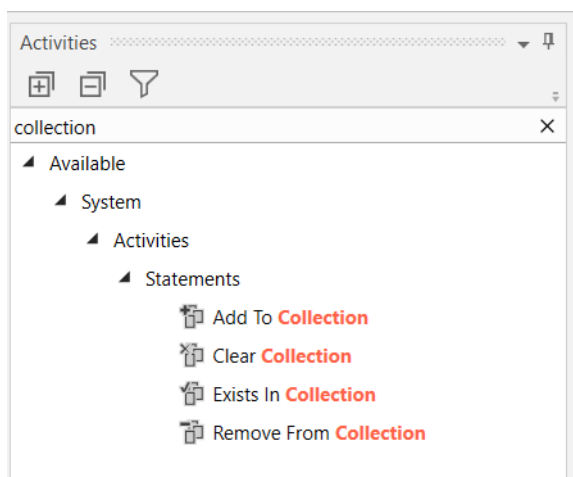
Name	Variable type	Scope	Default
my_list	ICollection<String>	Sequence	Enter a VB expression
Create Variable			
Variables Arguments Imports			

Step 2: Add item to list

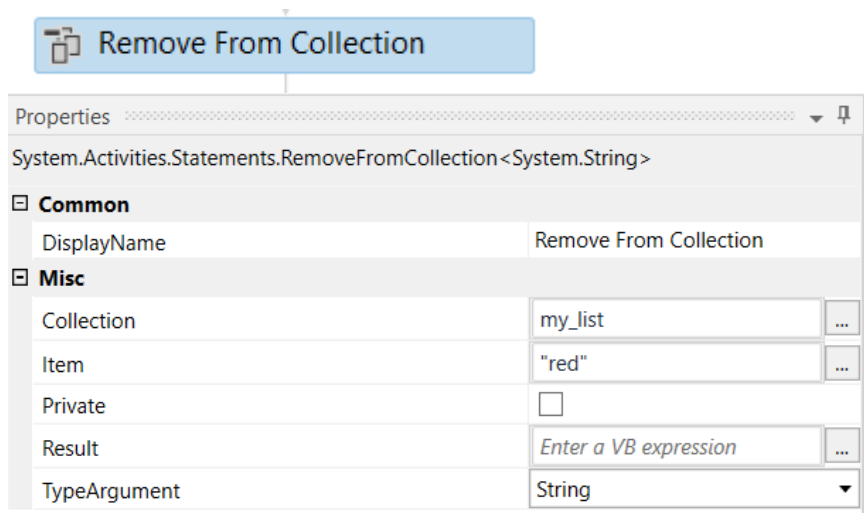
Next we add the colour “orange” to the list with an Invoke Method activity using the Add method



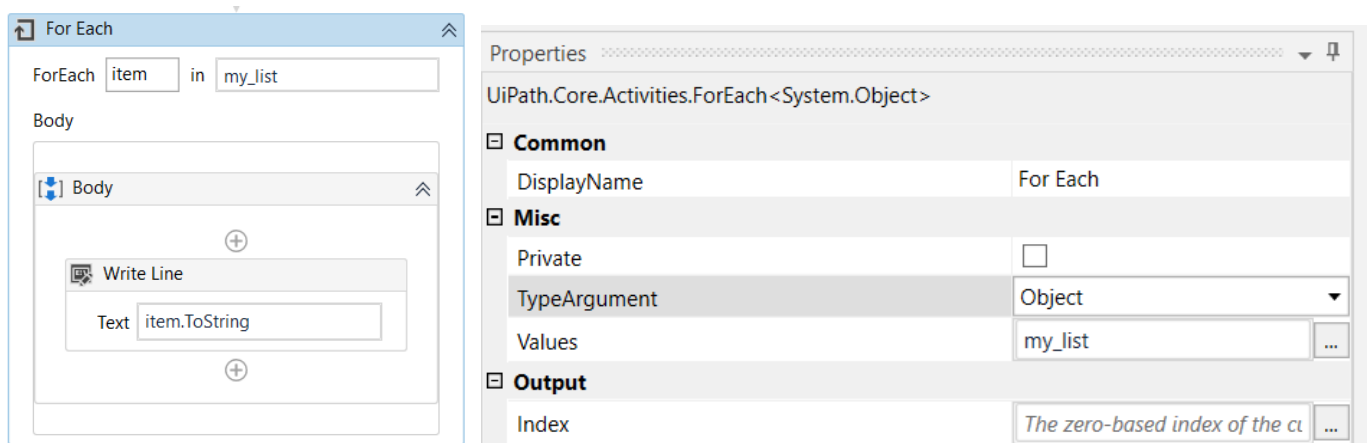
We could also add items using the “Add to Collection” activity from the Statements library.



Step 3: Remove item from list using “Remove From Collection” activity.



Write the final list to the output panel using a “For Each” set up like this:



Step 4: Finally, the output.

We started with blue, red and yellow. Then added orange and removed red. We are then left with blue, yellow and orange.

- blue
- yellow
- Orange

Click the link below to download the example project:

https://drive.google.com/drive/folders/1tPs_fu4Jsmf0fffUjq33MVZ5ZRzf2xXN?usp=sharing

4.5 How to Trim / Clean a String

There is always a possibility of extra spaces being added to the text by mistake, which needs to be handled before performing any operations on the data.

The TRIM() method of string can be used and it removes (or deletes) all leading and trailing white-space characters from the current String.

To remove the all leading and trailing white-space characters:

`our_String.Trim`

To remove the all leading white-space characters:

`our_String.TrimStart`

To remove the all trailing white-space characters:

`our_String.TrimEnd`

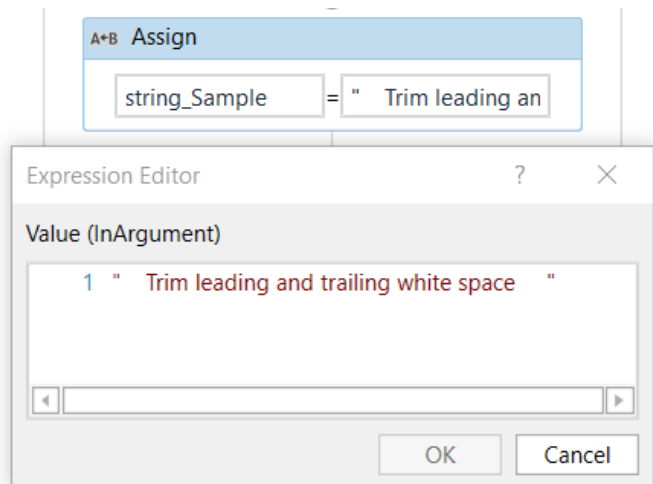
Note:

Replace our_String with your string variable

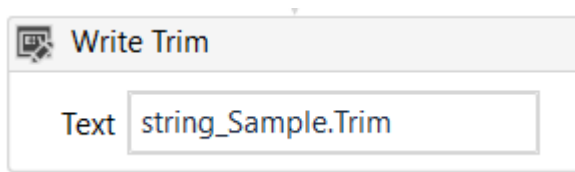
Implementation with UiPath

Let us implement a workflow which takes input from the user and displays the text by removing the extra spaces in it as per the method used.

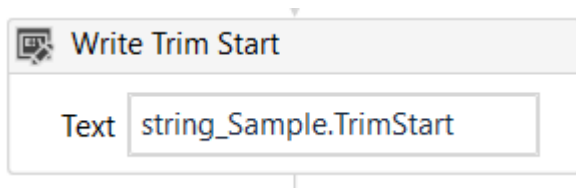
Step 1: Drag three “Assign” activities into the design panel and add the following sample string



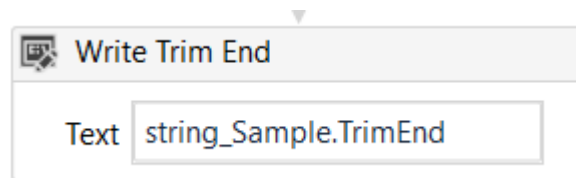
Step 2: Drag “Write Line” activity into the design panel and use TRIM method to remove all leading and trailing white-space characters if any.



Step 3: Use another “Write Line” activity and use TRIMSTART method to remove all leading white-space characters if any.



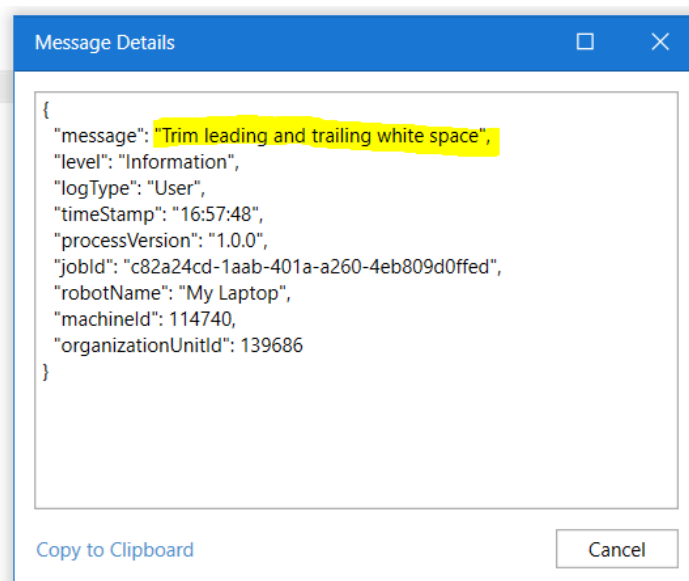
Step 4: Use one more “Write Line” activity and use TRIMEND method to remove all trailing white-space characters if any.



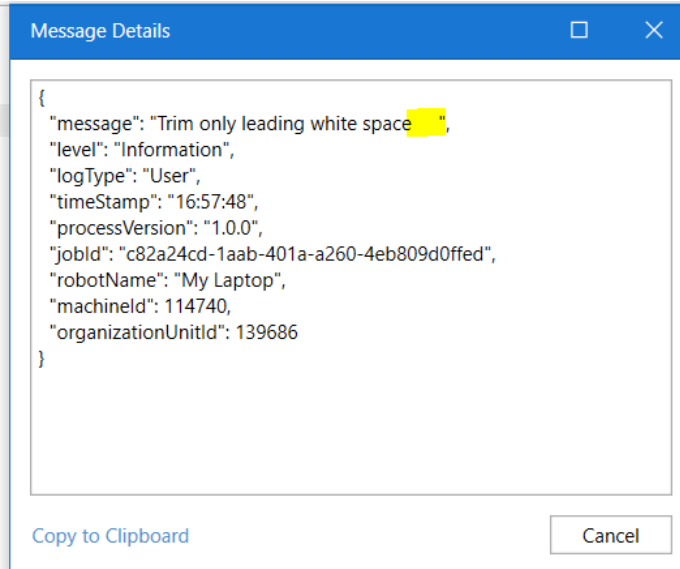
Step 5: Now, execute the workflow to display the appropriate results. In the output panel, it does not look like any trimming happened, but if you double click each of these three logs, you will see how the message output looks.

- ① Trim leading and trailing white space
- ① Trim only leading white space
- ① Trim only trailing white space

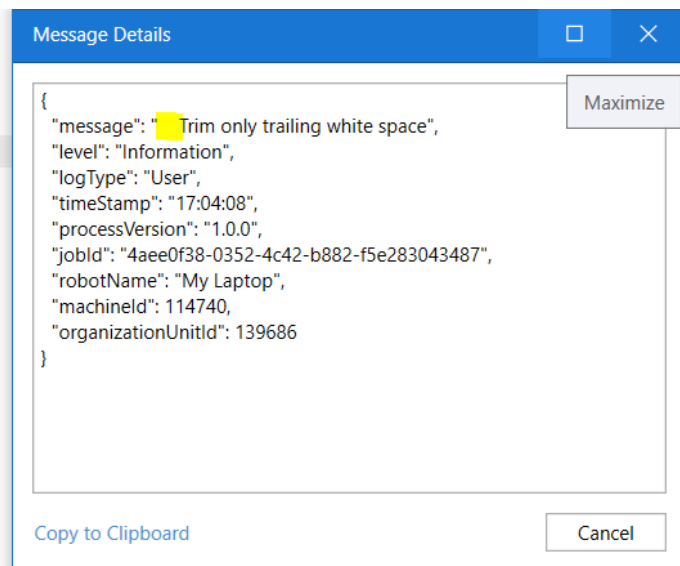
- ① Execution started for file: Main
- ① TrimAsString_Example execution started
- ① Trim leading and trailing white space
- ① Trim only leading white space
- ① Trim only trailing white space
- ① TrimAsString_Example execution ended in: 00:00:00



- Execution started for file: Main
- TrimAString_Example execution started
- Trim leading and trailing white space
- Trim only leading white space
- Trim only trailing white space
- TrimAString_Example execution ended in: 00:00:00



- Execution started for file: Main
- TrimAString_Example execution started
- Trim leading and trailing white space
- Trim only leading white space
- Trim only trailing white space
- TrimAString_Example execution ended in: 00:00:00



Click the link below to download the example project:

<https://drive.google.com/drive/folders/1RGv2hULPNpE0Da2MLh5pNSTjVA9HLJZ?usp=sharing>

4.6 How to Split a String

Sometimes we might want to break down a string datatype value into an array of substrings based on some separator/delimiter (a delimiter can be a space, a comma, a newLine e.t.c). You can not use a word as a delimiter, only a single character can be a delimiter. However, you can use the Replace method to convert a word to a character and then use that character as the delimiter. See the next section for how to use the Replace method.

Let's do an example where we split a comma separated string of numbers into an array using the comma as a delimiter. Assume we have a list of numbers assigned like so:

```
numbers_Data = "1,2,3,4,5"
```

Then we use an assign to split the string:

```
result_List = numbers_Data.Split(",")
```

Note:

1. Replace numbers_Data with any string variable
2. result_List is a string array i.e. String[]

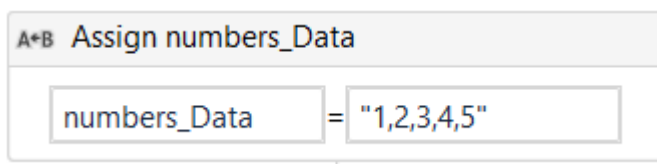
You can also use the split string activity to accomplish the same thing.

Implementation with UiPath

Let's develop a project which takes a string value consisting of numbers separator by comma and displays the values.

Step 1: Drag "Assign" activity into the design panel and initialize the string variable with some sample data as shown below.

```
numbers_Data = "1,2,3,4,5"
```



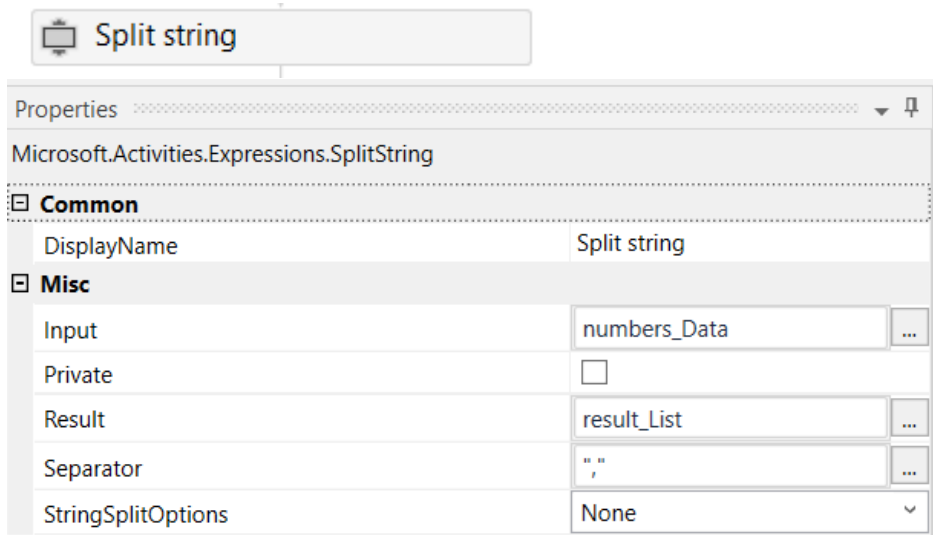
Step 2: Drag "Split String" activity into the design panel and supply the below-mentioned parameters.

Install "Microsoft.Activities" Package ,if you don't find the split activity in your activities panel

Input: numbers_Data

Result: result_List

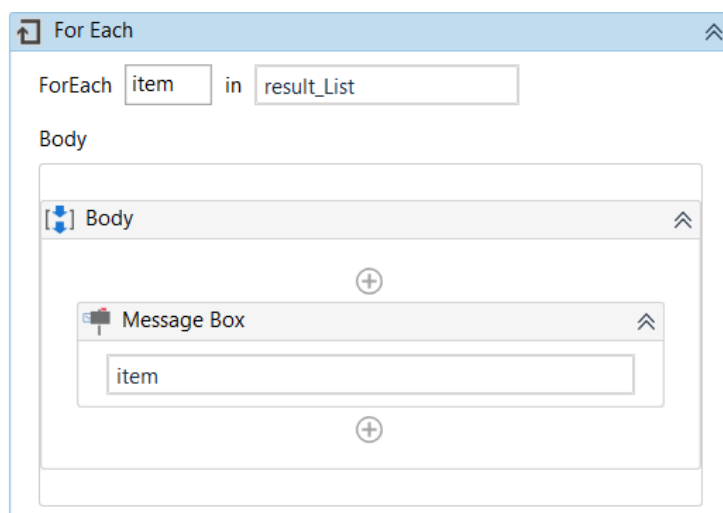
Separator: ",",



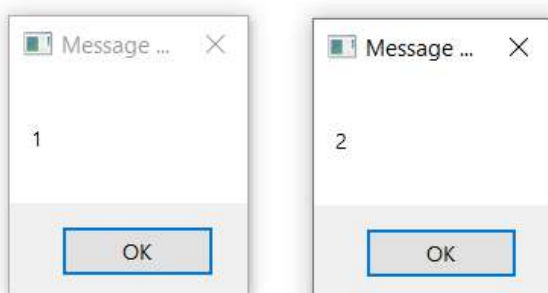
Where,

1. Input is our input String value
2. Result is our output List
3. Separator is the desired delimiter ex: comma, space, hyphen e.t.c

Step 3: Drag a “*For Each*” activity into the design panel and supply the above-created list variable into it. Then, drag a “*Message Box*” activity to display the values available in the list variable.



Step 4: Finally, execute the project to display all the values available in the list on by one!!!



etc.

How to get specific items from an array?

You can access specific items in the result_List array by referencing the index using round brackets. The first item starts from index 0. For example, to get the first item in the result_List array, you pull the zero'th index like so `result_List(0)` this would return 1.

Same applies to the other indices:

```
result_List(0) = 1  
result_List(1) = 2  
result_List(2) = 3  
result_List(3) = 4  
result_List(4) = 5
```

To grab the last item in the array we can use the 4th index in the array like we did above. This is because we know the 4th index is the last item. However, sometimes we don't know how many items are in the array, so to grab the last item in the array we can use the Last method like so:

```
result_List.Last = 5
```

Click the link below to download the example project:

<https://drive.google.com/drive/folders/11xIFbLER0zaXvESGFkl0cjtW09iyfio-?usp=sharing>

4.7 How to Replace Text in a String

We may need to replace characters or a string of characters with something else. We can use an expression like so:

```
sample_String = sample_String.Replace("Before","After")
```

Note:

Set your string variable in place of sample_String

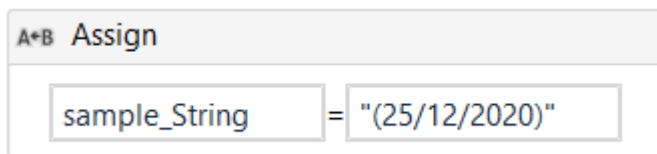
Set "Before" to the character or characters you want to replace in your sample_String

Set "After" to the character or characters you want to replace the "Before" with.

Implementation with UiPath

Let us assume that we have a string of a date in string format that looks like this (25/12/2020) and we want it to look like this 25-12-2020. We can use the Replace method to remove the round brackets and also replace the forward slashes with hyphens.

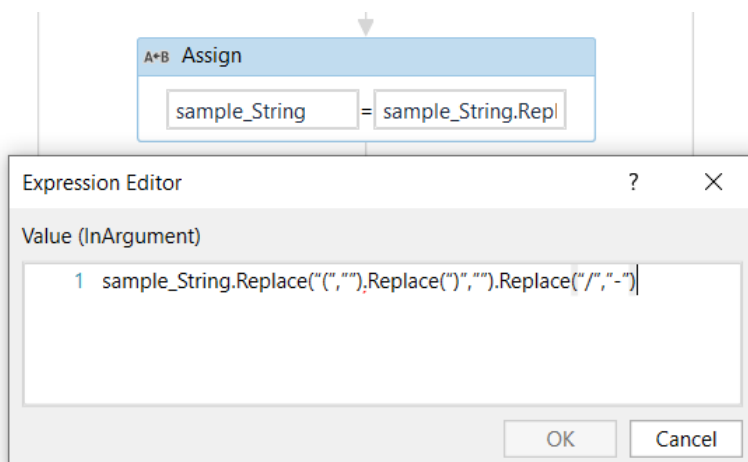
Step 1: We drag in an "Assign" and set the value of the sample_String



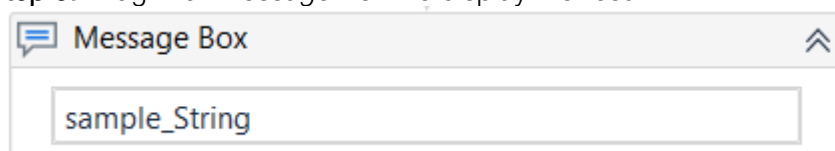
Step 2: We drag in another "Assign" and set the following expression

```
sample_String = sample_String.Replace("(", "").Replace(")", "").Replace("/", "-")
```

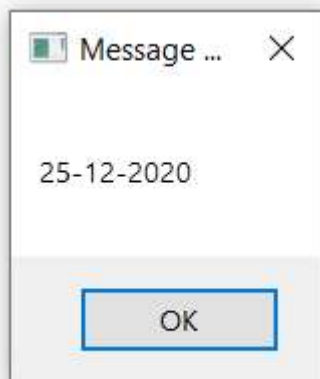
This expression first replaces the open bracket with an empty string, then replaces the close bracket with an empty string, then replaces the forward slash with a hyphen.



Step 3: Drag in a "Message Box" to display the result.



Step 4: Run the workflow



Click the link below to download the example project:

<https://drive.google.com/drive/folders/1wPOVuLYTv3kjNllueEV9whJSz7ekgpFw?usp=sharing>

4.8 How to Find Common/Uncommon Items Between Two Lists

Lists in programming are commonly used to store the collection of items such as String, Int or many other data types. We need to sometimes identify the common/uncommon items between the two *Lists* or two *Arrays*. Let us see how to implement it.

Common Values: `list_1.Intersect(list_2)`

UnCommon Values: `list_1.Except(list_2)`

Note:

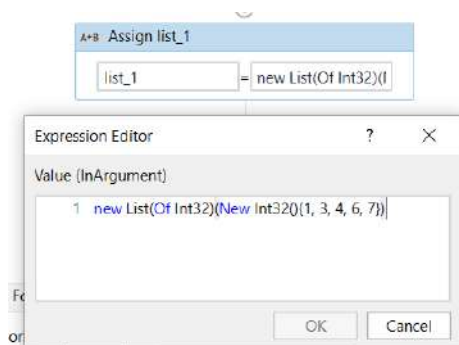
Replace list_1 and list_2 with your appropriate lists

Implementation with UiPath

Let us implement a workflow which takes a two sample lists and displays the common and uncommon values.

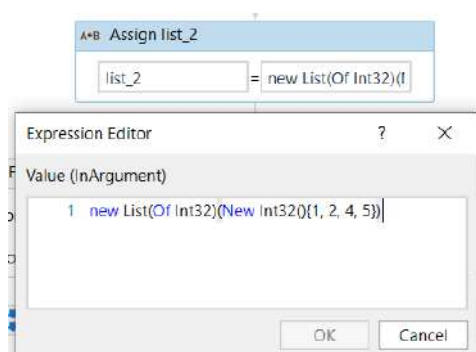
Step 1: Drag an “Assign” activity into the design panel and initialize the list 1 with some sample data as shown below.

```
list_1 = new List(Of Int32)(New Int32(){1, 3, 4, 6, 7})
```



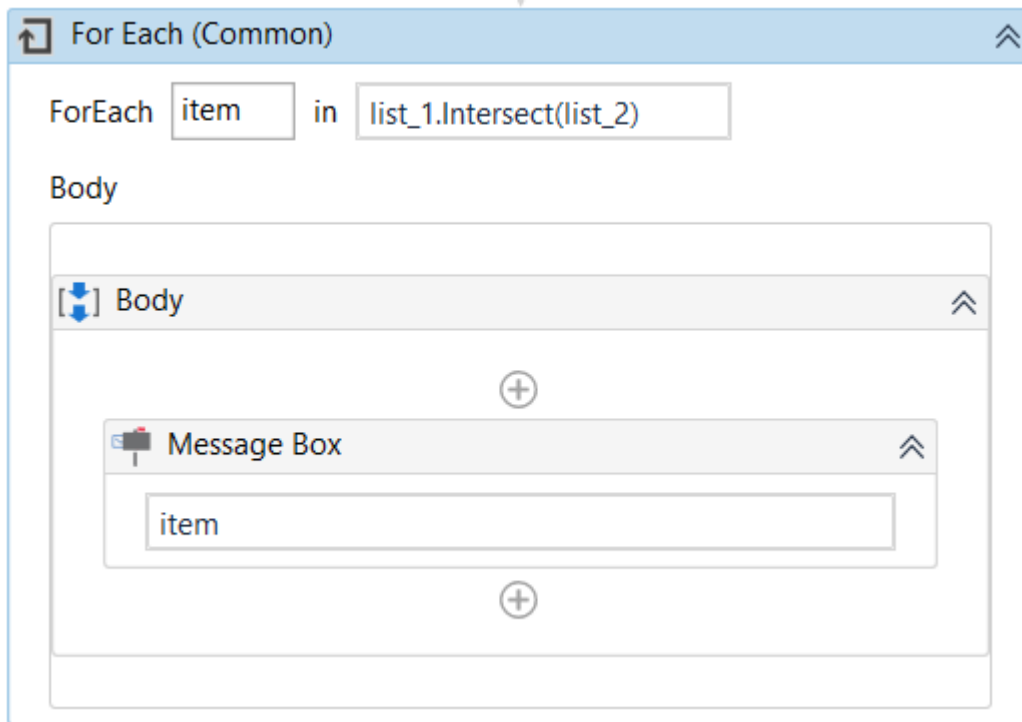
Step 2: Drag another “Assign” activity into the design panel and initialize the list 2 with some sample data as shown below.

```
list_2 = new List(Of Int32)(New Int32(){1, 2, 4, 5})
```



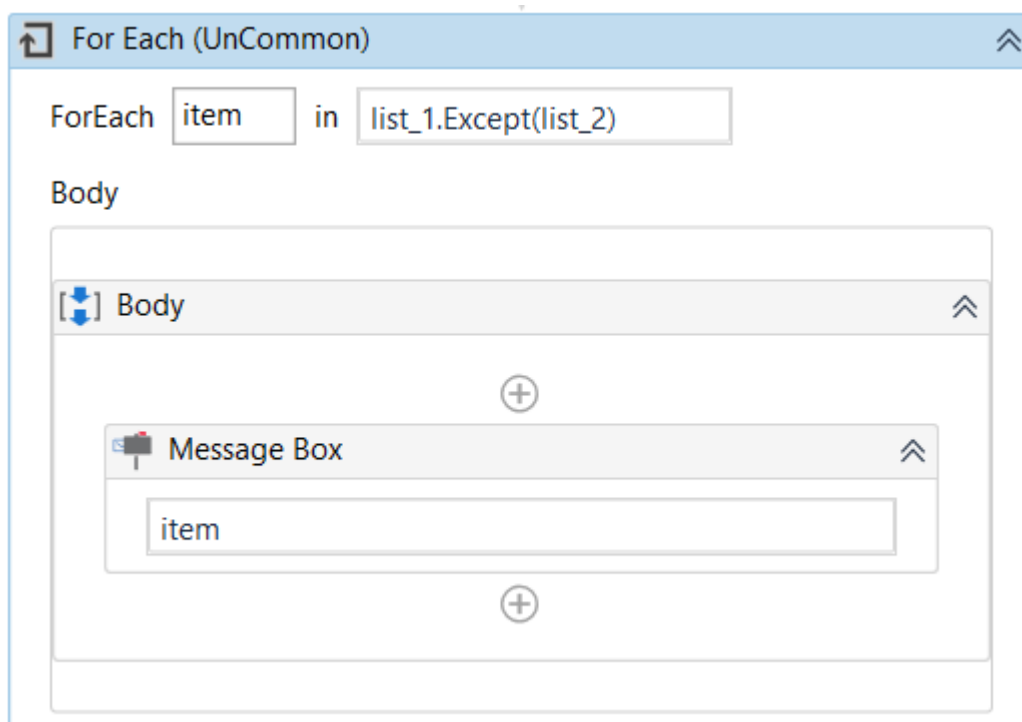
Step 3: Drag a “For Each” activity into the design panel and supply the below-mentioned code into it and use a “Message Box” to display the Common values between two lists

`list_1.Intersect(list_2)`



Step 4: Drag a “For Each” activity into the design panel and supply the below-mentioned code into it and use a “Message Box” to display the Uncommon values between two lists

`list_1.Except(list_2)`



Step 5: Finally, run the workflow.

Click the link below to download the example project:

https://drive.google.com/drive/folders/1ToPantAsBJRGOJ726CVQe_lzWjtI1UXc?usp=sharing

4.9 How to Check If a String is Numeric

Sometimes we need to validate a string is a number before entering it into a system or before trying to convert it to a double or integer. Let us see how to check if the provided string is numeric or if it is not numeric.

```
booleanResult = Information.IsNumeric(user_Input)
```

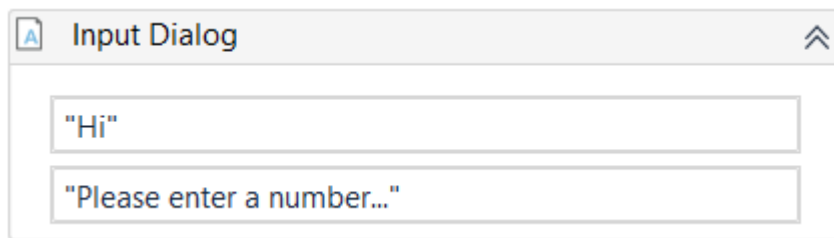
Note:

1. inputVar is the argument to the IsNumeric Function
2. booleanResult returns the true or false value based on the input

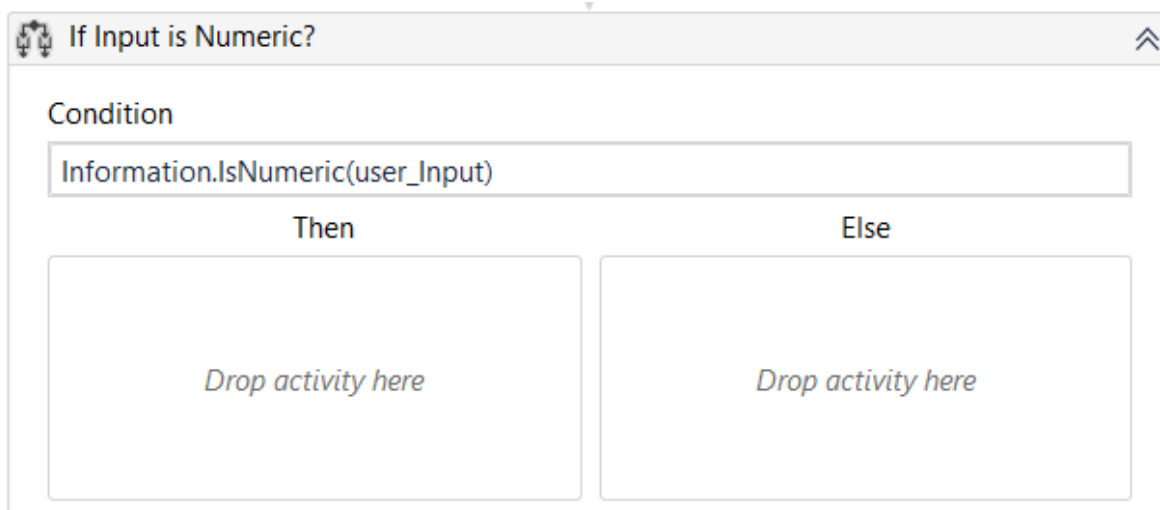
Implementation with UiPath

Let us implement a workflow which takes a string input from the user and checks if it is a numeric or not a numeric.

Step 1: Drag a “*Input Dialog*” activity into the design panel and try to save the user provided value into a variable for further operations.



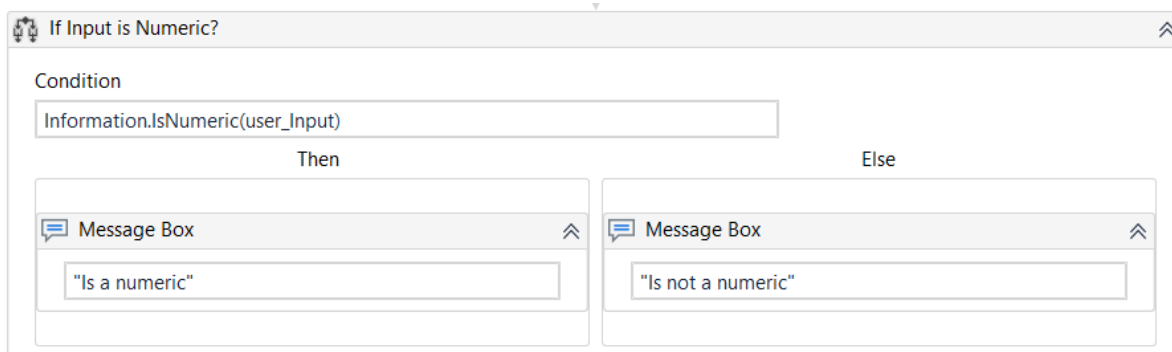
Step 2: Drag “*If*” activity into the design panel and fill it with above-mentioned code.



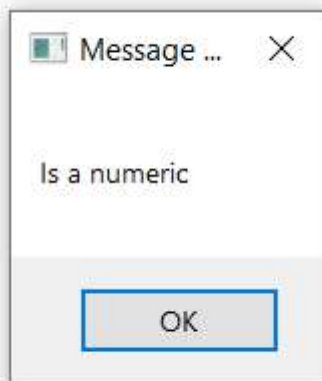
Step 3: Drag two “*Message Box*” activities into the If activity and represent the appropriate messages.

Note:

“If” activity returns true, then “Is a Numeric”
else “Is not a numeric”

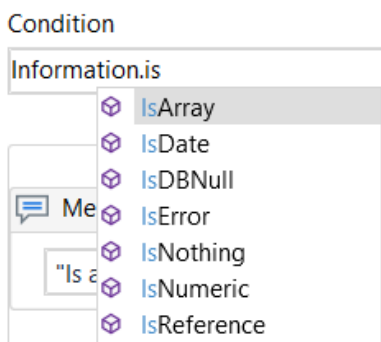


Step 4: Finally, let's run the project



Note:

You can also check for other data types besides just IsNumeric:



Click the link below to download the example project:

<https://drive.google.com/drive/folders/1RfqmC2R3TjUGCQoGQzV0uJHaY7YwpHfC?usp=sharing>

4.10 How to Get the Word Count from a String

For a few business processes, you might be required to validate a sentence and then proceed forward. So how can we find the count of the words in a string? We can split a sentence using a space as a delimiter then count the number of items in the array.

```
no_OfWords = (our_String).Split(" ").Count.ToString
```

Note:

Simply replace the our_String with your required sentence as a string.

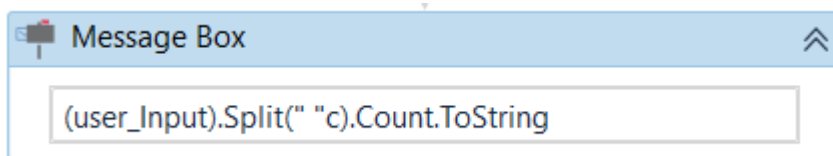
Implementation with UiPath

Let us implement a workflow which takes an input from the user which is a sentence and displays the word count.

Step 1: Drag a *"Input Dialog"* activity into the design panel and save the user provided input into a variable. i.e. "My name is Leon Petrou"



Step 2: Drag a *"Message Box"* activity into the design panel and fill it with above-mentioned code.



Step 3: Run the workflow.



Click the link below to download the example project:

<https://drive.google.com/drive/folders/18OK6-0iSzckHFIMPUsrjwTMiPZsRRCT2?usp=sharing>

4.11 How to Count the Characters in a String

For a few business processes, you might be required to validate a the number of characters in a string before attempting to enter it into a input field. This is because some input fields online have limited characters for input. So how can we find the number of characters in a string? We can use the `.Length` method.

```
no_OfCharacters = (our_String).Length
```

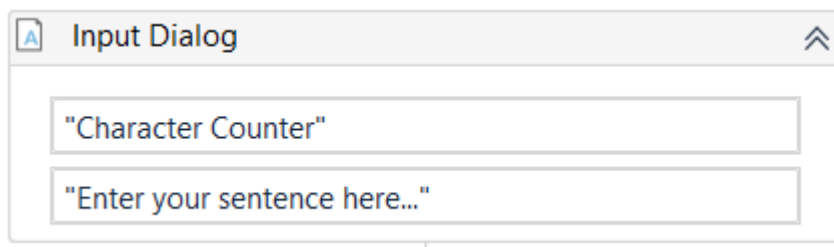
Note:

Simply replace the “our_String” with your required “Sentence”

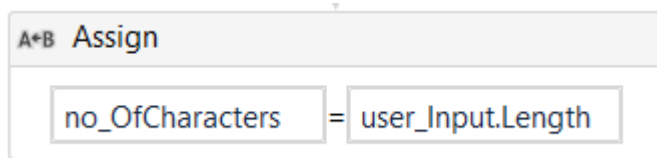
Implementation with UiPath

Let us implement a workflow which takes an input from the user which is a sentence and displays the character count. Only a character count of 10 or less is valid, else it is invalid.

Step 1: Drag an “*Input Dialog*” activity into the design panel and save the user provided input into a variable. i.e. “Leon Petrou”

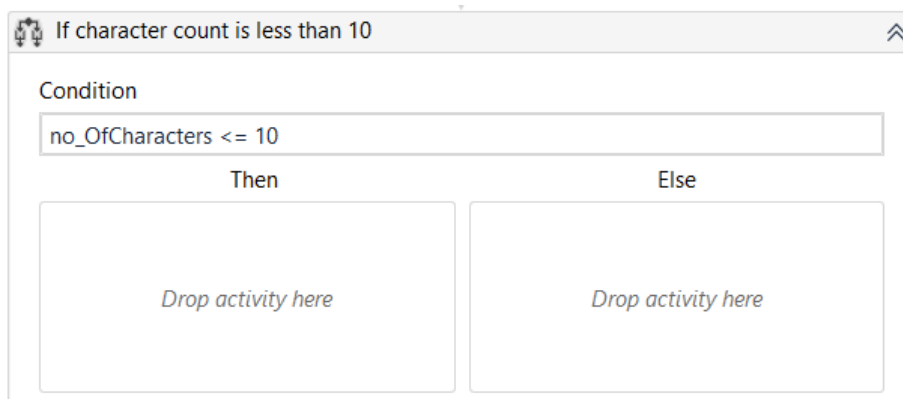


Step 2: Drag an “*Assign*” activity into the design panel and enter the above-mentioned code. Set the no_OfCharacters to an Int32 variable type.



Step 3: Check if the total number of characters is less than or equal to 10 using this code:

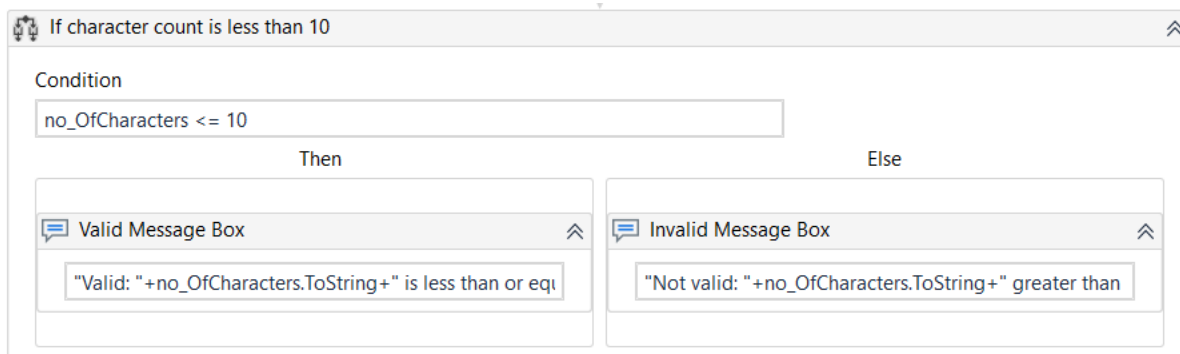
```
no_OfCharacters <= 10
```



Step 4: Add two “Message Box” activities with the following code:

Then: "Valid: " + no_OfCharacters.ToString + " is less than or equal to 10"

Else: "Not valid: " + no_OfCharacters.ToString + " greater than 10"



Step 5: Let's input “Leon Petrou” and run the workflow. We get this output:



Click the link below to download the example project:

<https://drive.google.com/drive/folders/1xkXk-VXoeUfmjb1g4xS6yCWLAINJytHS?usp=sharing>

4.12 How to Convert a List to an Array

Converting one datatype to another is one of the most common things we do in programming.

Let's see how to convert a list to an array and Vice Versa. This is useful because both lists and arrays have their own sets of methods and functions that may be useful. For example, you can add and remove items from a list but you can't do that with an array. In this example we will:

1. Convert List to Array
2. Convert Array To List

Convert List to Array

```
our_Array = our_List.ToArray
```

Let us implement a workflow which converts a "List" to an "Array"

Step 1: Create variables as shown below

Name	Variable type	Scope	Default
our_List	List<String>	Sequence	Enter a VB expression
our_Array	String[]	Sequence	Enter a VB expression
Create Variable			

Step 2: Drag an "Assign" activity into the design panel and initialize the list of string type.

A*B Assign our_List

our_List

 =

New List(Of String)

Step 3: Drag another "Assign" activity into the design panel
Use the above-mentioned code and its done.

A*B Assign Our_Array

our_Array

 =

our_List.ToArray

Click the link below to download the example project:

<https://drive.google.com/drive/folders/154g98JoEk3MEIT39jj3xDHsnBCOxKQDW?usp=sharing>

Convert Array to List

Let us implement a workflow which converts an "Array" to "List"

Step 1: Create variables as shown below.

Name	Variable type	Scope	Default
our_List	List<String>	Sequence	<i>Enter a VB expression</i>
our_Array	String[]	Sequence	<i>Enter a VB expression</i>
Create Variable			

Step 2: Drag “Assign” activity into the design panel and initialize the Array of string type.

A*B Assign

our_Array = {"Leon", "Petrou"}

Step 3: Drag another “Assign” activity into the design panel. Use the above-mentioned code and its done.

A*B Assign

our_List = our_Array.ToList

Click the link below to download the example project:

https://drive.google.com/drive/folders/1-AUilPDb-6pYVs89SKL_jw2AKtwYDZ2H?usp=sharing

4.13 How to Reverse a String

Reversing a string is something that may be required when dealing with strings. That can be done by creating an array of characters and reversing that array then rejoining all the characters together again.

```
String reversedString = stringtoBeReversed.AsEnumerable.Reverse.ToArray
```

Note:

1. Simply replace the “stringToBeReversed” with your required “string”
2. It returns an array of char *and then*;

```
string.Join("",reversedString)
```

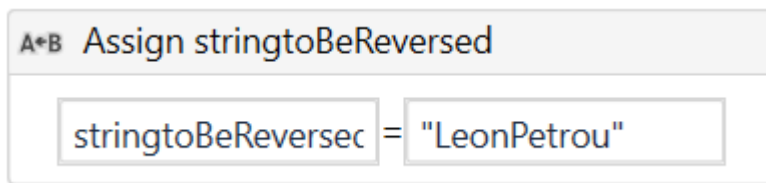
Note:

1. Here we are converting an array of char to string type
2. Here “” indicates the separator which is an empty string.

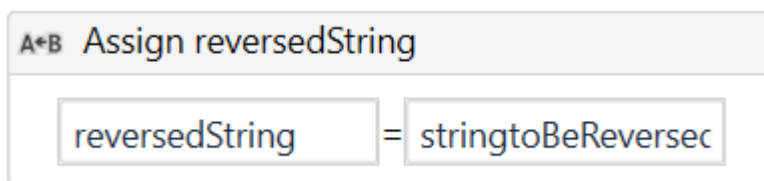
Implementation with UiPath

Let us implement a workflow which takes “LeonPetrou” as input and outputs the reverse of it, i.e. “uortePhoeL”

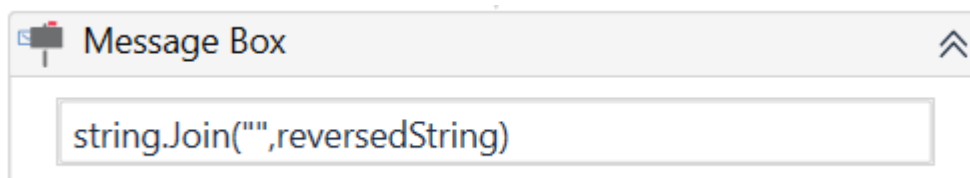
Step 1: Drag an “Assign” activity into the design panel and pass a value to it.



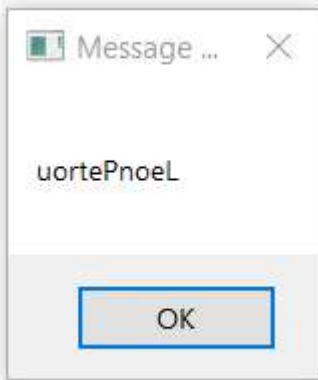
Step 2: Drag an “Assign” activity into the design panel and obtain an array of char in reverse order by using the above-mentioned first code.



Step 3: Drag a “Message Box” activity into the design panel and obtain the reverse by using the above-mentioned second code.



Step 4: Run the workflow. My name is now backwards.



Click the link below to download the example project:

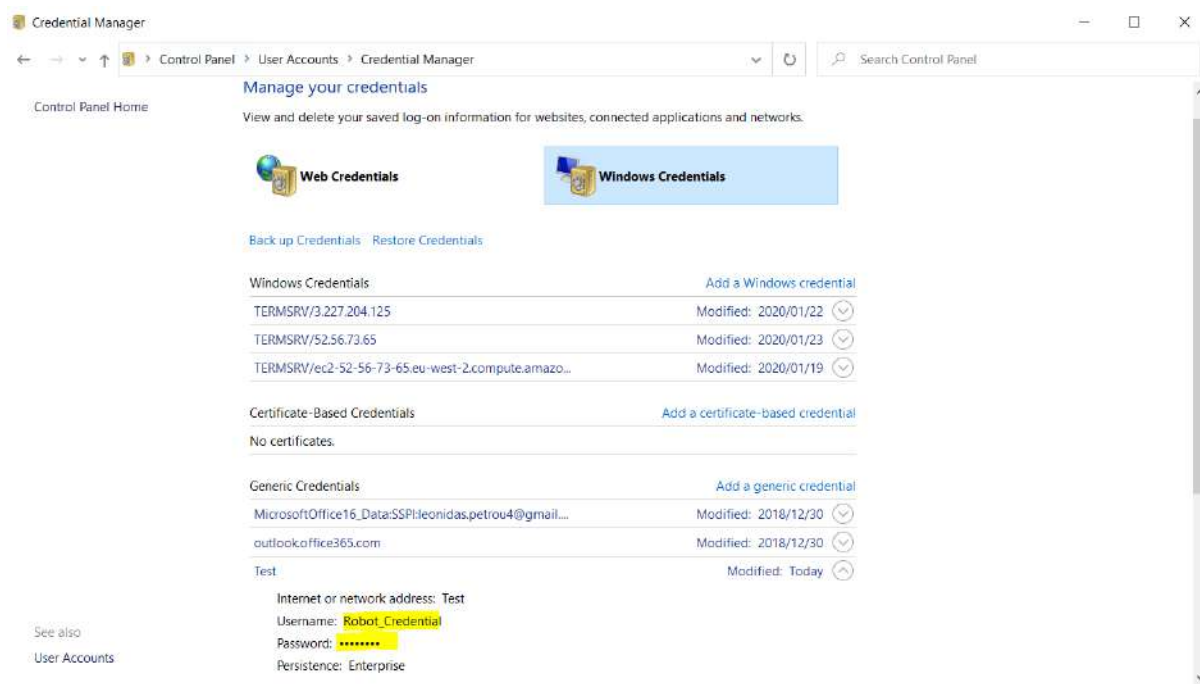
<https://drive.google.com/drive/folders/1l6VcGnPQTgYRNxMYUqFhC7DWGYqlXTgt?usp=sharing>

4.14 How to Convert a SecureString DataType to a String DataType

Almost all processes fetch credentials from Windows Credentials or Orchestrator which are of data type SecureString. However, the Mail activities that the UiPath provide us with only accepts the string data type for the password field. And so we will need to convert the SecureString to a String. Let's see how to handle it:

1. Fetch username and password from Windows Credentials.
2. Convert Password from SecureString to String Datatype.

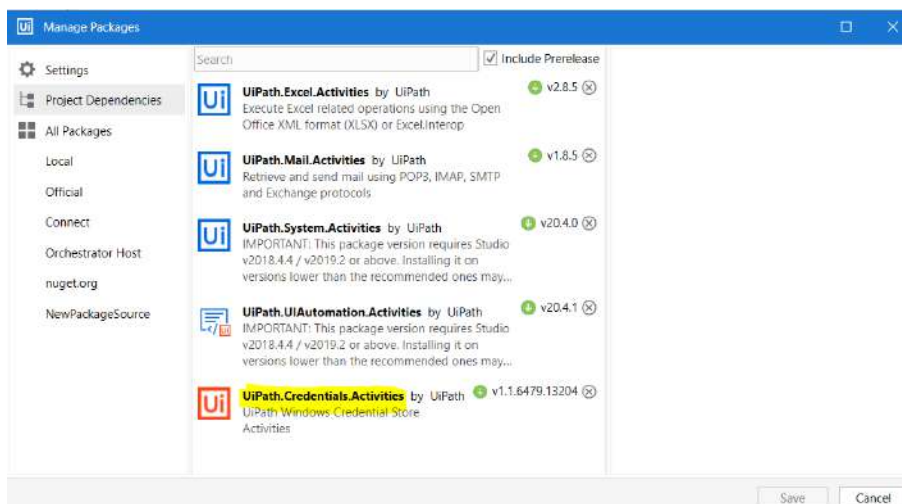
Assuming we have a credential saved with the name *Robot_Credential* under Windows Credentials in your control panel as shown below.



Now let us see how to fetch it and convert the SecureString to string data type...

Step 1: Fetch username and password from Windows Credentials

First we must install the UiPath.Credentials.Activities package:



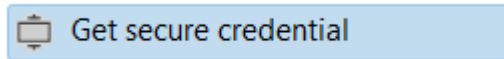
Use a “*Get Secure Credentials*” activity:

Target: Credentials Name (String)

Results: Status Code (Boolean)

Password: Fetched Password (SecureString)

Username: Fetched Username (String)



Properties

UiPath.Credentials.Activities.GetSecureCredential

Common

DisplayName: Get secure credential

Input

CredentialType: Generic

PersistenceType: Enterprise

Target: "Robot_Credential" ...

Misc

Private: ☐

Result: credentialFound ...

Output

Password: Password ...

Username: Username ...

Step 2: Convert Password from SecureString to String Datatype. Use an “*Assign*” activity:

`password_String = new System.Net.NetworkCredential(string.Empty, Password).Password`

Assign

password_String = new System.Net.Ne

Expression Editor

Value (InArgument)

1 new System.Net.NetworkCredential(string.Empty, Password).Password

OK Cancel

Then we can pass the password_String variable to a “*Message Box*”.

Message Box

password_String

Step 3: Then the output:



Click the link below to download the example project:

https://drive.google.com/drive/folders/10nkLSSS_Z6JmnS_95CW6UyQP-Xd35QV?usp=sharing

5 Email

5.1 How to Retrieve E-mail Body / Subject / Sender Details

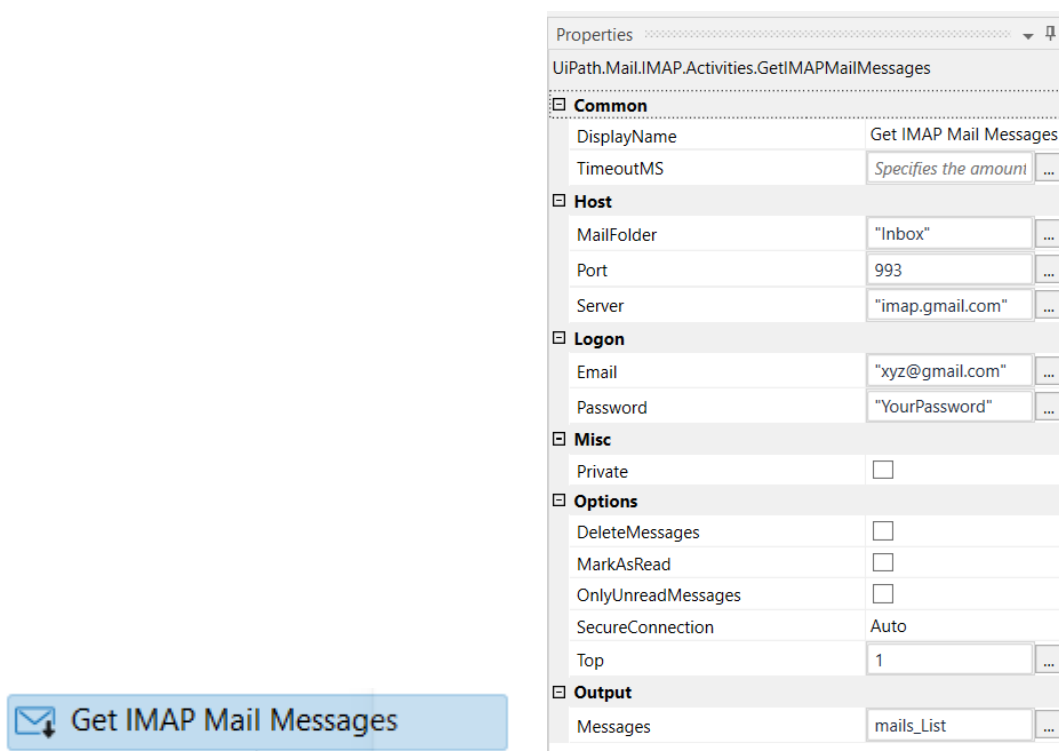
Emails are widely used in automation for the purpose of communication between users and robots. Sometimes emails act as the input to a process. Let us see how to retrieve a mail's Body, Subject and Sender details using UiPath activities.

Implementation with UiPath

Let us implement a workflow which will read the emails from Gmail server and retrieve the emails.

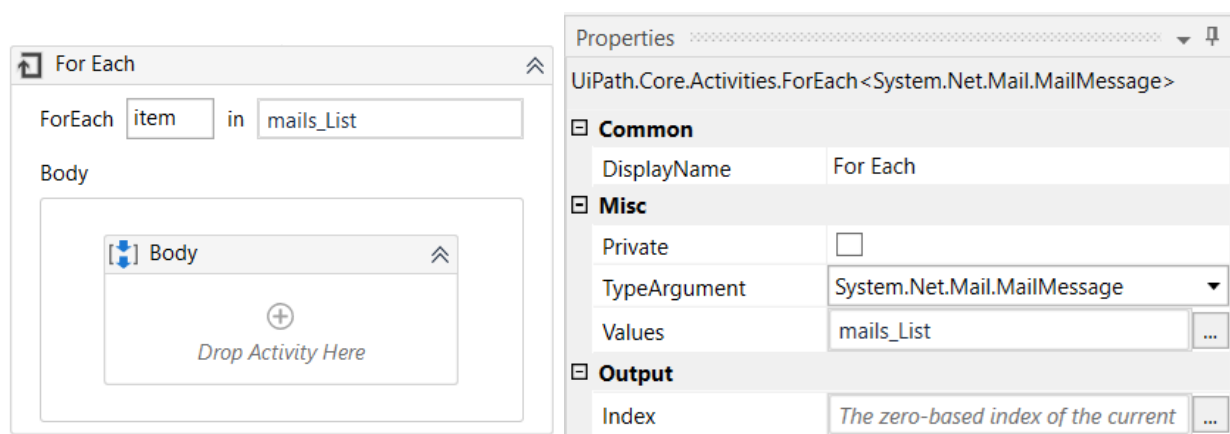
Step 1: Drag a “Get IMAP Mail Messages” activity into the design panel and supply the required parameters to it as shown below. Remember to allow “less secure apps” in your gmail settings. You can do that here:

<https://myaccount.google.com/lesssecureapps>

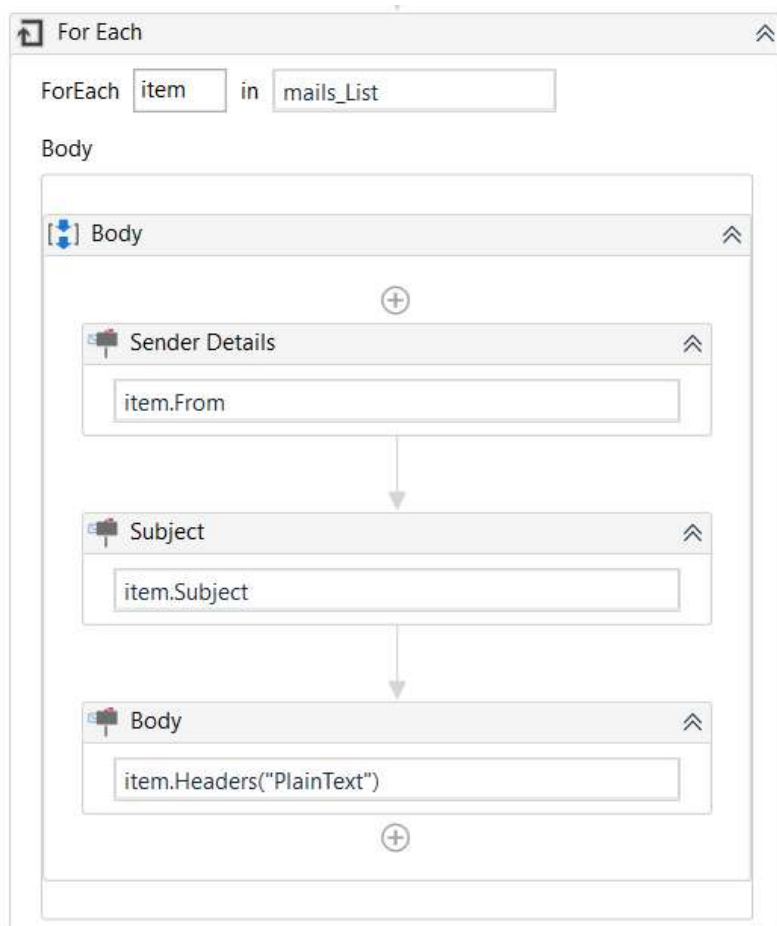


Step 2: Drag a “For Each” activity into the design panel and pass the above-created mail List to it.

Change the “TypeArgument” in the properties panel to *System.Net.Mail.MailMessage*

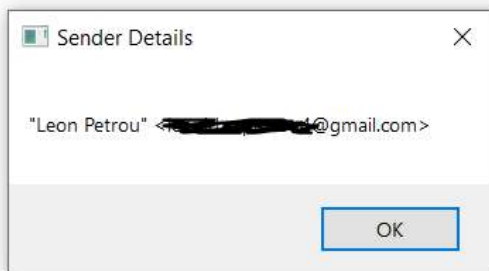


Step 3: Drag three “Message Box” activities into the design panel to display the Sender details, Subject and Body of the received mail.

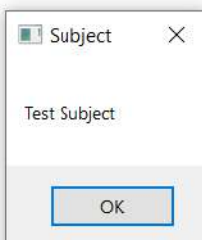


Step 3: Now, execute the workflow.

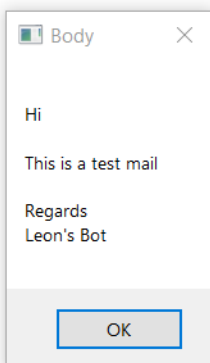
Mail Sender:



Mail Subject:



Mail Body:



Click the link below to download the example project:

https://drive.google.com/drive/folders/1gynb4AgdD_-fmHwoK3aeqHVC5UNTnb4g?usp=sharing

5.2 How to Download Specific Files from a Mail

Mails are widely used in automation processes to receive inputs or to notify users of the bot's progress. UiPath provides us the ability to download the attachments from emails. By default, it downloads all the attachments available, but what if we only want to download a specific extension files? Let's see how to do that.

Filter = `".*(.xlsx|.XLSX|.xls|.pdf)"`

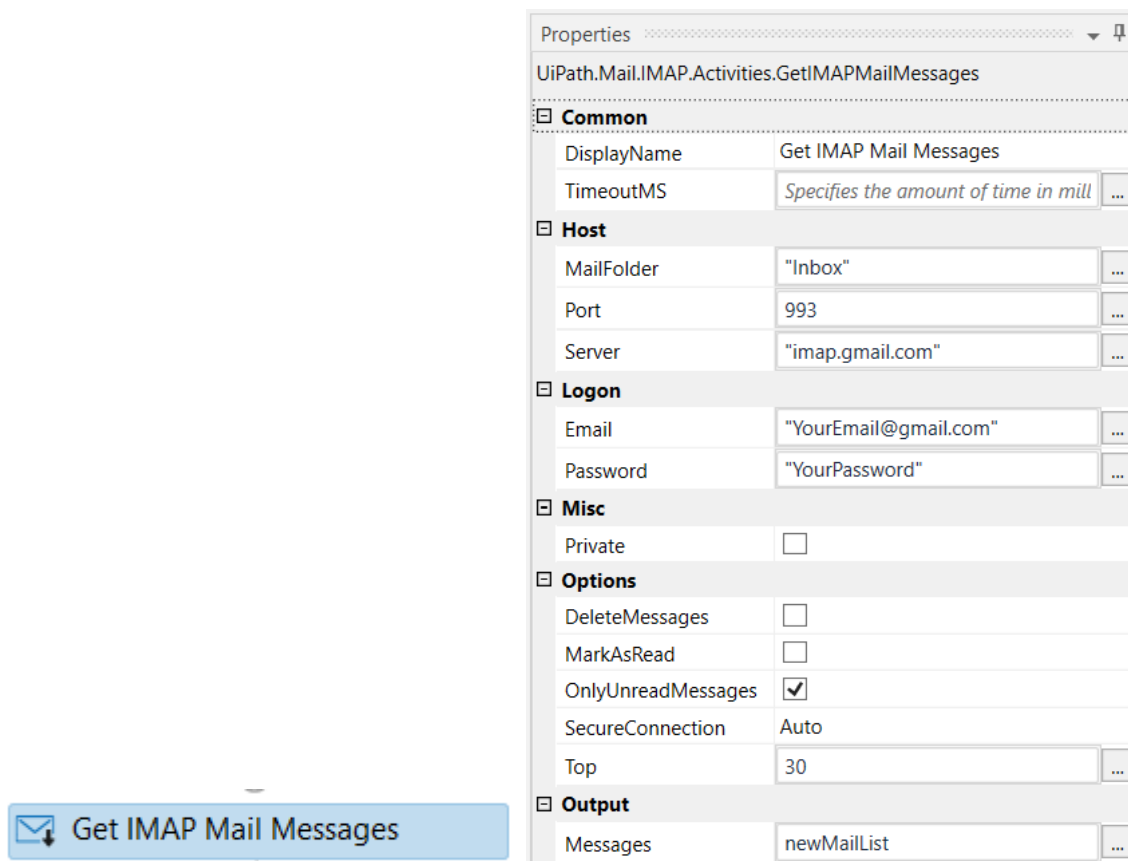
Note:

Pass the *EXTENSIONS* according to your choice. This is a *Regex* expression where the `|` represents "Or". You can also set this to only download files with specific names.

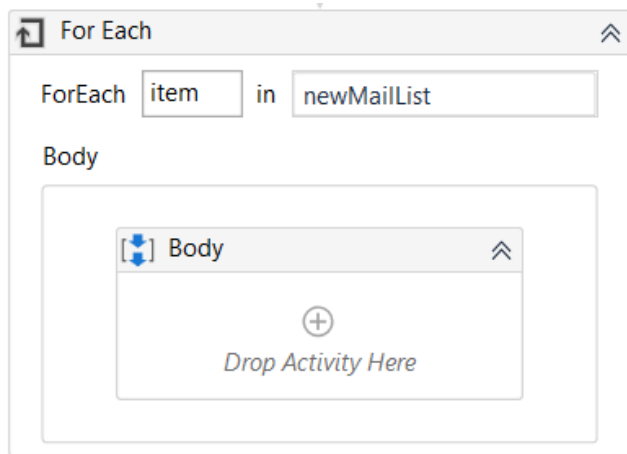
Implementation with UiPath

Let us implement a workflow that *read emails* from an account and then download the attachments according to our requirement.

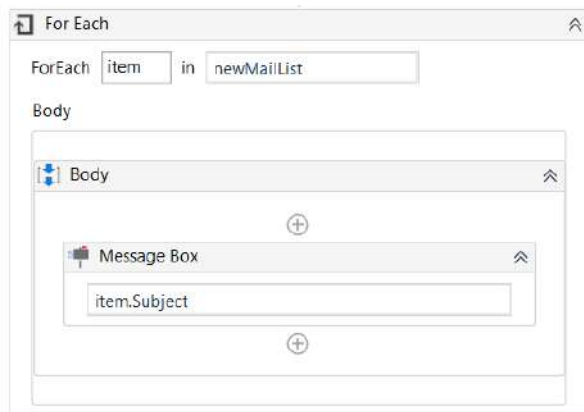
Step 1: Drag a "Get IMAP Mail Messages" activity into the design panel and populate its properties with required fields.



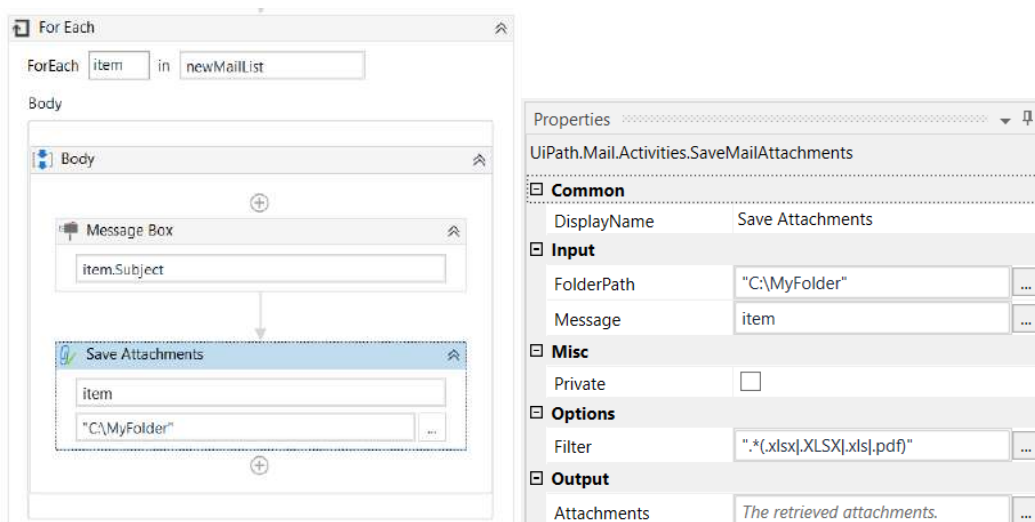
Step 2: Drag a “For Each” activity into the design panel and pass the above “newMailList” variable to it and change the type in the properties window to “System.Net.Mail.MailMessage”.



Step 3: Drag a “Message Box” activity into the “For Each” activity and pass the `item.Subject` value into it to display the subject.



Step 4: Drag a “Save Attachments” activity into the “For Each” activity just below the “Message Box” activity and pass the required fields as shown below.



Step 5: Finally, we run the workflow. We should be able to download the files as per our defined criteria.

← → ▾ ↑ 📁 > This PC > Windows (C:) > MyFolder				
★ Quick access	Name	Date modified	Type	Size
	📄 FirstExcel.xlsx	2020/06/07 17:38	Microsoft Excel W...	7 KB
	📄 MyPDF.pdf	2020/06/01 14:18	Adobe Acrobat D...	451 KB
	📄 SecondExcel.xlsx	2020/06/07 17:38	Microsoft Excel W...	7 KB

Click the link below to download the example project:

https://drive.google.com/drive/folders/1uOv0tRT9YArRAF1-jXV5QG_7Xt8MkacD?usp=sharing

5.3 How to Reverse a Mail List

Emails are involved in many of the processes that we automate. Using UiPath's default activities we will receive the emails in FIFO (First In First Out) order, but what if we want them to be in LIFO (Last in First Out)? Let us see how to do that?

```
new_MailList = old_MailList.OrderBy(Function(x) x.Headers("date")).ToList
```

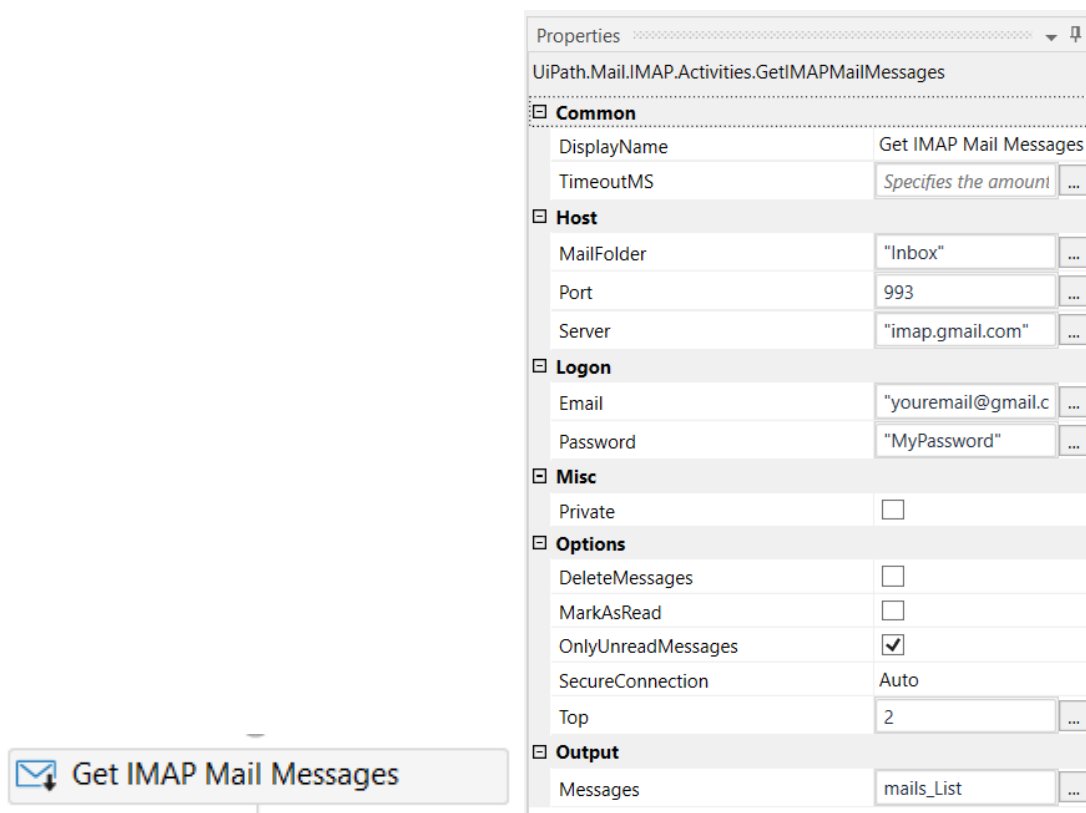
Note:

1. old_MailList is the default list that you get from the output of the Get Mail Message activity
2. New_MailList is the new list with mails ordered in LIFO

Implementation with UiPath

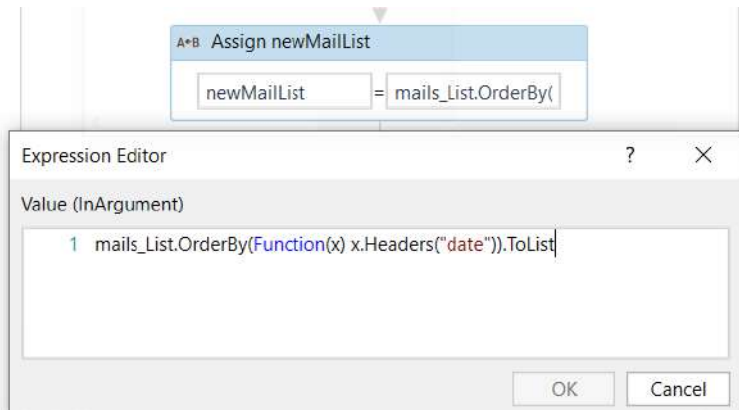
Let us implement a workflow that *read emails* from an account and then provides us the emails in *LIFO order*, instead of FIFO.

Step 1: Drag a “Get IMAP Mail Messages” activity into the design panel and populate its properties with required fields. Use your gmail address and password. Remember to allow “less secure apps” in your gmail settings. You can do that here: <https://myaccount.google.com/lesssecureapps>

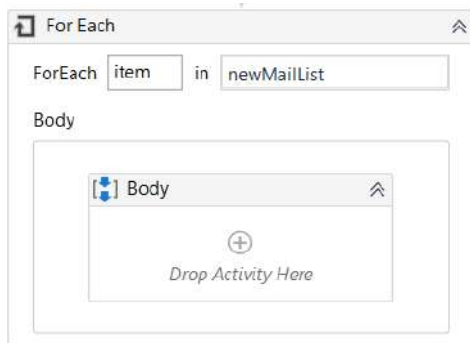


Step 2: Drag an “Assign” activity into the design panel and populate it with an above-mentioned code example:

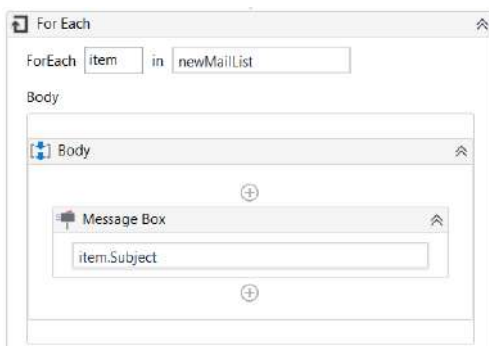
```
mails_List.OrderBy(Function(x) x.Headers("date")).ToList
```



Step 3: Drag a “For Each” activity into the design panel and pass the above “newMailList” variable to it and change the type in the properties window to “System.Net.Mail.MailMessage”



Step 4: Drag “Message Box” activity into the For Each activity and pass the “item.Subject” value into it to display the subject.



Step 5: Run the workflow.



Click the link below to download the example project:

<https://drive.google.com/drive/folders/1LpiTH3g5BJ2HHix3q5YLjvguWkraGdI2?usp=sharing>

6 DateTime Operations

6.1 How to Find the Date Difference (Days, Months and Years)

Date operations are frequently used in business automation. In a few business automations, decisions have to be taken based on the date differences. Let's see how to calculate the difference between two dates using dates written in string format.

```
required_Difference =  
DateDiff(DateInterval.Day,Convert.ToDateTime(date1),Convert.ToDateTime(date2)).ToString
```

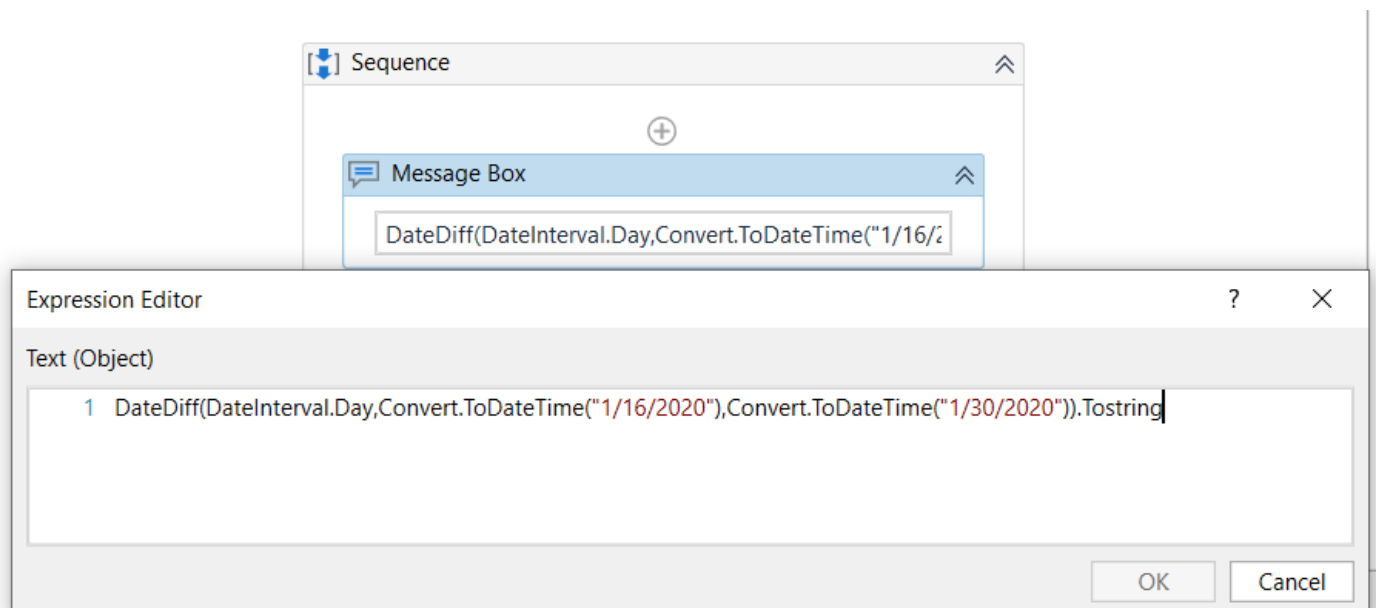
Note:

1. Replace the "date1" with your first "Date"
2. Replace the "date2" with your second "Date"
3. Replace "required_Difference" with your output string variable.
4. For day difference use – Dateinterval.Day
5. For month difference use – Dateinterval.Month
6. For year difference use – Dateinterval.Year

Implementation with UiPath

Let us implement a workflow which takes two dates and displays the day difference.

Step 1: Drag a "Message Box" activity into the design panel and populate it with the above-mentioned code.



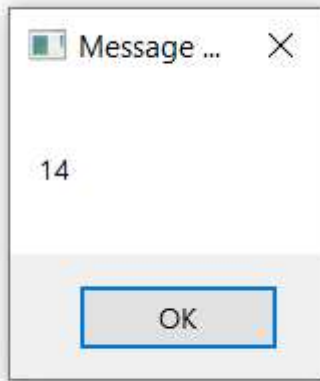
Note:

It follows the pattern of "month/day/year"

Date1 = "1/16/2020"

Date2 = "1/30/2020"

Step 2: Finally, when you run the workflow, we will get the day difference, in this case 14 days.



Similarly, you can calculate the month and year differences (see Note above).

Click the link below to download the example project:

https://drive.google.com/drive/folders/1Rou3zFx_0g68zFoY0nNTlvXw7gAenLWP?usp=sharing

6.2 How to Compare Time in UiPath

In our automation processes, we might come across situations where we are required to switch between various workflows based on time.

So how can we do that?

```
Datetime.Parse(DateTime.Now.ToString("HH:mm:ss")) < datetime.Parse(TimeVar)
```

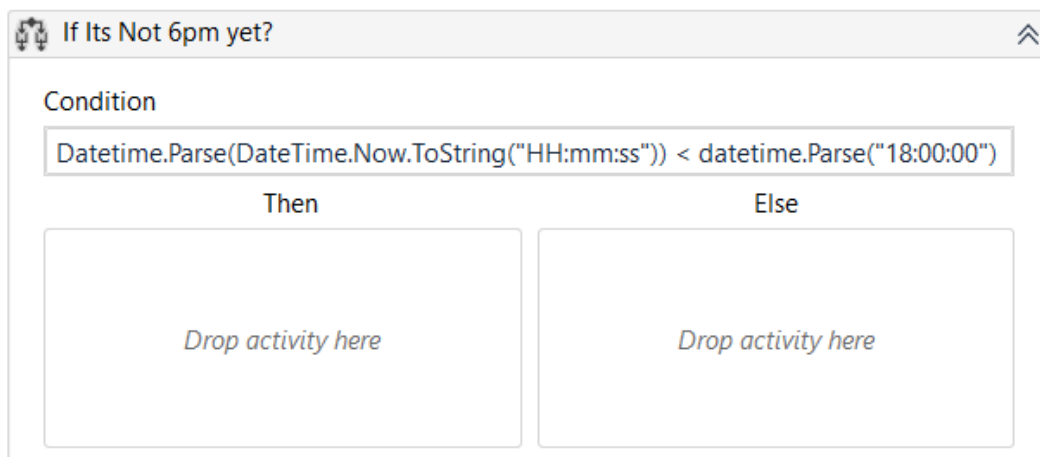
Note:

Replace the "TimeVar" with your required "TimeSpan". E.g. 18:00:00 or 02:00:00 etc.

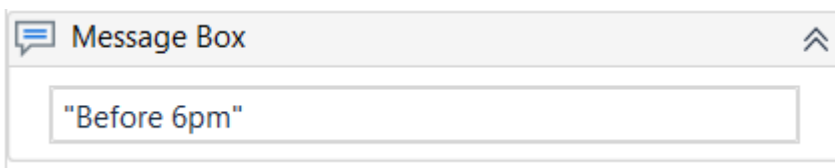
Implementation with UiPath

Let us implement a workflow which compares the current time with the provided time and displays the appropriate messages.

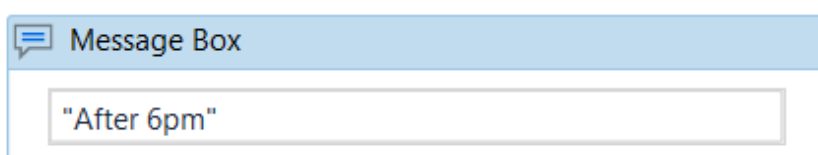
Step 1: Drag an "If" activity into the design panel and populates its condition property with the above-mentioned code with "18:00:00" in it.



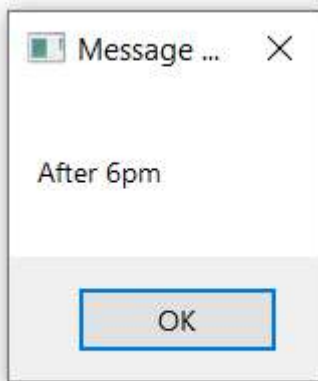
Step 2: Drag a "Message Box" activity into the design panel and write "Before 6pm" in it.



Step 3: Drag another "Message Box" activity into the design panel and write "After 6pm" in it.



Step 4: Based on the current time set on your windows machine, we can see the appropriate message.



Click the link below to download the example project:

<https://drive.google.com/drive/folders/1oIVoKPvrnZRWPw3jf4a2RIkIW-SWJctT?usp=sharing>

6.3 How to Get the Amount of Days in a Month

Sometimes we need to count the number of days in a given month. For example, sometimes a specific process must happen on the last day of the month or a day relative to that. Let's see how to do get the number of days in a month.

```
Int date_Variable = Datetime.DaysInMonth(year,month)
```

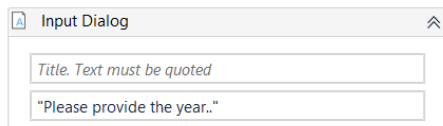
Note:

1. Replace the "year" with your required "Year"
2. Replace the "month" with your required "Month"

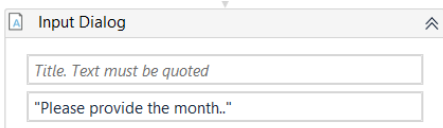
Implementation with UiPath

Let us implement a workflow which takes "Year" and "Month" as input and outputs the number of days in that month i.e.

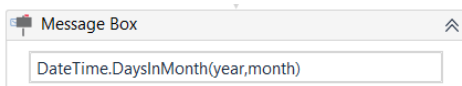
Step 1: Drag an *"Input Dialogue"* activity into the design panel and create a variable to save the *"year"* provided by the user.



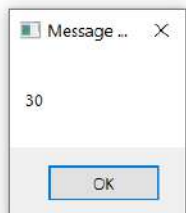
Step 2: Drag another *"Input Dialogue"* activity into the design panel and create a variable to save the *"month"* provided by the user.



Step 3: Drag a *"Message Box"* activity into the design panel to display the days count.



Step 4: Finally!!!



Click the link below to download the example project:

<https://drive.google.com/drive/folders/1oBghLjI3ILX9E7JkQ04CD65IRkM8l43J?usp=sharing>

6.4 How to Convert a DateTime to String

Different business processes may demand different DateTime formats. Check out the patterns of DateTime available and pick the appropriate format

Format	E.g. Result
DateTime.Now.ToString("MM/dd/yyyy")	05/29/2015
DateTime.Now.ToString("dddd, dd MMMM yyyy")	Friday, 29 May 2015
DateTime.Now.ToString("dddd, dd MMMM yyyy")	Friday, 29 May 2015 05:50
DateTime.Now.ToString("dddd, dd MMMM yyyy")	Friday, 29 May 2015 05:50 AM
DateTime.Now.ToString("dddd, dd MMMM yyyy")	Friday, 29 May 2015 5:50
DateTime.Now.ToString("dddd, dd MMMM yyyy")	Friday, 29 May 2015 5:50 AM
DateTime.Now.ToString("dddd, dd MMMM yyyy HH:mm:ss")	Friday, 29 May 2015 05:50:06
DateTime.Now.ToString("MM/dd/yyyy HH:mm")	05/29/2015 05:50
DateTime.Now.ToString("MM/dd/yyyy hh:mm tt")	05/29/2015 05:50 AM
DateTime.Now.ToString("MM/dd/yyyy H:mm")	05/29/2015 5:50
DateTime.Now.ToString("MM/dd/yyyy h:mm tt")	05/29/2015 5:50 AM
DateTime.Now.ToString("MM/dd/yyyy HH:mm:ss")	05/29/2015 05:50:06
DateTime.Now.ToString("MMMM dd")	May 29
DateTime.Now.ToString("yyyy-MM-ddT'HH':'mm':'ss.fffffffK")	2015-05-16T05:50:06.7199222-04:00
DateTime.Now.ToString("ddd, dd MMM yy HH':'mm':'ss 'GMT'")	Fri, 16 May 2015 05:50:06 GMT
DateTime.Now.ToString("yyyy-MM-ddT'HH':'mm':'ss")	2015-05-16T05:50:06
DateTime.Now.ToString("HH:mm")	05:50
DateTime.Now.ToString("hh:mm tt")	05:50 AM
DateTime.Now.ToString("H:mm")	5:50
DateTime.Now.ToString("h:mm tt")	5:50 AM
DateTime.Now.ToString("HH:mm:ss")	05:50:06
DateTime.Now.ToString("yyyy MMMM")	2015 May

1. *d >> Represents the day of the month i.e 1,2,16 or 31*
2. *dd >> Represents the day of the month i.e 01,05,14 or 31*
3. *ddd >> Represents the abbreviated name of the day i.e Mon, Tues or Wed*
4. *dddd-> Represents the full name of the day i.e Monday, Tuesday or Wednesday*
5. *M >> Represents the month number i.e 1,5 or 12*
6. *MM >> Represents the month number with leading zero i.e 04,09 or 12*
7. *MMM >> Represents the abbreviated month Name i.e Jan, May or Dec*
8. *MMMM >> Represents the full month name i.e. January, June or December*
9. *y >> Represents the year i.e. 2019 as 19*
10. *yy >> Represents the year with a leading zero i.e. 2019 as 019*
11. *yyy >> Represents the year i.e 2019*
12. *yyyy >> Represents the year i.e 2019*
13. *h >> Represents the 12-hour clock i.e 4,11 or 2*
14. *hh >> Represents the 12-hour clock with a leading 0 i.e. 04,05 or 12*
15. *H >> Represents the 24-hour clock i.e 13,18 or 22*
16. *HH >> Represents the 24-hour clock with a leading 0 i.e. 04,09 or 22*
17. *m >> Represents the minutes i.e. 12,40 or 56*
18. *mm >> Represents the minutes with a leading zero i.e 04,09 or 23*
19. *s >> Represents the seconds i.e 9,35 or 40*
20. *ss >> Represents the seconds with leading zero i.e. 04,09 or 35*

6.5 How to Convert Milliseconds to TimeSpan

There might be situations where we might need to pass time as TimeSpan rather than in Milliseconds. This occurs with the delay activity which requires a Timespan as input. Let us see how to do that...

`Timespan.FromMilliseconds(milliseconds)`

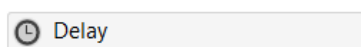
Note:

Where milliseconds is an integer in milliseconds e.g. 3000

Implementation with UiPath

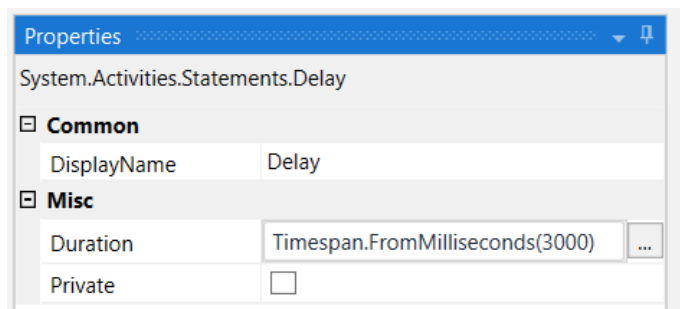
Let us convert 3000 Milliseconds to TimeSpan i.e. 00:00:03

Step 1: Drag a “Delay” activity into the design panel



Step 2: Type the above code with 3000 into the “Duration” property of Delay activity.

i.e. `Timespan.FromMilliseconds(3000)`



Click the link below to download the example project:

<https://drive.google.com/drive/folders/1pZNHS1I79TnF-59x5sOAFmyWCB3qetNC?usp=sharing>

6.6 How to Convert TimeStamp to 12/24hrs Format

Different processes may demand different DateTime format based on their requirements. Let's see how to convert TimeStamp formats...

1. DateTime in 12-hour format
2. DateTime in 24-hour format

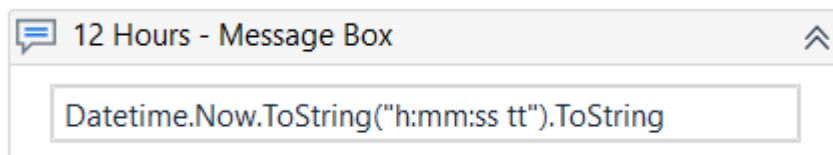
6.6.1 DateTime in 12-hour format

`DateTime.Now.ToString("h:mm:ss tt")`

Where,

- DateTime.Now = Gives Current Date and Time
- .ToString("h:mm:ss tt") = Converts the Current Date and Time to 12 hours format

Drag a "Message box" activity into the design panel and type the above code.



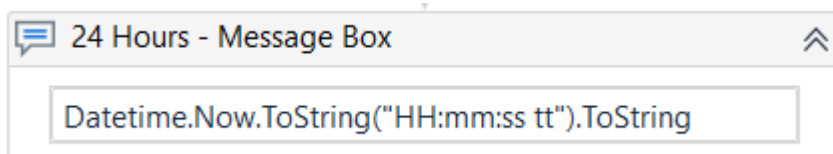
6.6.2 DateTime in 24-hour format

`Now.ToString("HH:mm:ss tt")`

Where,

- DateTime.Now = Gives Current Date and Time
- .ToString("HH:mm:ss tt") = Converts the string to 24 hours format

Drag a "Message box" activity into the design panel and type the above code.



Click the link below to download the example project:

<https://drive.google.com/drive/folders/1tnrnoip0HSCulaF9sPzguFYyYlQxZ9na?usp=sharing>

6.7 How to Convert String to DateTime and DateTime Manipulation

In this module, we will be looking into different DateTime Manipulations

1. Converting String to DateTime Datatype
2. Converting Date from one String format to another desired format
3. Including Hours, Minutes and Seconds to your Date
4. Adding or removing days from your respective date

6.7.1 Converting String to DateTime Datatype

Let us implement a workflow that takes a string and converts it to a DateTime data type.

Step 1: Drag an “Assign” activities into the design panel and assign the string value and the format of DateTime.

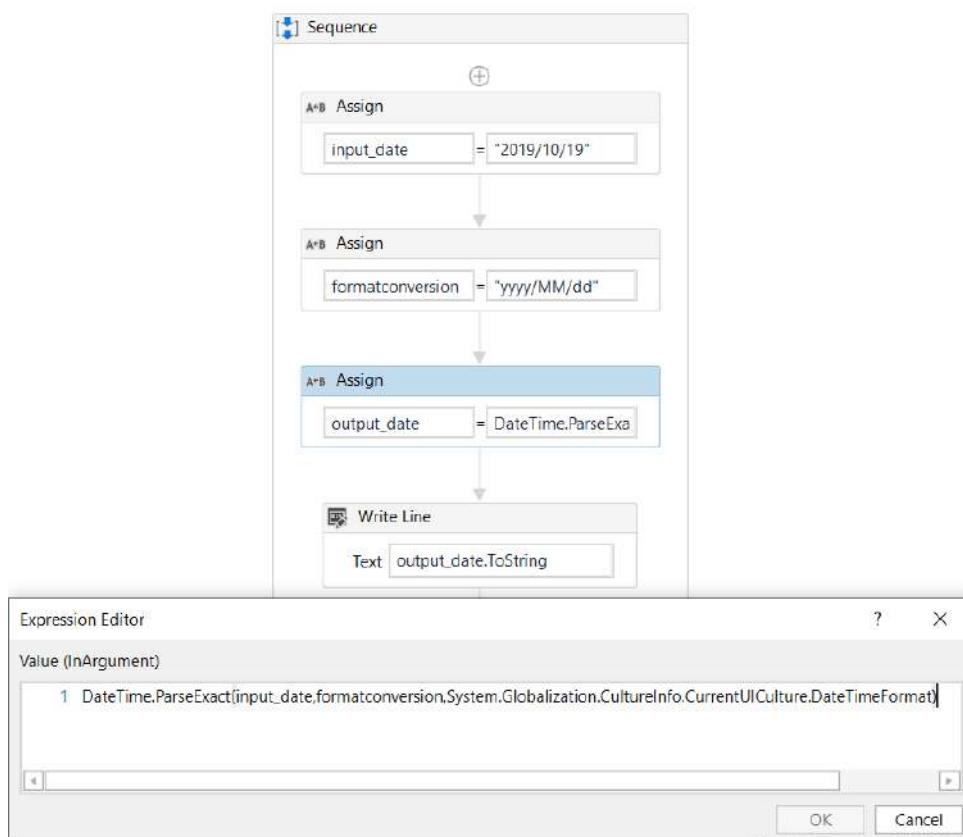
Use the “Write Line” activity to display the result.

Example:

If your input is like 19/09/2019 your format should be dd/MM/yyyy...

If your input is like 2019/09/19 then your format should be yyyy/MM/dd...

`output_date=DateTime.ParseExact(input_date,formatconversion,System.Globalization.CultureInfo.CurrentCulture.DateTimeFormat)`



Output:

10/19/2019 00:00:00

6.7.2. Converting Date from one String Format to Another Desired Format

Let us implement a workflow that takes a Date string and converts it to another format.

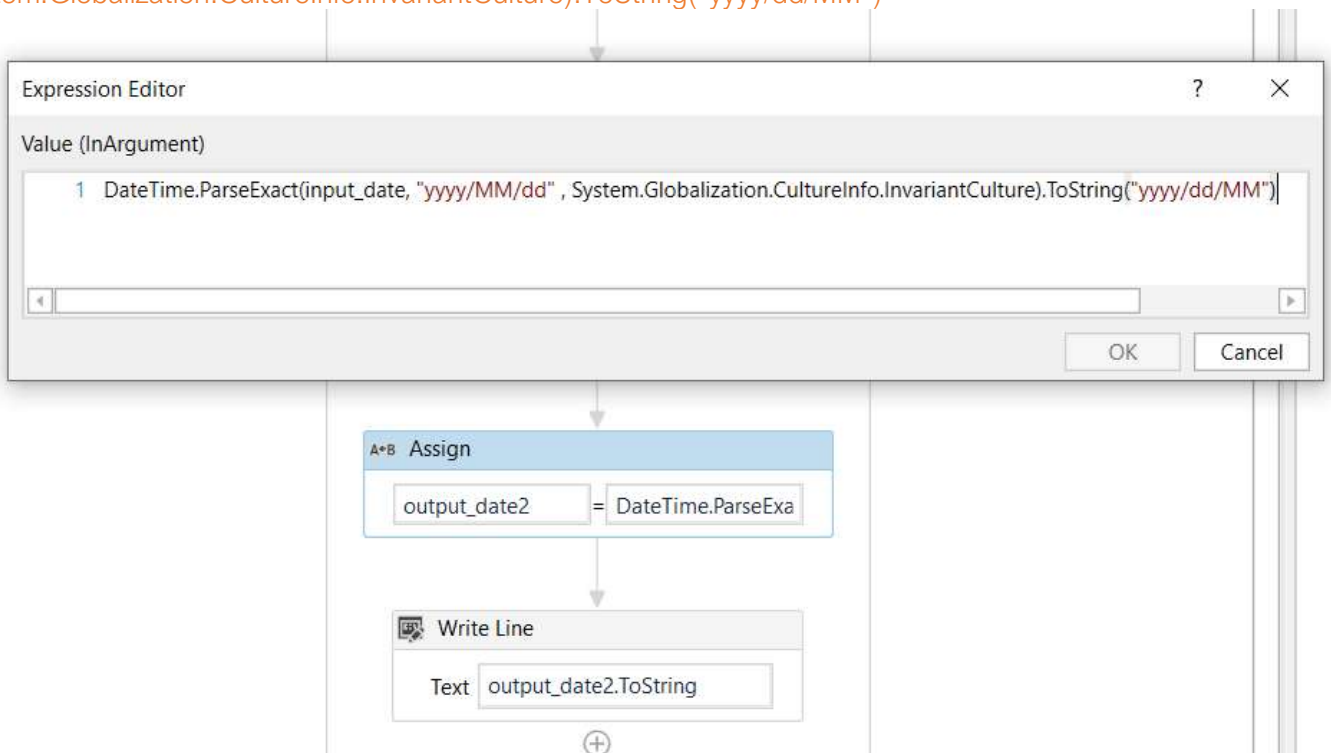
Example:

If your input is like 19/09/2019 i.e.dd/MM/yyyy format, then you can convert it to another format like yyyy/dd/MM

Step 2: Drag “Assign” activities into the design panel and assign the string value and the format of DateTime.

Use the “Write Line” activity to display the result.

```
output_date1=DateTime.ParseExact(input_date, "yyyy/MM/dd",  
System.Globalization.CultureInfo.InvariantCulture).ToString("yyyy/dd/MM")
```



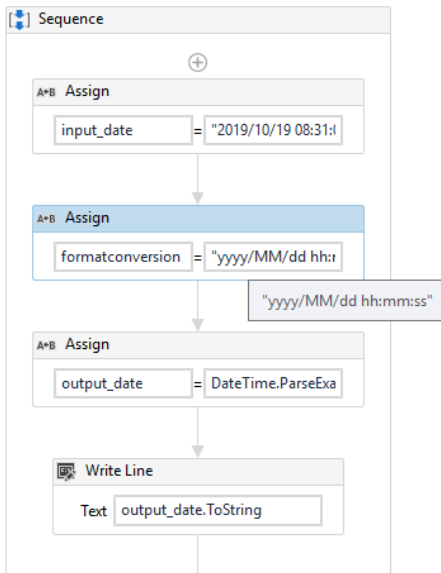
Output:

2019/19/10

6.7.3 Adding Hours, Minutes and Seconds to Your Date

You can have the time along with your date
Just add *hh:mm:ss* to the above formatconversion variable...

For 12 hour format, it is hh:mm:ss
For 24 hour format, it is HH:mm:ss

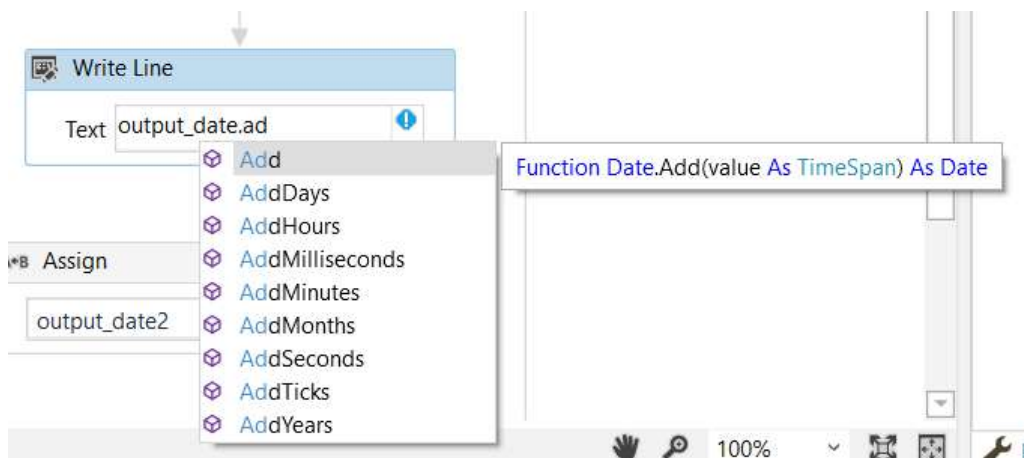


Output:

10/19/2019 08:31:00

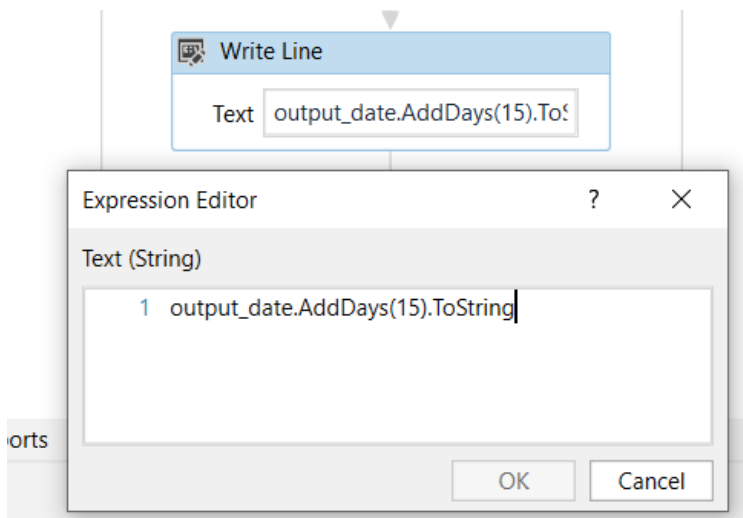
6.7.4. Adding or removing days from your respective date

You can add days / hours / seconds etc. to a DateTime variable like so:



For example, to add 14 days to the output_date 10/19/2019 we use this code:

```
output_date.AddDays(15).ToString
```

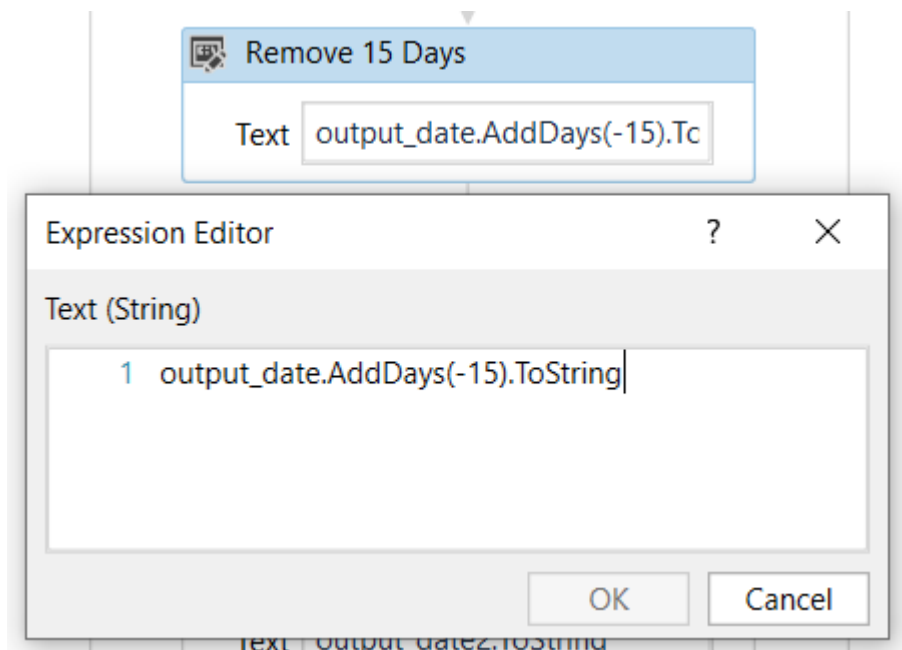


Output:

11/03/2019 00:00:00

To remove days simply add a negative sign like this to get a previous date:

output_date.AddDays(-15).ToString



Output:

10/04/2019 00:00:00

Click the link below to download the example project:

https://drive.google.com/drive/folders/1VJ7B4utKHw_2a-98i4qldFfJqxs4lob?usp=sharing

7 Invoke Code

As processes become more complex, we might need to write our own code. More customization is possible in UiPath by using either VB.NET or also C#. We have an activity to achieve the above, let's see how to use it...

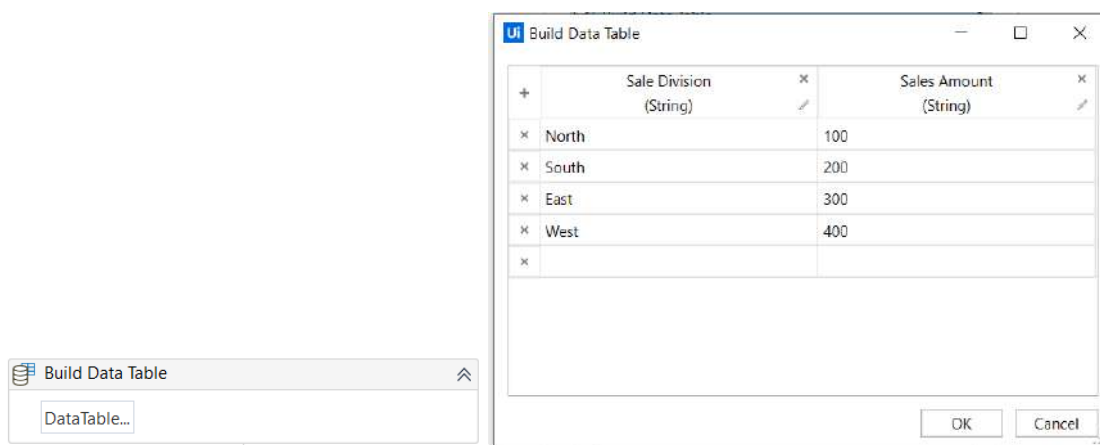
7.1 How to Sum a Column Using Invoke Code

Let's use the Invoke Code activity to calculate the sum of all values in a column.

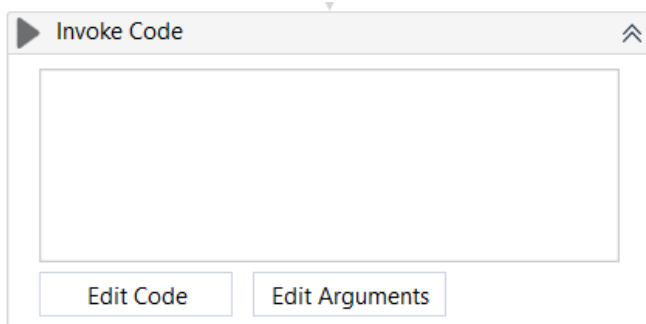
Implementation with UiPath

Let us implement a workflow which takes a sample data table as input and outputs the *SUM of total sales*.

Step 1: Drag a “Build Data Table” activity into the design panel and populate it with sample data.



Step 2: Drag a “Invoke Code” activity into the design panel. Ensure the language in the properties panel is set to VBNet.



Step 3: In the “Invoke Code” activity, click ‘Edit Arguments’ to pass the required parameters as below.

Note:

Sample_DT is our DataTable (IN Argument)

Sum is the summation result (OUT Argument)

Invoked code arguments

Name	Direction	Type	Value
Sample_Dt	In	DataTable	dt_Sample
Sum	Out	Int32	Sum

Create Argument

OK Cancel

Step 4: In the “Invoke Code” activity, click ‘Edit Code’ to write the execution code.

Note:

The below code is used to count the total sales of all the divisions.

```
Dim row As DataRow
sum=0
For Each row In Sample_Dt.Rows
sum= clnt(row("Sales Amount").toString)+Sum
Next row
```

Code Editor

VB.Net Code

```
1 Dim row As DataRow
2 sum=0
3 For Each row In Sample_Dt.Rows
4     sum= clnt(row("Sales Amount").toString)+Sum
5 Next row
```

OK Cancel

Code explanation:

Dim row As DataRow creates a row variable of data type data row

sum=0 assigns a variable sum to the value of 0

For Each row In Sample_Dt.Rows loops through each row in the data table (this is like a For Each Row activity)

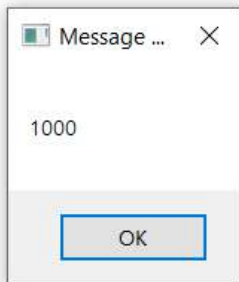
sum= clnt(row("Sales Amount").toString)+Sum adds the “Sales Amount” column value to the sum variable

Next row continues to next row until all rows have been looped

Step 5: Drag a “*Message Box*” activity into the design panel and pass the “Sum” variable into it.



Step 6: Finally, we run the workflow. Now, we can see total sales value being displayed.



Click the link below to download the example project:

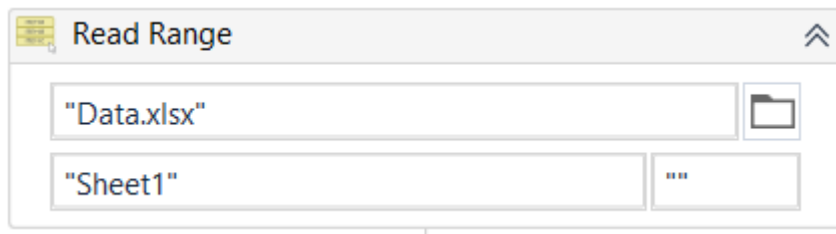
<https://drive.google.com/drive/folders/1SOopLrKpkcXUuK-vTxnd9qT92pomiG76?usp=sharing>

7.2 How to Process and Add a Column Using Invoke Code

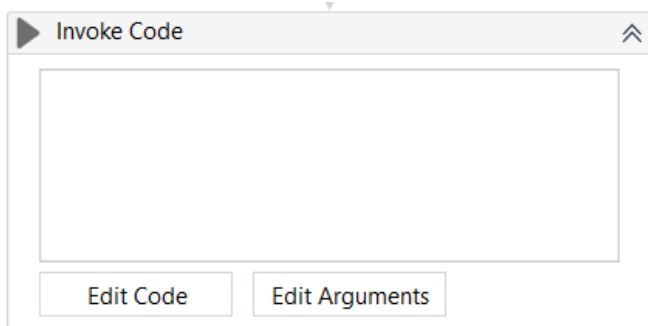
Let's use the Invoke Code activity to calculate the percentage of a column and add it in a new column.

Implementation with UiPath

Step 1: Drag in a *“Read Range”* to get the data from our excel worksheet into a DataTable. Set the Add Headers property to True.



Step 2: Drag a *“Invoke Code”* activity into the design panel. Ensure the language in the properties panel is set to VBNet.

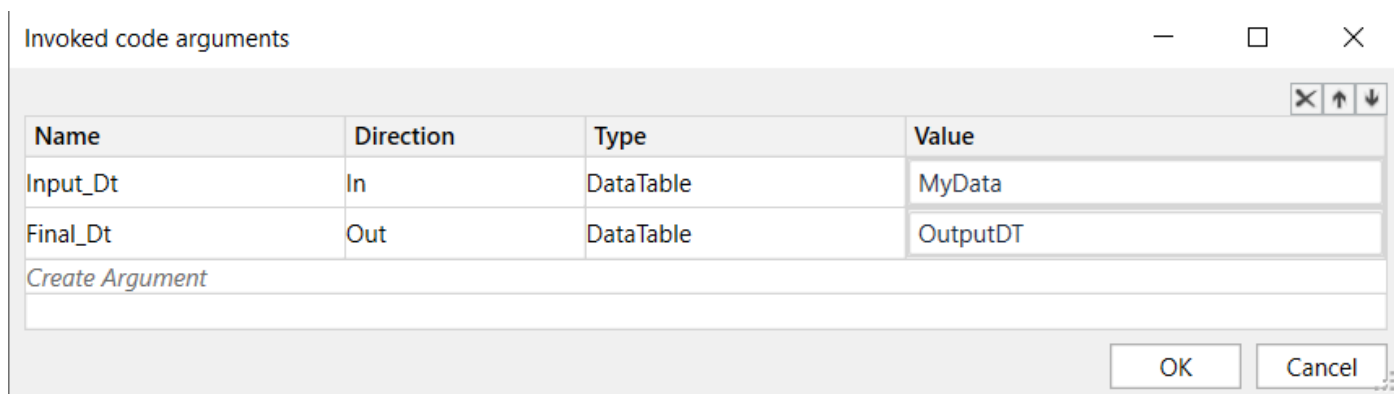


Step 3: In the *“Invoke Code”* activity, click *“Edit Arguments”* to pass the required parameters as below with the appropriate directions.

Note:

Input_Dt is our input DataTable (In Argument)

Final_Dt is our final output DataTable with the added column (Out Argument)

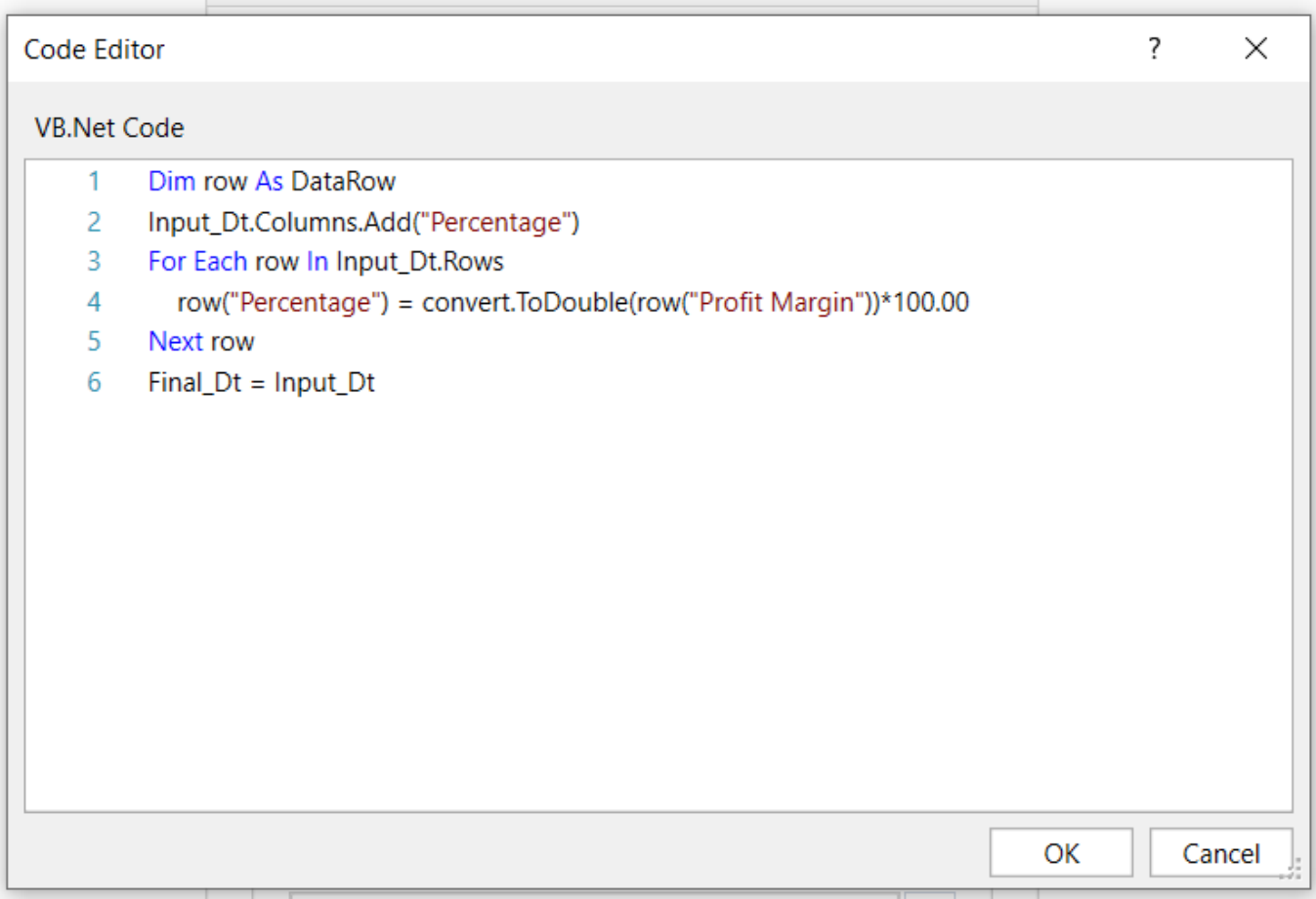


Step 4: In the “*Invoke Code*” activity, click “Edit Code” to write the execution code.

Note:

Below code is used to multiply the Profit Margin column by 100

```
Dim row As DataRow
Input_Dt.Columns.Add("Percentage")
For Each row In Input_Dt.Rows
    row("Percentage") = convert.ToDouble(row("Profit Margin"))*100.00
Next row
Final_Dt = Input_Dt
```



Code explanation:

`Dim row As DataRow` creates a row variable of data type data row

`Input_Dt.Columns.Add("Percentage")` creates a new column with the name “Percentage”

`For Each row In Sample_Dt.Rows` loops through each row in the data table (this is like a For Each Row activity)

`row("Percentage") = convert.ToDouble(row("Profit Margin"))*100.00` multiplies the Profit Margin column by 100 and adds it to the corresponding row in the Percentage column

`Next row` continues to next row until all rows have been looped

`Final_Dt = Input_Dt` Make the Final DataTable equal to the Input DataTable to output it (we could have also set the Input_DT direction in ‘Edit Arguments’ to In/Out and then there would be no need for an additional variable)

Step 5: Use a “Write Range” activity to output the final data table. Set the ‘Add Headers’ property to True.



Write Range

File: Data.xlsx

Sheet: Sheet1

Range: A1

OutputDT:

Step 6: Run the workflow...

Input:

	A	B	C	D	E	F	G
1	Profit	Sales	Profit Margin				
2	234	6543	0,035763411				
3	123	6456	0,019052045				
4	654	5724	0,114255765				
5	232	4723	0,049121321				
6	532	5678	0,093694963				
7	532	5678	0,093694963				
8	532	5678	0,093694963				
9							
10							
11							
12							
13							
14							
15							
16							

Output:

	A	B	C	D	E	F	G
1	Profit	Sales	Profit Margin	Percentage			
2	234	6543	0,035763411	3.57634112792297			
3	123	6456	0,019052045	1.90520446096654			
4	654	5724	0,114255765	11.4255765199161			
5	232	4723	0,049121321	4.91213211941563			
6	532	5678	0,093694963	9.36949630151462			
7	532	5678	0,093694963	9.36949630151462			
8	532	5678	0,093694963	9.36949630151462			
9							
10							
11							
12							
13							
14							
15							
16							

Click the link below to download the example project:

https://drive.google.com/drive/folders/1HY2bhwgHyoU_sJ_lEaQ6jhGP0-mb7efE?usp=sharing

8 Conclusion

I hope this eBook was helpful, you can use it as a reference whenever you need to use VB.NET code or scripts in your automations.

8.1 Where to Look for VB.NET Code if You Can't Find it in this Guide?

This guide covers the most common use cases of VB.NET in UiPath. However, it does not cover everything that VB.NET is capable of. These are the best sites to search for a solution if you can't find it in this guide, and in this order:

1. UiPath Forum: <https://forum.uipath.com/>
2. Stack Overflow: <https://stackoverflow.com/>
3. A general Google search: <https://www.google.com/>
4. Official .NET documentation: <https://docs.microsoft.com/en-us/dotnet>
5. Still can't find what you're looking for? Ask here
<https://www.facebook.com/groups/2481744772043143/>

8.2 How to Take Your UiPath RPA Skills to an Advanced Level?

If you would like to take your UiPath RPA skills to an advanced level I have an advanced online training called the *Complete RPA Bootcamp*. Watch my free RPA training / webinar below to learn more:

<https://www.completepabootcamp.com/rpatraining>

Or enroll in the *Complete RPA Bootcamp* now with a special 75% off:

<https://www.completepabootcamp.com/special>

Copyright © 2020 by Leon Petrou, Inc.

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other non-commercial uses permitted by copyright law. For permission requests, write to the publisher, with subject line "Attention: Permissions Coordinator" at the address support@completepabootcamp.com. This eBook is not associated with the UiPath website or UiPath Inc. Additionally, this site is NOT endorsed by UiPath in any way. UiPath is a trademark of UiPath, Inc. Disclaimer: The contents of this eBook are for information purposes only. Leon Petrou, Inc. and the author Leon Petrou take no responsibility for any losses financially or other occurred due to the information provided in this book due to incorrect information, incomplete information or any other reason. Some content of this eBook has been reproduced from content on the UiPath Forum, Stack Overflow and other websites specifically Excel Cult authors include Sumanth Veerali and Sharath Raju. Credit is given to those respective authors.

Business Address: Leon Petrou, Inc, 651 N. Broad St.Suite 206, Middletown, DE 19709, United States