LECTURE 08. EXCEL AND DATA TABLES

Robotic Process Automation
[19 November 2019]

Elective Course, 2019-2020, Fall Semester

Camelia Chisăliţă-Creţu, Lecturer PhD
Babeş-Bolyai University

Acknowledgements

This course is presented to our Faculty with the support of UiPath Romania.

Contents

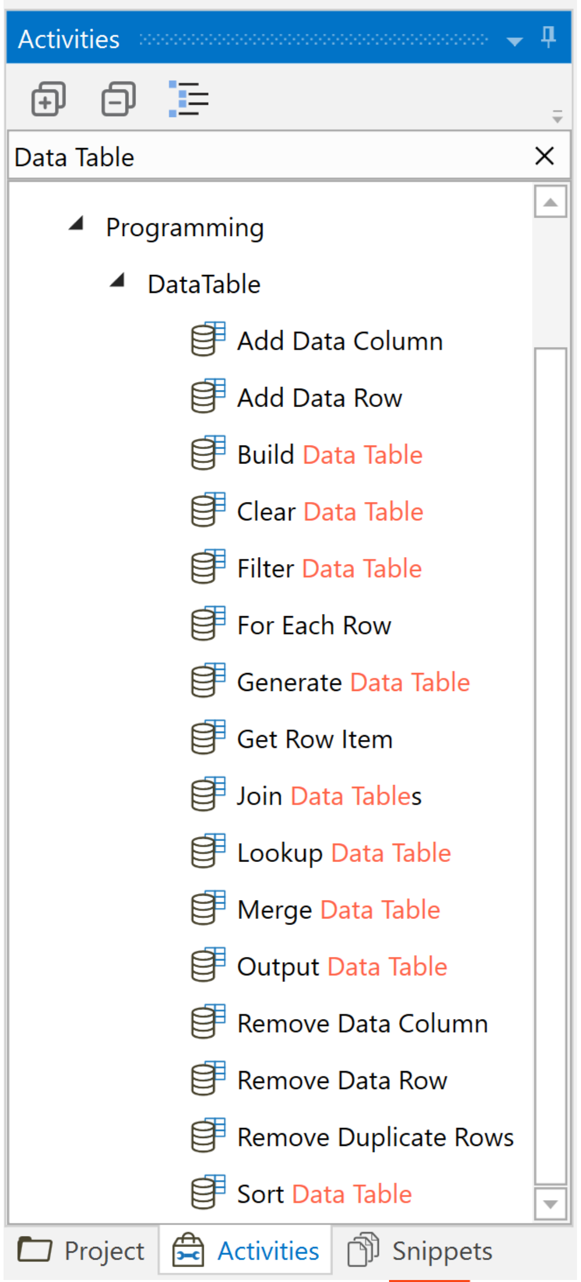
Data Table. Details

- **Data Table** variable type characteristics in UiPath:
  - **a data structure with flexible length;**
  - it is used to store data in the form of rows and columns, similar to Excel;
  - it enables:
    - to migrate data from one database to another;
    - to process data (filter, create new data, etc.);
  - it can be iterated by using a **For Each Row** activity;
  - ways to create a data table:
    - web **data scrapping** (see **Lecture 05. UI Interaction**);
    - building it from scratch:
      - **Build Data Table**, **Add Row**, **Add Data Column** activities;
    - processing data from **Excel sheets** and **.csv** files:
      - various activities: **Excel Application Scope, Read/Write Range, Read/Write Cell, Output Data Table, Select, Filter, etc;**

Row/ Column	First	Last	Club Member
0	"John"	"Doe"	Yes
1	"Jane"	"Doe"	No
2	"Jane"	"Doe"	Yes
3	"John"	"Doe"	No

Data Table. UiPath Activities

- UiPath provides a series of activities for **Data Table** variables;
- these activities are found under the **Programming** library, **Data Table** section.

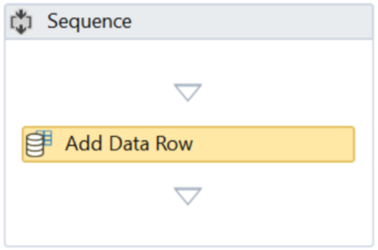

Activities

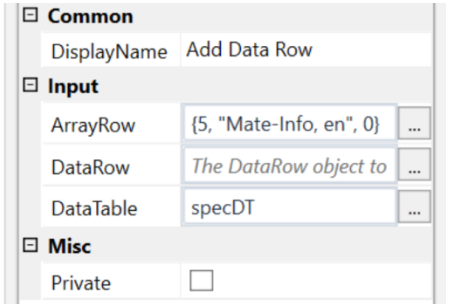
Data Table

- Programming
  - DataTable
    - Add Data Column
    - Add Data Row
    - Build Data Table
    - Clear Data Table
    - Filter Data Table
    - For Each Row
    - Generate Data Table
    - Get Row Item
    - Join Data Tables
    - Lookup Data Table
    - Merge Data Table
    - Output Data Table
    - Remove Data Column
    - Remove Data Row
    - Remove Duplicate Rows
    - Sort Data Table

Add Data Row Activity. Details

- **Add Data Row** activity
  - allows to add a new data row into a data table;
- relevant properties:
  - **[input] DataTable** = **DataTable** variable
  - **[input] Array Row** = **Array** variable
    - an array consisting of the values that will be added to the data table;
    - each value within the array has the data type that corresponds to the column in the data table;
      - E.g.: {5, "Mate-Info, en", 0} or {(SpecDataTable.Rows.Count+1).ToString,name};
  - **[input] Data Row** = **DataRow** variable
    - if such object is provided then the **Array Row** property is ignored.

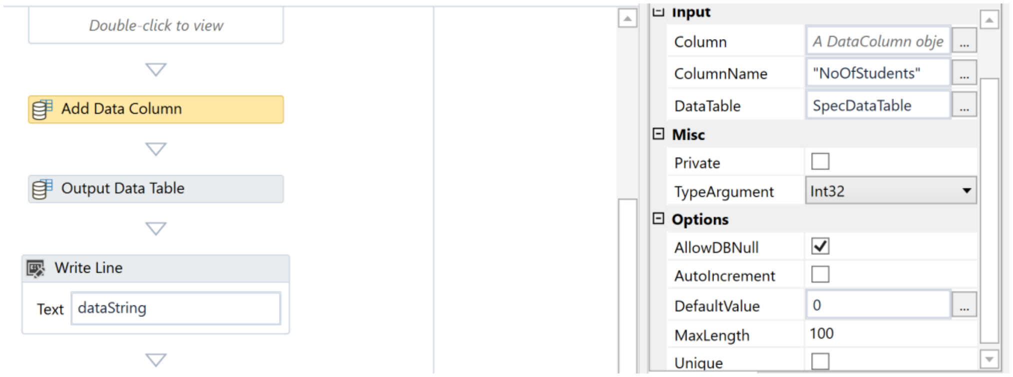




Common	
DisplayName	Add Data Row
Input	
ArrayRow	{5, "Mate-Info, en", 0}
DataRow	*The DataRow object to*
DataTable	specDT
Misc	
Private	☐

Add Data Column Activity. Details

- **Add Data Column** activity
  - allows to change the structure of the data table by adding a new column;
- relevant properties:
  - **[input] DataTable** = **DataTable** variable
  - **[input] Column Name** = **String**
    - E.g.: “NoOfStudents”
  - **[input] TypeArgument** = **Type** variable
    - the data type for the new column that is added into the data table;
  - **[options] AllowDBNull, AutoIncrement, DefaultValue, MaxLength, Unique**
    - attributes of the new column that can be set.

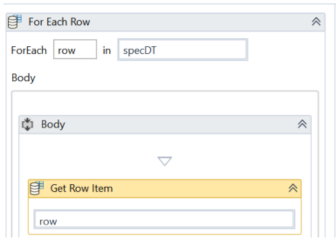


Text dataString

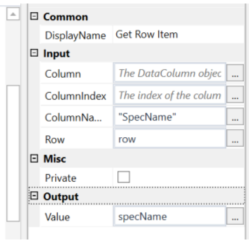
Input	
Column	A DataColumn obje
ColumnName	"NoOfStudents"
DataTable	SpecDataTable
Misc	
Private	☐
TypeArgument	Int32
Options	
AllowDBNull	✓
AutoIncrement	☐
DefaultValue	0
MaxLength	100
Unique	☐

Get Row Item Activity. Details

- **Get Row Item** activity
  - allows to get the value of a column from a specific row in the data table;
  - is usually placed in a **For Each Row** activity that allows to iterate over a data table;
- relevant properties:
  - **[input] Row** = **DataRow** variable
    - a variable used to iterate the data table;
  - **identification criterion (one option can be chosen)**
    - **[input] Column** or **[input] ColumnIndex** = 1 or
      - the index of the column whose value is extracted;
      - the first column in the data table has the index 0;
    - **[input] ColumnName** =“SpecName”
      - the name of the column used as sorting criterion;
      - it is the preferred identification criterion compared to **Index**;
  - **[output] Value** = **GenericValue** variable.



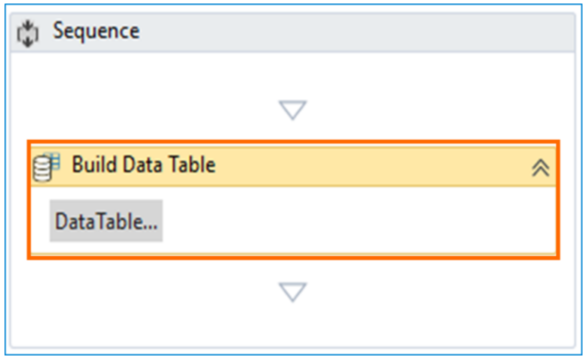
For Each Row
ForEach row in specDT
Body
Body
Get Row Item
row

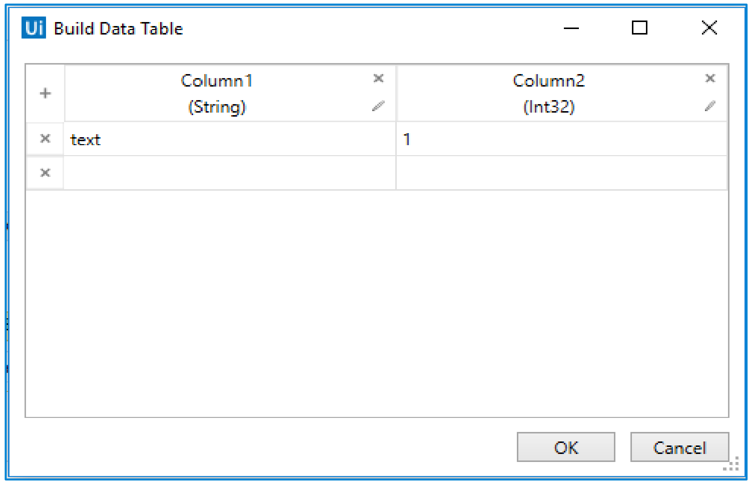


Common	
DisplayName	Get Row Item
Input	
Column	*The DataColumn objec*
ColumnIndex	*The index of the colum*
ColumnNa...	"SpecName"
Row	row
Misc	
Private	☐
Output	
Value	specName

Build Data Table Activity. Details

- **Build Data Table** activity
  - is used when a user has to store the data manually inside the data table;
  - allows to reorder existing columns;
- Steps to use the activity:
  1. drag and drop **Build Data Table** activity inside the sequence;
  2. click on the **Data Table** to customize the data table;

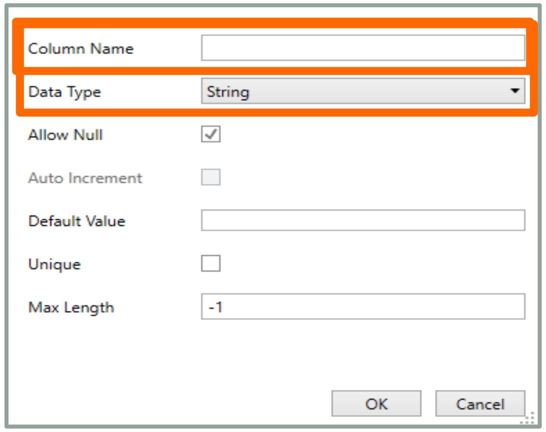




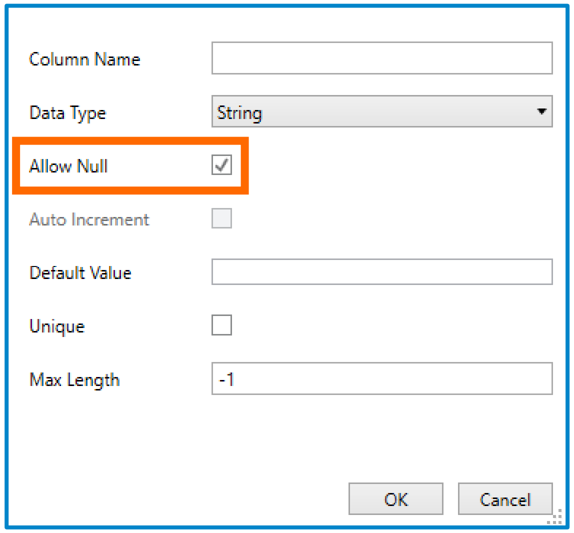
	Column1 (String)	Column2 (Int32)
×	text	1
×		

Build Data Table Activity. Steps (1)

- Steps to use the activity:
  1. drag and drop **Build Data Table** activity inside the sequence;
  2. click on the **Data Table** to customize the data table;
  3. click on the “+” sign in order to add a column button;
     - set the name and data type of the column;
     - if **Allow Null** is checked, it is not compulsory to have additional data in the column;



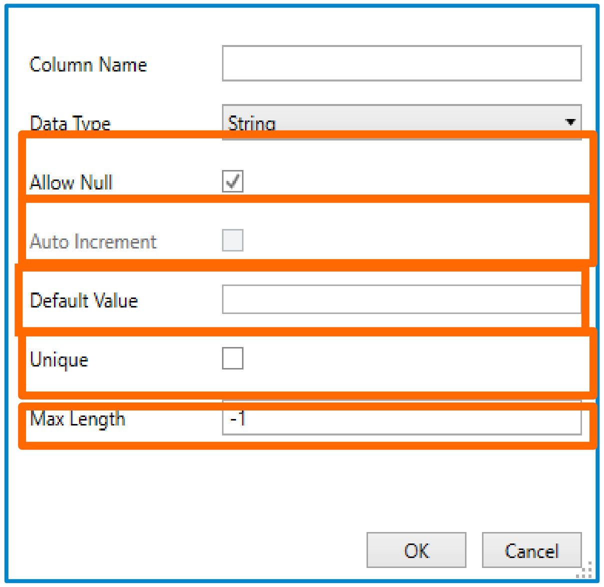
Column Name	
Data Type	String
Allow Null	✓
Auto Increment	
Default Value	
Unique	
Max Length	-1



Column Name	
Data Type	String
Allow Null	✓
Auto Increment	
Default Value	
Unique	
Max Length	-1

Build Data Table Activity. Steps (2)

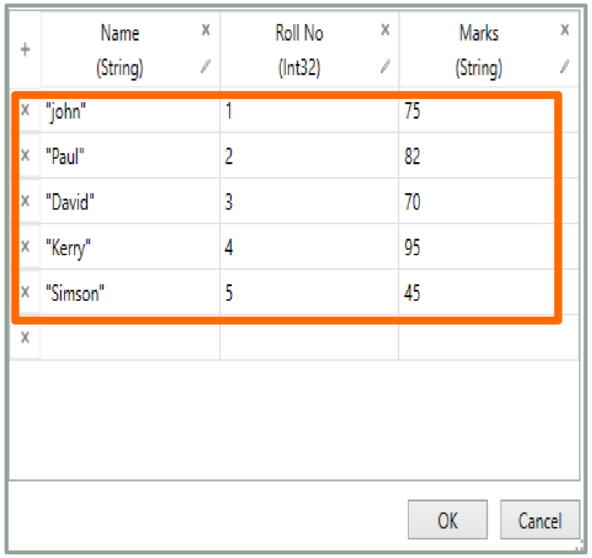
- Steps to use the activity:
  5. Other attribute can be set:
     - **Default value:** if that column is blank, then it will automatically take default value inside it;
     - **Max Length:** The number of characters allowed for the column; if the user does not want to apply the length of maximum data, then the -1 is set; -1 is also a set of the default value;
     - **Unique:** if the dataset is selected for a specific value then the user can create or develop unique data in the data table;
     - **Auto Increment:** when the user sets the **Data Type** in the form to **Int32** then the checkbox is enabled; the data automatically increase by 1 every time a new row is added.



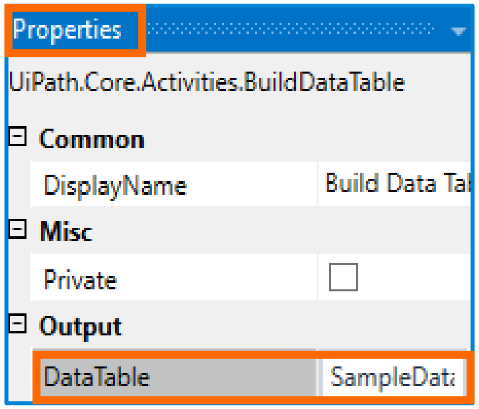
Column Name	
Data Type	String
Allow Null	✓
Auto Increment	☐
Default Value	
Unique	☐
Max Length	-1

Build Data Table Activity. Steps (3)

- Steps to use the activity:
  6. add data inside the data table;
  7. click on the **Properties** tab of the built **Data Table**;
     - create the variable in the **Output** section with the name “Sample Data.”



Name (String)	Roll No (Int32)	Marks (String)
"john"	1	75
"Paul"	2	82
"David"	3	70
"Kerry"	4	95
"Simson"	5	45



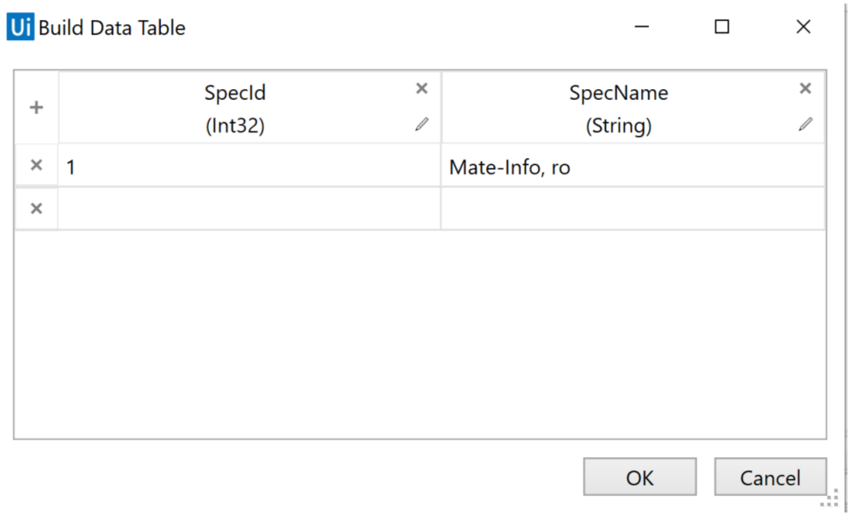
Properties

UiPath.Core.Activities.BuildDataTable

Common	
DisplayName	Build Data Tal
Misc	
Private	☐
Output	
DataTable	SampleDat:

Demo 1. Build Data Table

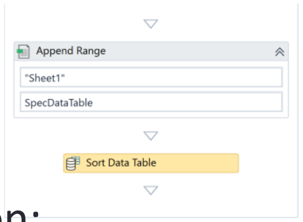
- Use the Build Data Table, Add Data Row and Add Data Column activities to build the data table with the following data:
  - SpecId (int);
  - SpecName (string);
  - NoOfStudents (int);
- the first two data fields are added by using Build Data Table activity, while the third one is added by using Add Data Column activity;
- the values for the first two data fields are read from the standard input, while the values for the third data field is computed (in a subsequent demo).



SpecId (Int32)	SpecName (String)
1	Mate-Info, ro

Sort Data Table Activity. Details

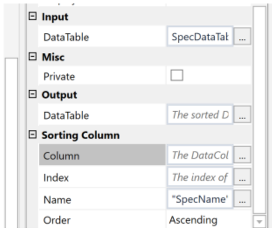
- **Sort Data Table** activity
  - allows to sort the data from a given data table considering some column details;
  - can be placed inside or outside an **Excel Application Scope** activity, as long as the scope of the **DataTable** variables used are set to include the **Sort Data Table** activity;
- relevant properties:
  - **[input] DataTable** = **DataTable** variable *and* **[output] DataTable** = **DataTable** variable
    - the resulting data table after performing sorting;
    - the output can be the input data table;
  - **sorting criterion (one option can be chosen)**
    - **[input] Column** or **[input] Index** = 1 or
      - the index of the column used as sorting criterion;
      - the first column in the data table has the index 0;
    - **[input] Name** =“year”
      - the name of the column used as sorting criterion;
      - it is the preferred sorting criterion compared to **Index**;
  - **[input] Order** = **Ascending/Descending.**


Append Range

"Sheet1"

SpecDataTable

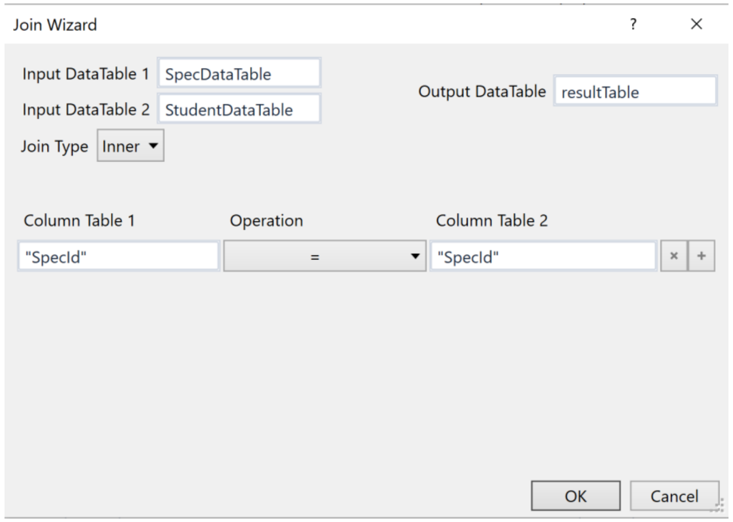
Sort Data Table



Input	
DataTable	SpecDataTal
Misc	
Private	☐
Output	
DataTable	*The sorted D*
Sorting Column	
Column	*The DataCol*
Index	*The index of*
Name	"SpecName'
Order	Ascending

Join Data Table Activity. Details

- [**Application Integration** activities --> **DataTable** section]
- **a join operation** allows
  - to combine data from multiple tables based on certain conditions (usually when field values are equal);
- there are several types of join actions: **left join, inner, full;**
- **Join Data Table** activity
  - allows to build new data starting from multiple data tables;
- relevant properties:
  - **[input] Input DataTable 1** = **DataTable** variable
  - **[input] Input DataTable 2** = **DataTable** variable
  - **[output] Output DataTable** = **DataTable** variable
  - **[input] Type** = **Inner/Left/Full** join
  - **[input] Condition**
    - uses logical operators (and, or).

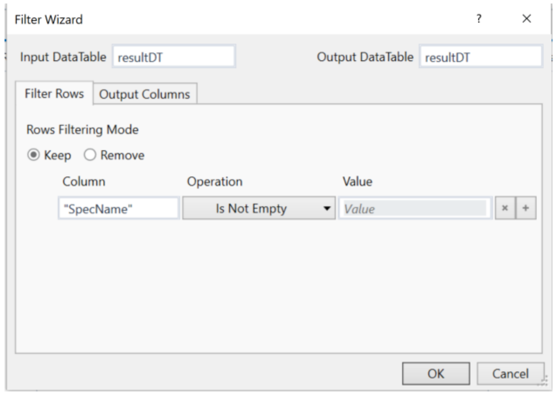


Input DataTable 1 SpecDataTable
Input DataTable 2 StudentDataTable
Output DataTable resultTable
Join Type Inner

Column Table 1	Operation	Column Table 2
"SpecId"	=	"SpecId"

Filter Data Table Activity. Details

- **Filter Data Table** activity
  - provides a wizard that allows to configure the filter actions;
- relevant properties:
  - **[input] DataTable** = **DataTable** variable
  - **[output] DataTable** = **DataTable** variable
    - ihe input and output data table can be the same **DataTable** variable;
  - **[options] Filter Rows, Select Rows**
    - attributes for the filtering actions are st based on the wizard options:
      - **Filter Rows:**
        - allows to indicate the filtering conditions;
        - for numeric columns UiPath convert the value to **Double**;
          - E.g.: “SpecId”, “=“, 3.00 even if the data type of **SpecId** field is **Int32**;
      - **Output Columns:**
        - allows to change the structure of the output data table by instantiating a **DataTable** variable and populating it with data, accordingly.



Filter Wizard

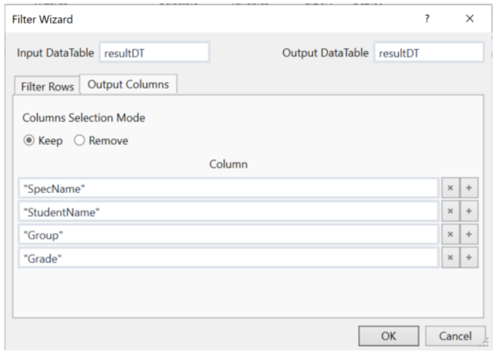
Input DataTable resultDT
Output DataTable resultDT

Filter Rows | Output Columns

Rows Filtering Mode
[✓ Keep] Remove

Column	Operation	Value
"SpecName"	Is Not Empty	Value

OK Cancel



Filter Wizard

Input DataTable resultDT
Output DataTable resultDT

Filter Rows | Output Columns

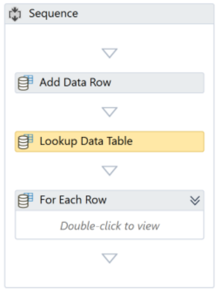
Columns Selection Mode
[✓ Keep] Remove

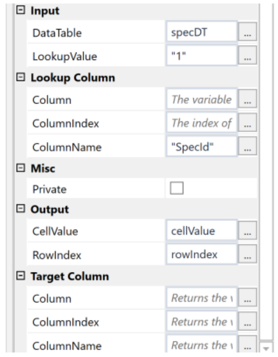
Column
"SpecName"
"StudentName"
"Group"
"Grade"

OK Cancel

Lookup Data Table Activity. Details

- **Lookup Data Table** activity
  - looks for a specified value into the given data table;
  - can be customized in various ways in order to provide specific found value and/or row index only;





Input	
DataTable	specDT
LookupValue	"1"
Lookup Column	
Column	*The variable*
ColumnIndex	*The index of*
ColumnName	"SpecId"
Misc	
Private	☐
Output	
CellValue	cellValue
RowIndex	rowIndex
Target Column	
Column	*Returns the*
ColumnIndex	*Returns the*
ColumnName	*Returns the*

- relevant properties:
  - **[input] DataTable** = **DataTable** variable
  - **[input] LookupValue** = **String** variable
    - the value searched for into the data table;
  - **[lookup column] Column** or **ColumnName** or **ColumnIndex**
    - one of available options can be chosen;
  - **[output] CellValue** = **GenericValue** variable
    - the found value that was found on the looked for column at a returned row index;
  - **[output] RowIndex** = **Int32** variable
    - the row index of the returned value on **CellValue** property;
  - **[target column] Column** or **ColumnName** or **ColumnIndex**
    - if set, it return in **CellValue** the found value, otherwise **CellValue=null**.

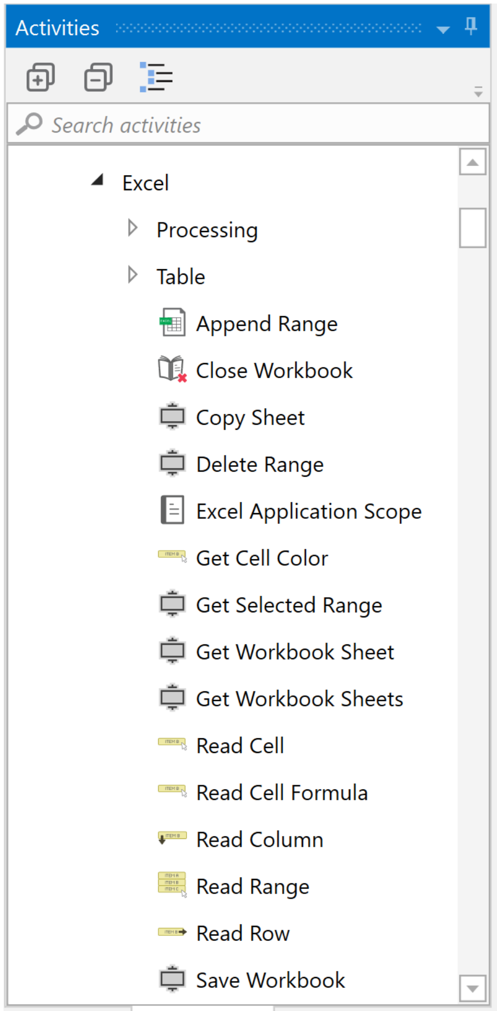
Excel and Data Table. Details

- **Excel** is an application whereas the **Data Table** variables in UiPath Studio mimic Excel operations;

Excel	Data Table
• the Excel application is a package that enables spreadsheet activities with whole work.	• Data Table is a prototype of spreadsheet that presents data in rows and columns without any headers.

Excel Integration. UiPath Activities

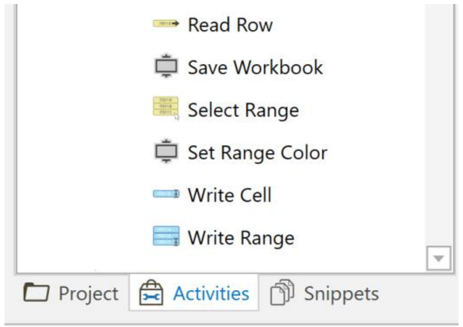
- UiPath provides a series of activities for working with **Excel** files;
- These are activities that are already integrated, being found under the **Application Integration** library, **Excel** section.



Activities

Search activities

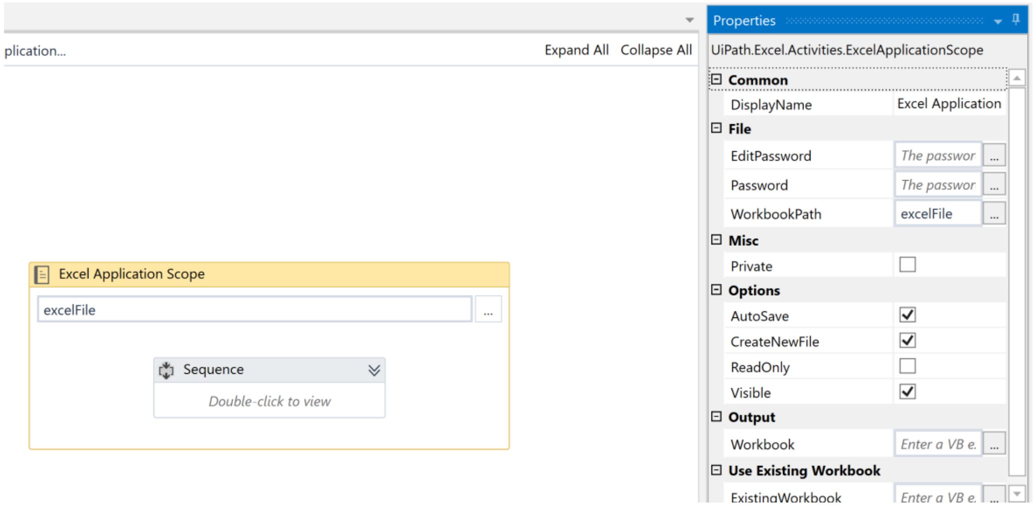
- Excel
  - Processing
  - Table
  - Append Range
  - Close Workbook
  - Copy Sheet
  - Delete Range
  - Excel Application Scope
  - Get Cell Color
  - Get Selected Range
  - Get Workbook Sheet
  - Get Workbook Sheets
  - Read Cell
  - Read Cell Formula
  - Read Column
  - Read Range
  - Read Row
  - Save Workbook



- Read Row
- Save Workbook
- Select Range
- Set Range Color
- Write Cell
- Write Range

Excel Application Scope Activity. Details (1)

- **Excel Application Scope** activity
  - is a container for other activities that work on the same Excel file;
- relevant properties:
  - **WorkbookPath** = *path*+"filename.xlsx"
    - *path* is relative to the location of the current project;
  - **Visible** = **checked/unchecked**
    - **checked** = it reads the file using MS Excel;
    - **unchecked** = it performs the operations internally, directly on the file;



UiPath.Excel.Activities.ExcelApplicationScope

Common	
DisplayName	Excel Application
File	
EditPassword	The passwor
Password	The passwor
WorkbookPath	excelFile
Misc	
Private	☐
Options	
AutoSave	✓
CreateNewFile	✓
ReadOnly	☐
Visible	✓
Output	
Workbook	Enter a VB e.
Use Existing Workbook	
ExistingWorkbook	Enter a VB e.

Excel Application Scope Activity. Details (2)

Checked = Use Excel Application

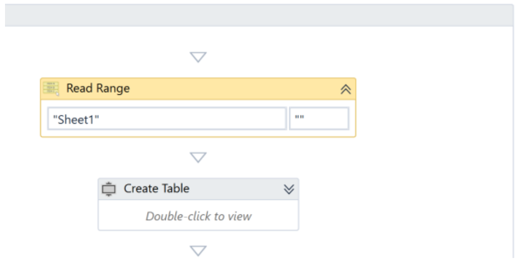
- it requires MS Excel to be installed;
  - The actions are performed through MS Excel application;
- multiple processes can use the same file;
- visible real-time changes into the file;
  - recommended for:
    - debugging;
    - checking the progress of workflow;

Unchecked = Direct Access

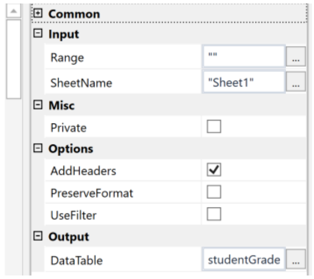
- it does note require MS Excel;
- only one process can use the file;
- it works only for .xlsx file format only.

Read Range Activity. Details

- **Read Range** activity
  - is reads a portion of the Excel file and stores it intro a Data Table variable, used for later operations;
- relevant properties:
  - **[input] SheetName**= “Sheet1”
    - the name of the spreadsheet that will be read from the Excel file;
  - **[input] Range** = “” or “A1:L10”
    - the cell range that will be read from the spreadsheet;
    - “” is the **default range**, i.e., all the data from the spreadsheet;
  - **[input] AddHeaders** = checked/unchecked
    - if checked, the output data table column can be accessed by their names;
  - **[output] DataTable** = **DataTable** variable.


Read Range

"Sheet1" | ""


Common

Input

Range	""
SheetName	"Sheet1"

Misc

Private	☐

Options

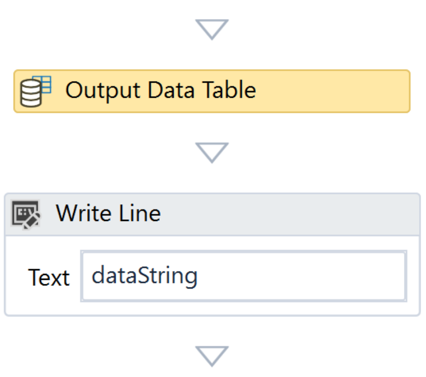
AddHeaders	✓
PreserveFormat	☐
UseFilter	☐

Output

DataTable	studentGrade

Output Data Table Activity. Details

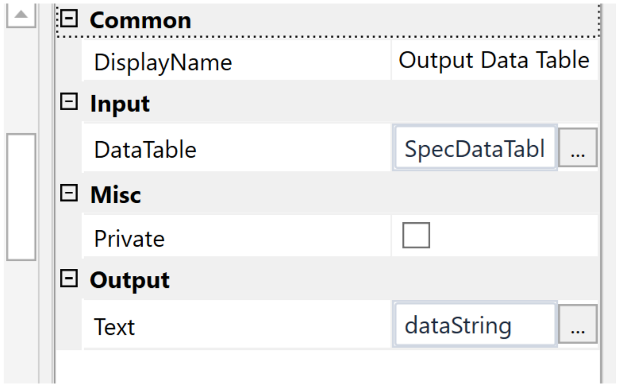
- **Output Data Table** activity
  - converts the information from the data table to string in order to print it properly;
  - it is not meant to print the data table;
  - Is used together with a **Write Line** or **Message Box** activity;
- relevant properties:
  - **[input] DataTable** = **DataTable** variable
    - a data table that whose data will be converted to string;
  - **[output] Text** = **String** variable
    - a variable that stores the data converted to a string.



Output Data Table

Write Line

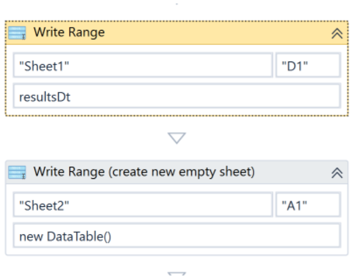
Text: dataString



Common	
DisplayName	Output Data Table
Input	
DataTable	SpecDataTabl
Misc	
Private	☐
Output	
Text	dataString

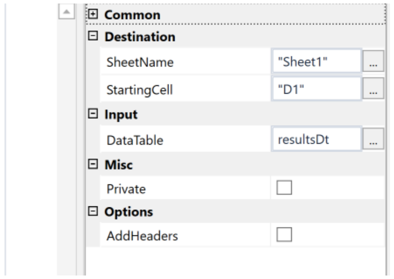
Write Range Activity. Details

- **Write Range** activity
  - allows to write the content of a data table into the file with the specified name inner most **Excel Application Scope** activity;
  - if the file does not exist, it will be created;
  - If there is data on the output file, it will be overwritten, i.e., similar to a Paster operation that starts from the **StartingCell**;
- relevant properties:
  - **[input] DataTable** = **DataTable** variable
  - **[input] SheetName**= "Sheet1"
    - if it does not exists, a new one is created;
  - **[input] StartingCell** = "A1"
    - the cell range that the writing will start from;
  - **[input] AddHeaders** = checked/unchecked
    - if checked, the written data in the file will have column headers written as well.



Write Range
"Sheet1" "D1"
resultsDt

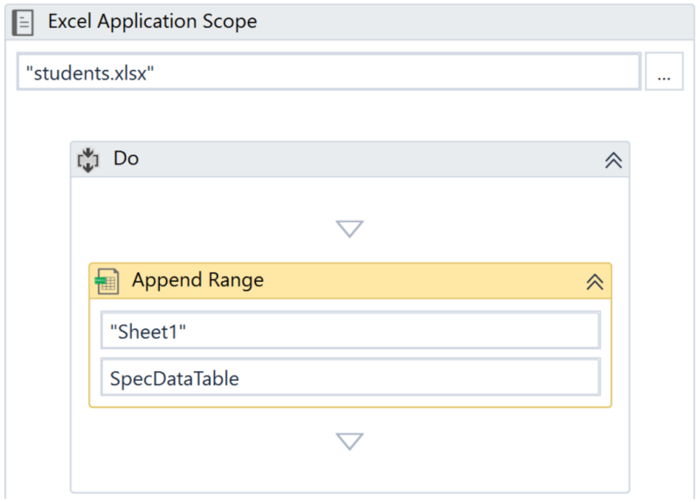
Write Range (create new empty sheet)
"Sheet2" "A1"
new DataTable()

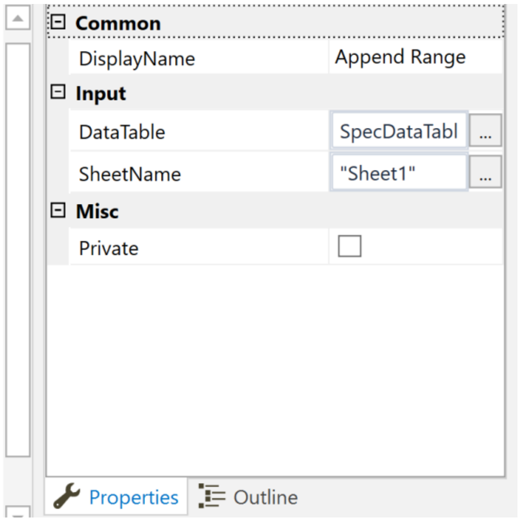


Common	
Destination	
SheetName	"Sheet1"
StartingCell	"D1"
Input	
DataTable	resultsDt
Misc	
Private	☐
Options	
AddHeaders	☐

Append Range Activity. Details

- **Append Range** activity
  - adds the given data after all existing data into the Excel file, without overwriting;
  - is always enclosed within an **Excel Application Scope** activity;
- relevant properties:
  - **[input] DataTable** = **DataTable** variable
  - **[input] SheetName**= “Sheet1”

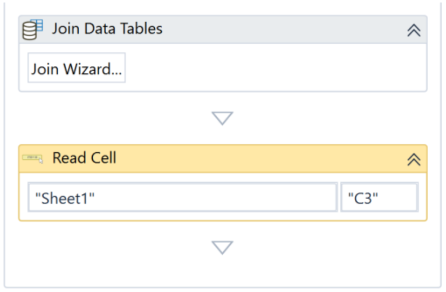

Excel Application Scope
"students.xlsx"
Do
Append Range
"Sheet1"
SpecDataTable

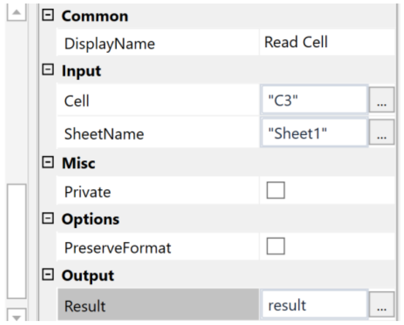


Common	
DisplayName	Append Range
Input	
DataTable	SpecDataTabl
SheetName	"Sheet1"
Misc	
Private	☐

Read Cell Activity. Details

- **Read Cell** activity
  - allows to extract the data from into a designated cell and save it into a variable for later use;
  - is always enclosed within an **Excel Application Scope** activity;
- relevant properties:
  - **[input] Cell** = “C2”
    - a string that indicates the cell to read from;
  - **[input] SheetName**= “Sheet1”
    - the name of the spreadsheet the data will be read from;
  - **[output] Result** = **GenericValue** variable
    - the name of the variable used to store the extracted data.



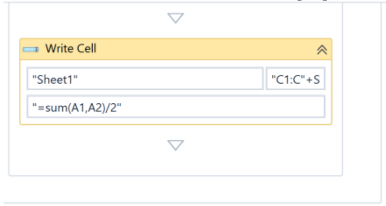


Common	
DisplayName	Read Cell
Input	
Cell	"C3"
SheetName	"Sheet1"
Misc	
Private	☐
Options	
PreserveFormat	☐
Output	
Result	result

Write Cell Activity. Details

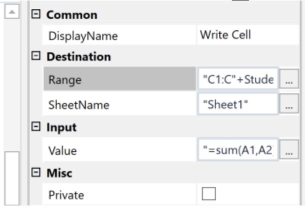
- **Write Cell** activity
  - allows to write the given data from into the specified cell of the Excel file indicated by the inner most **Excel Application Scope** activity;
  - is always enclosed within an **Excel Application Scope** activity;
- relevant properties:
  - **[input] Range**= “C2” or "C1:C"+StudentDataTable.Rows.Count.ToString
    - a string that indicates the cell where to write to;
  - **[input] SheetName**= “Sheet1”
    - the name of the spreadsheet where the data will be written to;
  - **[input] Value** = **GenericValue** variable *or*

    a formula-based expression, e.g., "=sum(A1,A2)/2"
    - the name of the variable that stores the value to be written into the file or a formula-based expression that will be applied to the entire range of cells.


Write Cell

"Sheet1" "C1:C"+S

"=sum(A1,A2)/2"


Common

DisplayName	Write Cell

Destination

Range	"C1:C"+Stude
SheetName	"Sheet1"

Input

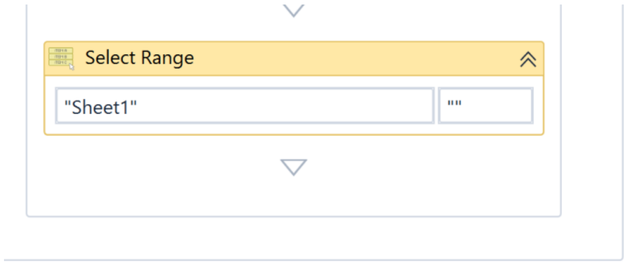
Value	"=sum(A1,A2

Misc

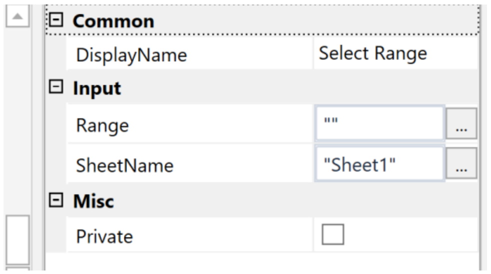
Private	☐

Select Range Activity. Details

- Select Range activity
  - allows to select the cells found in a specified range;
  - **on its own it does not have a direct outcome**, i.e., it is usually combined with a copy, delete, or other operation, **directly on MS Excel**;
  - is always enclosed within an Excel Application Scope activity;
- relevant properties:
  - **[input]** Range = "C2:C4"
    - a string that indicates the cell range to be selected;
  - **[input]** SheetName= "Sheet1"
    - the name of the spreadsheet that contains the selected cell range;


Select Range

"Sheet1"	""


Common

DisplayName	Select Range

Input

Range	""
SheetName	"Sheet1"

Misc

Private	☐

Demo 2. Excel and Data Table Activities

- Consider the Specs.xlsx file with the following data columns:
  - SpecId (int), SpecName (string), NoOfStudents (int).
- Consider the Students.xlsx file with the following data columns:
  - SpecId (int), StudentName (string), Group (int), Lab (int), Project (int), Grade (int).

- Design a process that allows to:

1. Compute the final Grade for each student, based on the Lab an Project grades;
2. Build a ranking file Ranking.xlsx consisting of the following details: SpecName (string), StudentName (string), Group (int), Grade (int), ordered descending by their grade;
3. Build a classbook file Classbook.xlsx with the following details:
   - For each specialization there a distinct spread sheet, that consists of StudentName (string), Group (int), Grade (int), ordered by groups and student names;
4. Update the last data column from Spec.xlsx file with the number of students from the corresponding specialization.

References

- UiPath Academy - https://academy.uipath.com
  - Level 1 – Foundation Training, Lesson 9;
- UiPath Docs - https://docs.uipath.com/studio
  - Data Table variables - https://docs.uipath.com/studio/docs/data-table-variables
  - Integrated Excel activities - https://docs.uipath.com/activities/docs/app-integration-excel