# **AUTOMATION ANYWHERE ENTERPRISE – CSV Processing**

# ***Version History***

|  |  |  |
| --- | --- | --- |
| Date | Author | Change Description |
| 10/1/2018 | Brendan Sapience | First Release |
|  |  |  |

# ***Notes***

1. No prerequisite for this metabot, it should be self-contained
2. Using certain features of this metabot require knowledge of Regular Expression, what a Regular Expression Pattern is and what Regular Expression Matching Groups are. A great place to start exploring Regular Expressions (RegEx for short) is <regex101.com>
3. Regex101.com can be used to test the RegEx that are available as parameters below
4. All options below are case sensitive (Column Names, etc.)

# ***Metabot Functions/Logics***

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | **Inputs** | **Outputs** | **Comments** |
| Add Column After | String – vInputFilePath  String - vInputColumnName  String – vInputColumnNameToAdd  String - vInputDataFiller |  | Add column after a given Column (based on Column Name). vInputDataFiller is the content to use to fill out every cell of the new column. |
| Add Column Before | String – vInputFilePath  String- vInputColumnName  String – vInputColumnNameToAdd  String - vInputDataFiller |  | Add column before a given Column (based on Column Name). vInputDataFiller is the content to use to fill out every cell of the new column. |
| Delete Column | String – vInputFilePath  String- vInputColumnName |  | Delete a column and all its content (based on column name) |
| Delete Lines If Cell Matches Regex | String – vInputFilePath  String – vInputColumnNameToProcess  String (RegEx) - vInputRegExToMatch |  | Delete all Lines IF and only IF the Cell in column vInputColumnNameToProcess matches the Regular Expression in vInputRegExToMatch |
| Get Column Names | String – vInputFilePath | String - vOutputListOfColumns | Returns all column names (values are comma separated) |
| Get Line Content | String – vInputFilePath  Integer - vInputLineNumber | String - vOutputLineContent | Returns the content of a line (values are comma separated) |
| Keep Lines If Cell Matches Regex | String – vInputFilePath  String – vInputColumnNameToProcess  String (RegEx) - vInputRegExToMatch |  | Keep all Lines IF and only IF the Cell in column vInputColumnNameToProcess matches the Regular Expression in vInputRegExToMatch |
| Set Cell Content | String – vInputFilePath  String- vInputColumnName  Integer - vInputLineNumber  String - vInputNewCellValue |  | Set the content of cell from column vInputColumnName and line number vInputLineNumber to the value stored in vInputNewCellValue |
| Transform Column Content with Regex Group Match\* | String – vInputFilePath  String- vInputColumnName  String (RegEx) - vInputRegExWithGroup |  | Extract the first Group Match of the regular expression in vInputRegExWithMatch against the content of all cells from column vInputColumnName and replace the value of each cell with the substring value (the RegEx group match). |
| Append Text to Column If Column Matches Regex Pattern | String – vInputFilePath  String- vInputColumnNameToMatch  String (RegEx) – vInputRegExToMatch  String - vInputNameOfColumnToModify  String - vInputStringToAppendToColumn |  | Matches regular expression vInputRegExToMatch to the content of each cell from column vInputColumnToMatch. If it matches, the string from vInputStringToAppendToColumn is appended in front of the content of column vInputNameOfColumnToModify |
| Get Cell Content | String – vInputFilePath  String- vInputColumnName  Integer - vInputLineNumber | vOutputCellContent | Returns the content of the cell from line vInputLineNumber at column vInputColumnName |
| Get Number of Columns | String - vInputFilePath | vOutputColumnNumber | Returns the number of columns in CSV file |
| Set Cell Values Based on Line Range | String – vInputFilePath  String- vInputColumnName  Integer – vInputRangeStartLine  Integer - vInputRangeEndLine  String - vInputNewCellValue |  | Changes the value of Cells in column vInputColumnName from line vInputRangeStartLine all the way to line number vInputRangeEndLine to vInputNewCellValue |
| Copy Cell Content to Other Column Based on Regex Expression Match\* | String – vInputFilePath  String- vInputSourceColumnName  String- vInputTargetColumnName  Integer – vInputSourceLineNumber  String (RegEx) - vInputRegExWithGroupMatch |  | Extract the first Group Match of the regular expression in vInputRegExWithGroupMatch against the content of cell from line vInputSourceLineNumber and column vInputColumnName and put this substring value on the same line at column vInputTargetColumnName. |
| Copy Column Content to Other Column Based on Regex Expression Match\* | String – vInputFilePath  String- vInputColumnNameSource  String- vInputColumnNameTarget  String (RegEx) - vInputRegExWithGroupMatch |  | Extract the first Group Match of the regular expression in vInputRegExWithGroupMatch against the content of column vInputColumnNameSource and put this substring value in column vInputColumnNameTarget. |
| Get Column Index | String – vInputFilePath  String- vInputColumnName | vOutputIndexNumber | Returns the index (position) of a column name (positions start at 0). |
| Get Number of Lines | String – vInputFilePath | vOutputNumberOfLines | Returns the total number of lines in csv (including header line) |
| Transform Cell Content with Regex Group Match\* | String – vInputFilePath  String- vInputColumnName  Integer – vInputLineNumber  String (RegEx) - vInputRegExWithMatch |  | Extract the first Group Match of the regular expression in vInputRegExWithMatch against the content of cell from line vInputLineNumber and column vInputColumnName and replace the value of the cell with this substring value. |
| Swap Columns | String – vInputFilePath  String- vInputColumnName1  String – vInputColumnName2 |  | Swap two columns in the CSV file. vInputColumnName2 will move to vInputColumnName1’s position and vice versa. |
| Enforce Column Order | String – vInputFilePath  String- vInputCsvColumnOrderToApply |  | Rearrange all columns of a file based on vInputCsvColumnOrderToApply. This parameter must be passed as a csv string, with ALL columns in order. Ex: if the current order is “Col1,Col2,Col3,Col4” and we want to invert the order, you will need to pass the following string as a parameter “Col4,Col3,Col2,Col1”. |
| Rename Column | String – vInputFilePath  String- vInputColumnName  String- vInputNewColumnName |  | Renames a column |
| Replace String In Column Content | String – vInputFilePath  String- vInputColumnName  String – vInputStringToReplace  String - vInputStringReplacement |  | Replaces a particular String with another one in an entire column (ex: replace 2018 with 2019 in column Invoice\_Date) (vInputStringReplacement cannot be empty, use “Delete String From column Content” instead) |
| Delete String from Column Content | String – vInputFilePath  String- vInputColumnName  String – vInputStringToReplace |  | Delete a substring from all values of one column |
| Get CSV as JSON | String – vInputFilePath  String- vInputSectionTitleForSingleValues  String – vInputSectionTitleForItemsSection  String – vInputCSVListOfColumnsOfStandardFields  String - vInputCSVListOfColumnsOfItemizedFields |  | Returns the content of the CSV file as a JSON file (in which there are two sections:  one titled after vInputSectionTitleForSingleValues which contains the values from all fields declared in vInputCSVListOfColumnsOfStandardFields  one titled after vInputSectionTitleForItemsSection which contains LIST of values from all fields declared in vInputCSVListOfColumnsOfItemizedFields |
| Split Column Content to New Columns | String – vInputFilePath  String- vInputColumnNameToReadFrom  String – RegexPatternWithMatches  String – vInsertAfterColumnName  String - vInputNewColumnNameStub |  | Split the content of an existing Column (based on Regex Matches) and copy each element into a new columns (all new columns are named vInputNewColumnNameStub[n], all columns are inserted after column vInsertAfterColumnName) |

\*: Functions use the concept of Regular Expression **Capturing Groups**. More info here: <https://www.regular-expressions.info/brackets.html>. Exactly 1 capturing group needs to be defined in the Regular Expression for the function to work.

Ex:

**“^My Value.\*$”** **does NOT** contain any Capturing Group and will **not result in any csv modification**.

**“^My Value(.\*)$”** **DOES** contain a capturing Group “(.\*)”, its content is used to modify the CSV according to the function