Practical Exercise - Walkthrough

Since there are two workflow parts, organize the project as a Flowchart.

A. First create a While loop that the entire automation will be contained in. This will make sure all sheets are indexed through until the end.

- Find and add a **While** activity to the main container
- Create an int32 variable called **Index**

o Set its default value to 1

- Create an int32 variable called **N**

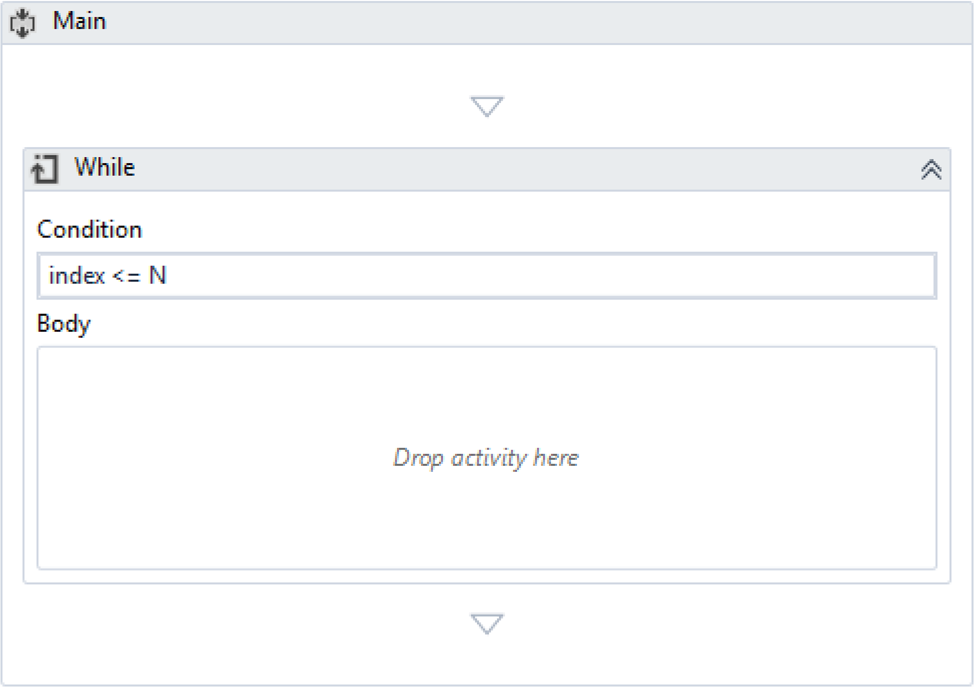
o Set its default value to 3 (it can be changed depending on the number of sheets)

- In the properties of the **While** activity:

o In the condition parameter, write: index <= N

▪ This makes sure that as long as the index is less than what **N** is set to, the automation will keep aggregating sheets

- This is what it should look like:



Condition

index <= N

Body

Next, activities need to be added that will read the sheet in each iteration and add them to the main sheet.

- Find and add a **Read Range** activity into the body of the **While** activity

o Set the workbook path to the full path of the sample Excel workbook (downloaded in the outline - previous material)

o Set the sheet name to: "Sheet" + index.ToString

▪ This assumes the workbook sheets will be named Sheet1, Sheet2, …, etc.

o In the output parameter, use the Ctrl-K shortcut to create a DataTable called **partialTable**

If we want the consolidated sheet to have the data from all the sheets, means we have to treat the addition of the first sheet data separately, to add the whole tale, together with the headers. The other sheets data only has to be appended at the end, without adding column headers

- Find and add an **If** activity below the **Read Range** activity

o Set the condition parameter to: Index = 1

- Find and add a **Write Range** activity to the 'Then' branch of the **If** activity

o Set the workbook path to "**Consolidated.xlsx**" - this will create a new workbook with the respective name in the project folder, if it doesn't already exist.

o Set the DataTable input to **partialTable** o This branch of the **If** will create a workbook with the first sheet as a base

- Find and add an **Append Range** activity to the 'Else' branch of the **If** activity

o Set the workbook path and DataTable input to the same values as the **Write Range** activity

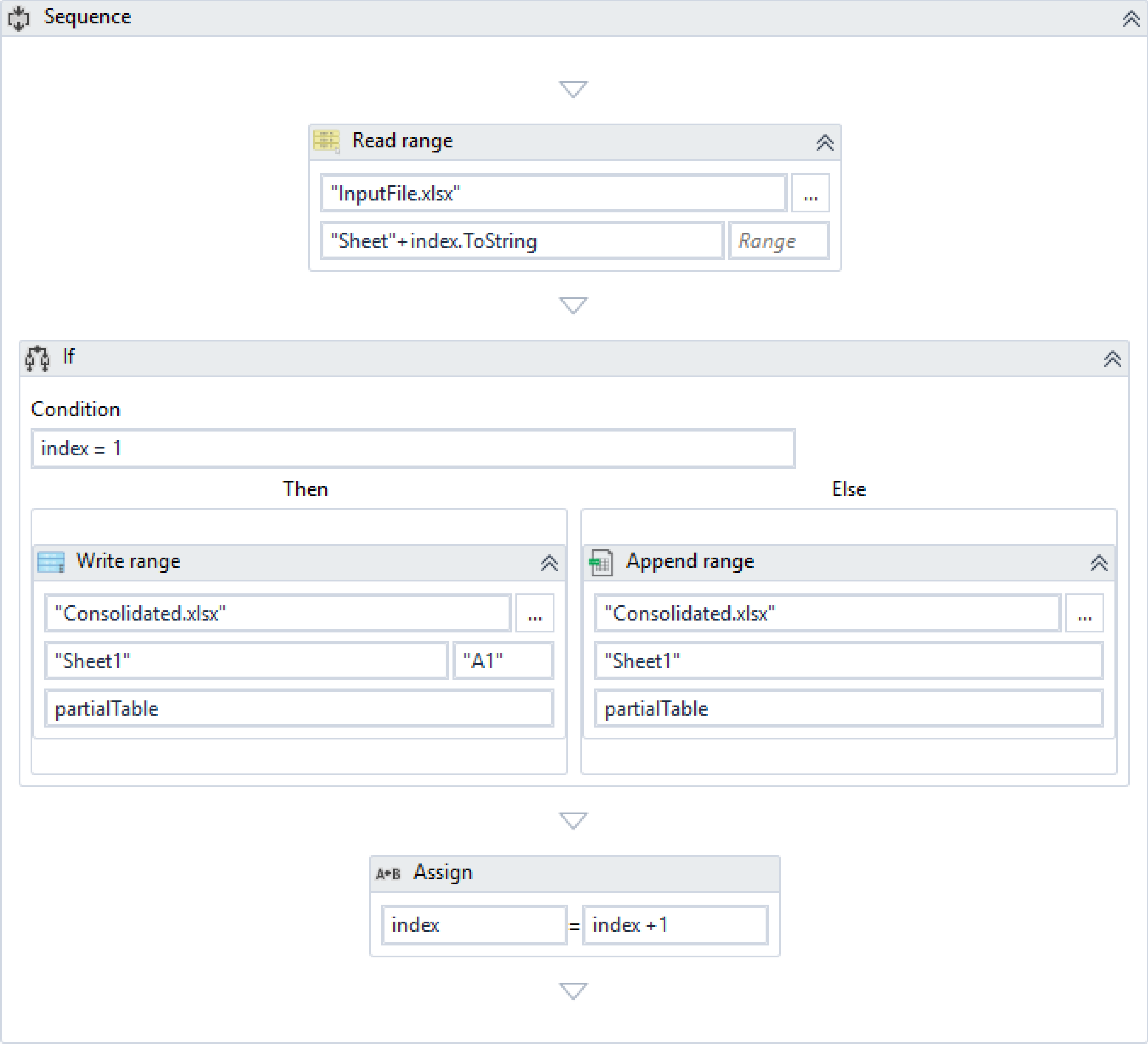
o This will append the ranged that is read from each of the next sheets to **Consolidated.xlsx**

- Finally, add an **Assign** activity below the **If** activity

o Assign **index + 1** to **index**

o This will increment the loop counter so the sheets will progress from 1 to 2 to 3 and so on…

- This is what the workflow should look like inside the body of the **While** activity:



Sequence

Read range
"InputFile.xlsx"
"Sheet"+index.ToString
Range

If
Condition
index = 1
Then
Else

Write range
"Consolidated.xlsx"
"Sheet1"
"A1"
partialTable

Append range
"Consolidated.xlsx"
"Sheet1"
partialTable

Assign
index = index + 1

B.Create a new sequence within the flowchart to filter the data.

- Find and add an **Excel Application Scope** activity to the new container and select the file you created in the previous step, Consolidated.xlsx

- Use a **Read Range** activity to extract the workbook data into a datatable, consolidatedDT.

- Use a **Filter Data Table** activity. Click on the **Filter Wizard**, add **consolidatedDT** as input and output, and select **remove rows**, because we want to get rid of the rows where the ID value is of less than 10.

    - Add a condition for “ID” < 10.0

- Add an **Output Data Table** activity to the new container, and then add **consolidatedDT** as the input data table. Create a new variable, **stringDT**, for the output text.

- Add a **Message Box** and use **stringDT** as the content.