Week 10 : Workbook and Excel Automation

- ตั้งชื่อ Project ด้วย week10_รหัสนักศึกษา เช่น week10_62070001
- ในแต่ละข้อ สร้าง sequence flow ด้วย เลขที่ข้อ_รหัสนักศึกษา เช่น 10.1_62070001
- บันทึก flow แต่ะละข้อให้เห็นชัดเจนพร้อมกับ run ผลลัพธ์ แสดงผลลัพธ์ในแต่ละข้อ และบันทึกคลิป ตามรูปแบบ เช่น week10_62070001.mp4 (ใช้โปรแกรมบันทึกหน้าจอ)
- ส่งงานทุกข้อเป็น 1 Project รวมใน 1 folder ที่มี input, output ทั้งหมดในโปรเจคนั้นๆ ตั้งชื่อไฟล์ week10_รหัสนักศึกษา.zip เช่น week10_62070001.zip

Exercise 10.1 : Print data from a workbook

Objective: Demonstrate how to print data from a workbook in the output panel using Read Range activity.

- Read the data using Read Range activity from excel file.
- Loop through each data and subtract it with current year to get year of birth.
- Display the result in the Output panel

Learning Outcomes

After completion of this exercise you will get familiar with the following:

✓ **"Sequence"** activity.

✓ **"Read Range"** activity.

✓ **"For Each Row**" activity.

✓ **"Output data table"** activity.

✓ **"Write Line"** activity.

Algorithm:

Step 1: START

Step 2: Add the Sequence in the main area and add **Read Range** activity under Workbook category in the Activities panel and insert in the designer panel.
Step 3: Select the excel file "ageStudent.xlsx"

Step 4: Go the Properties panel of Read Range activity and in the **DataTable property**, enter a new variable called **ageStudent**.

Step 5: Insert **For Each Row** activity below the Read Range activity. In the VB expression text box enter **ageStudent**.

Step 6: Insert **Assign activity** in the Body section of For Each Row activity. In the first text box enter **row(1)** and in the adjacent text box enter the expression: **Date.Today.Year() - CInt(row(0)) (***calculate year of birth)**

Step 7: Insert **Output Data Table** activity after For Each activity.

Step 8: Go to its Properties panel and **enter ageStudent in the DataTable** property.

Step 9: In the Output property, enter a new variable called **ageOutput.**

Step 10: Insert **Write Line** activity below Output Data Table activity. Enter ageOutput in the text box.

Step 11: Save and run the workflow.

Step 12: Go to the Output panel to see the output

Outcome:

In output pane, year of birth of all students are listed adjacent to their respective age.

Example: 25, 1995

Exercise 10.2 : Read data from a workbook and append data to another workbook.

Objective: Build a workflow using Read Range and Append Range activity to read data from a workbook and append data to another workbook.

- Read the data from the excel file using Read Range activity.
- Convert all city names in capital letters.
- Add the updated names in a new spreadsheet using Append Range activity.

Learning Outcomes

After completion of this exercise you will get familiar with the following:

✓ **"Sequence"** activity.

✓ **"Read Range"** activity.

✓ **"Add Data Column**" activity.

✓ **"For Each Row**" activity.

✓ **"Append Range"** activity.

Algorithm:

Step 1: START

Step 2: Use **Read Rang**e activity from Workbook category and to read data from the excel file path "**CityNames.xlsx**" and store in a data table variable.

Step 3: Use **Add Data Column** activity add a new column in the data table called "**City in Capitals**".

Step 4: Use **For Each Row** activity to iterate through each row items in the data table.

Step 5: Use **Assign** activity to convert each row item to **uppercase (Hint: row(0).ToString.ToUpper)**

Step 6: Use **Append Range** activity below For Each activity to store updated data tables with city names in capitals in a new excel file **"CityInCapital.xlsx".**

Step 7: STOP

Outcome: "cityInCapital.xlsx" file showing all city names in capital letters.

Exercise 10.3 : Calculation in Excel automation

Objective : Build a workflow that calculates total monthly deposit of a bank from an Excel file and store output in a new sheet.

- File contains three deposit categories – Cash In, On-Us Check, and Not On-Us Check.
- Calculate total amount received in all three categories for the month of June.
- Store calculated values in a **new sheet** in the **same excel** file.

Learning Outcomes

- After completion of this exercise you will get familiar with the following:

✓ **"Sequence"** activity.

✓ **"Build Data Table"** activity.

✓ **"Excel Application Scope"** activity.

✓ **"Read Range"** activity.

✓ **"For Each Row**" activity.

✓ **"Add Data Row**" activity.

✓ **"Write Range"** activity.

Algorithm

Step 1: START

Step 2: Use **Build Data Table** activity and in its Properties panel, enter a new variable **sumTable**.

Step 3: Create a data table with **three integer columns** with names "Cash In Total", "On-Us Check Total", and "Not On-Us Total".

Step 4: Use **Excel Application Scope** activity and select the excel file "**excel_report.xlsx**"

Step 5: Use **Read Range** activity in the Do container to read excel data from "**June Report**" sheet.

Step 6: Use an **Assign activity** within a **For Each Row** activity below the Read Range activity

- In the first text box, enter new integer variable **sum1**.
- In the second text box, enter expression **CInt(row (1)) + sum1**.

Step 7: Use **second Assign** activity in the Body section

- In the first text box, enter new integer variable **sum2**.
- In the second text box, enter expression CInt(row (2)) + sum2.

Step 8: Use **third Assign** activity in the Body section.

- In the first text box, enter new integer variable **sum3**.
- In the second text box, enter expression **CInt(row (3)) + sum3**.

Step 9: Use **Add Data Row** activity below For Each Row activity. Go to its Properties panel, and in the ArrayRow property enter the expression**: {sum1, sum2, sum3}**. In the Data Table property enter sumTable.

Step 10: Use **Write Range** activity below Add Data Row activity,

- Insert "**June Total**" in the first text box, "**sumTable**" in the second text box.
- Check the box of **Add Headers** property in the Properties panel

Step 11: STOP

Outcome: "excel_report.xlsx" file having a new "June Total" sheet with calculated values.

Exercise 10.4 : Compare Value in Excel automation

Objective : Demonstrate how to read and write in Excel by comparing first two columns and inserting result in third column using Excel activities.

- Create an excel file containing ten random numbers between 1 to 100 in two columns
- Read file and transport into data table, and add a third column
    - If value in first column is greater than second column, enter "Greater" in third column as result.
    - If value in first column is less than second column, enter "Lesser" in third column as result
    - If value in first column is equal to second column, enter "Equal" in third column as result
- Write back the updated data table in the same excel file in a new sheet

Learning Outcomes

- After completion of this exercise you will get familiar with the following:

✓ **"Sequence"** activity.

✓ **"Excel Application Scope"** activity.

✓ **"Read Range"** activity.

✓ **"Add Data Column**" activity.

✓ **"For Each Row**" activity.

✓ **"If**" activity.

✓ **"Append Range"** activity.

Algorithm

Step 1: START

Step 2: Insert **Excel Application Scope** activity in the designer panel. Click the ellipsis icon and select the excel file "**compareValue.xlsx**"

Step 3: Insert **Read Range** activity within the Do container of Excel Application Scope activity.

Step 4: Go to the Properties panel of Read Range activity, and in the Output property, enter a new variable called **dummyData**.

Step 5: Insert **Add Data Column** activity below Read Range activity. Go to the Properties panel. In the ColumnName property, enter "**Comparison**", and in the DataTable property enter **dummyData**.

Step 6: Insert **For Each Row** activity below Add Data Column activity. In the *Vb expression* text box enter **dummyData**.

Step 7: Insert **If** activity in the **Body** section of **For Each Row** activity. Enter condition **CInt(row(0))>CInt(row(1))**.

Step 8: In the **Then** section, insert an **Assign** activity. In the first text box enter row(2) and in the adjacent text box enter the string "Greater".

Step 9: Insert another **If** activity in the **Else** section of first **If** activity. Enter condition **CInt(row(0))<CInt(row(1))**.

Step 10: In the **Then** section, insert an **Assign** activity. In the first text box enter **row(2)** and in the adjacent box enter the string "Lesser".

Step 11: In the **Else** section enter another **Assign** activity. In the first text box enter **row(2)** and in the adjacent box enter the string "Equal".

Step 12: Insert **Append Range** activity below For Each Row activity. In the first text box enter "Sheet2", and in the second text box enter dummyData. It will insert the result in a new sheet in the same excel sheet.

Step 13: Save and run the workflow.

Step 14: Open the excel file to check the outcome

Outcome:

The comparison result is stored in third column in a new sheet2.