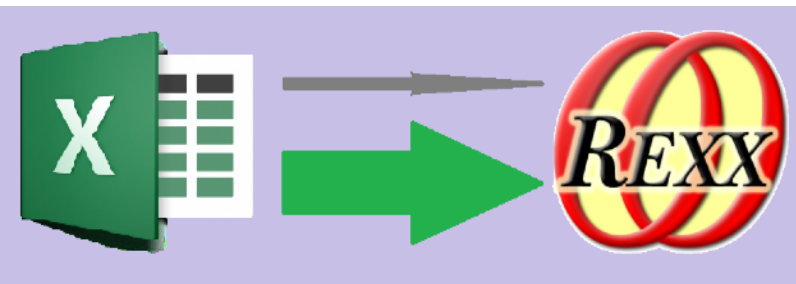


Extracting large amounts of data efficiently from excel using ooRexx

33rd International Rexx Language Symposium
September 2022

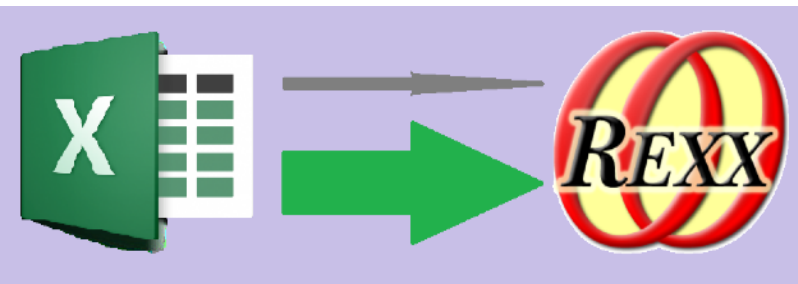


16th Annual Rexx Symposium

My first at Redondo Beach in 2005

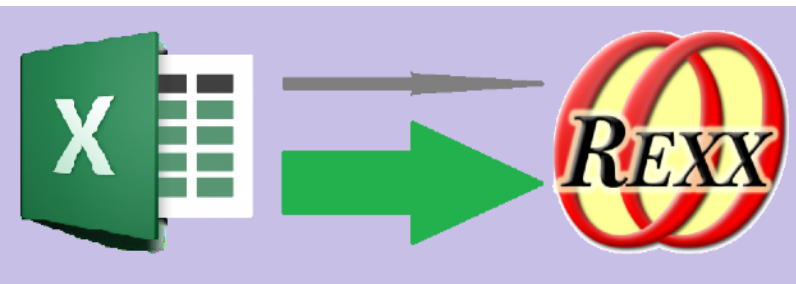
A struggle to convince my employers that it would be good value

- Mike presented a rexx HTML server as part of his weekend wiki presentation
- Howard gave us all copies of his book and we started collaborating
- Rony presented BSF4Rexx and Open Office automation and gave me a private tutorial on suppliers
- David presented the rxFtp object
- Lee presented the rxVb.printerObject and ...

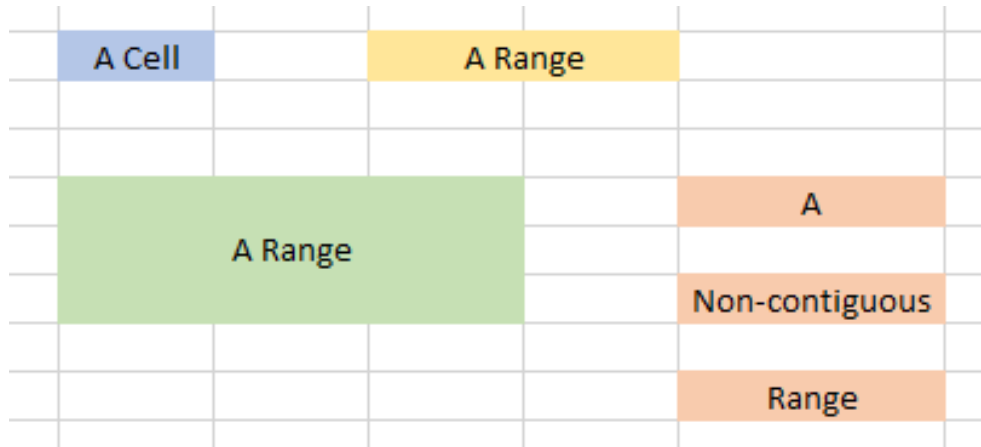


Automating Microsoft Excel Via ooRexx ActiveX/OLE

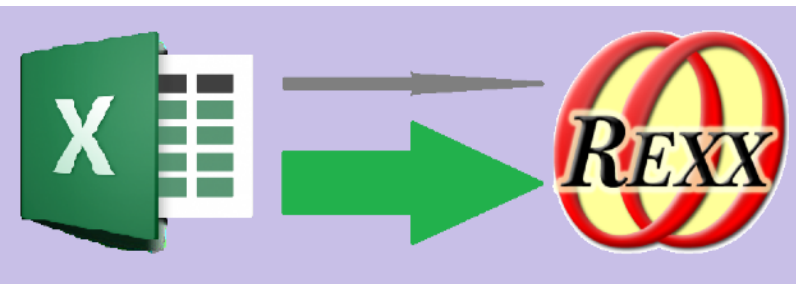
- For me a seminal talk.
- Lee demonstrated pretty much everything one could do to automate excel
- He also showed how to lever the Macro Recorder to produce ooRexx code
- Retrieving a value from a cell:
 - `bar = xlObj~Cells(row,column)~Value`
- I've used that method ever since



Retrieving values from excel

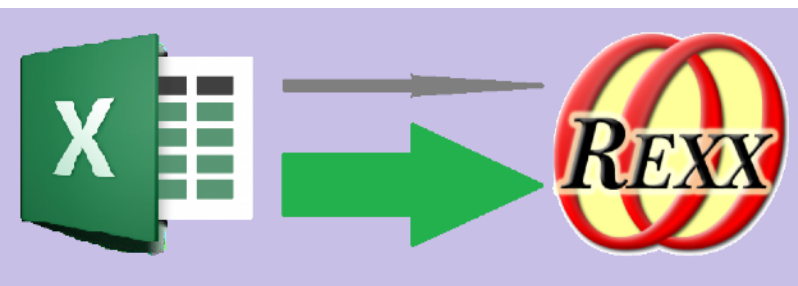


- A cell (like Lee used) is a range containing one cell
- Ranges can contain more than one cell
- Single Cell ranges return a value from the value method
- Contiguous multi-cell ranges return an array of values from the value method
- Where there is more than one row and more than one column the value method returns an ooRexx 2-dimensional array.



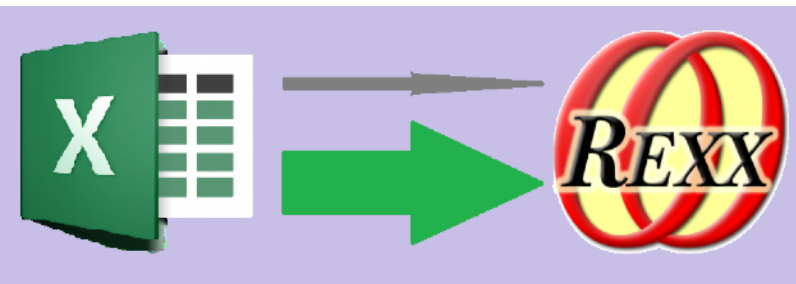
Comparing Speeds

- Last year I had to retrieve a lot of data exported to excel worksheets as reports by Quickbooks online
- I found that it is much quicker to fetch the values for the entire report into a 2-dimensional array than to fetch a cell at a time.
- LETS HAVE A DEMO
 - Using sample data
 - We can compare the speed of import



The arrayTable Class

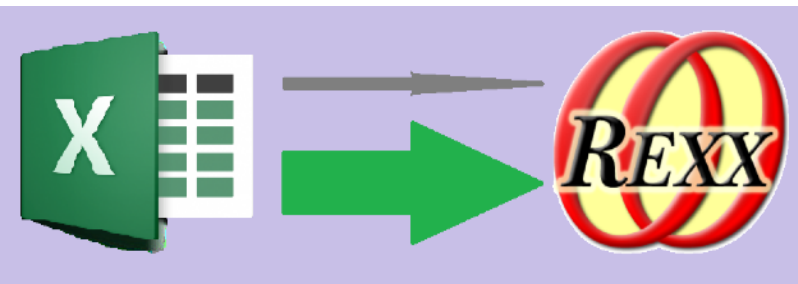
- Assistance class to make using this technique easy
 - Class Method
 - **fromXlSheet** – creates and populates an arrayTable from an excel worksheet ole proxy
 - Instance Methods
 - **[]** – allows access to the data by row number and column header or number
 - **makearray** – allows use of ooRexx **do row over arraytable** syntax
 - **addDataTransform** – optionally allows post-processing of read data
 - **toXlSheet** – data (values only) can be written to an excel worksheet ole proxy
 - **unknown** – all other messages forwarded to the two dimensional array of data



Creating an arrayTable from an excel worksheet

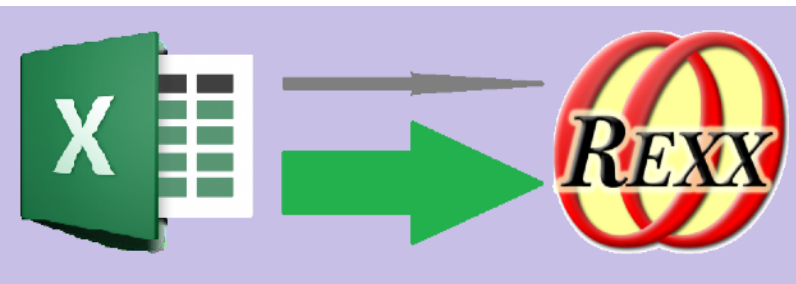
- Two public routines available
 - OpenOrActivateWorkbook – opens an excel spreadsheet or makes it active if already open
 - activateOrCreateSheet – activates (or creates) a worksheet within an excel workbook
- Code to create ArrayTable of demo data

```
file      = 'sampledatainsurance.xlsx'  
xlObj     = OpenOrActivateWorkbook(file)  
wksObj    = activateOrCreateSheet(xlObj, 'PolicyData')  
policies  = .arrayTable~fromXlSheet(wksObj)  
  
::requires "arrayTable.cls"
```



Reading from an arraytable

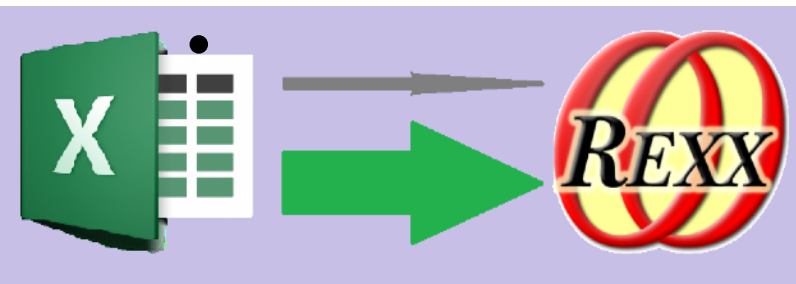
```
polices = .arrayTable~fromXlSheet(wksObj)
do row over polices
  rList~addRow(, ,polices[row, 'Policy']           -
              ,polices[row, 'Expiry']             -
              ,polices[row, 'Location']           -
              ,polices[row, 'State']              -
              ,polices[row, 'Region']             -
              ,polices[row, 'InsuredValue']       -
              ,polices[row, 'Construction']       -
              ,polices[row, 'BusinessType']       -
              ,polices[row, 'Earthquake']         -
              ,polices[row, 'Flood']              )
end /* DO */
drop polices
```



Handling headers and margins

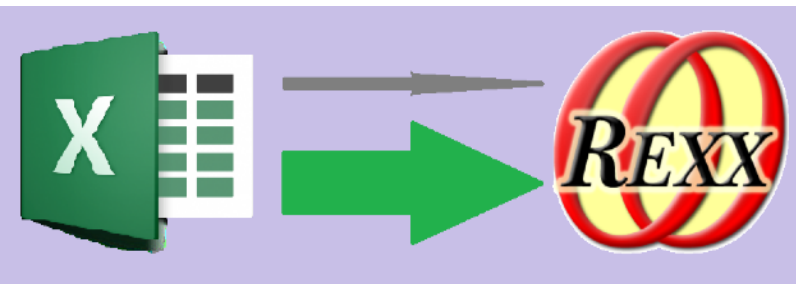
- fromXlSheet
 - You can specify margins to ignore around the edge of the populated area of the worksheet
 - Unless you pass an array of column headers, the first row of the table is assumed to be column headers.
- Let's see a sample report as it would be exported from QuickbooksOnline

```
arrayTable = .arrayTable~fromXlSheet(wksObj, 2, 5,, -3)
```



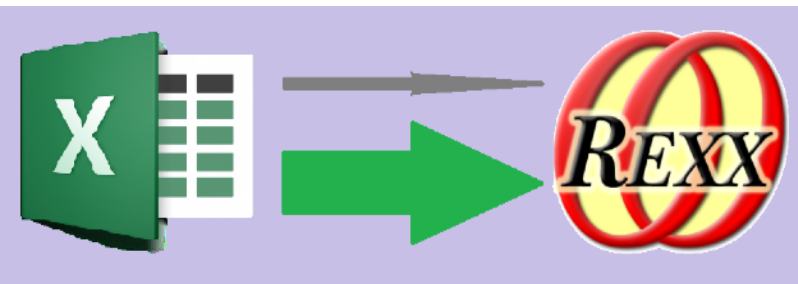
These parameters would work for any report from Quickbooks online

Do we have time to go on to dataTransforms or should we stop here for questions?



Data Transforms

- I was moving data from excel to mySql, so I always wanted to mySql escape it
- Rather than code that all over the place I wanted to give the task to the array table to apply at the end of the `[]` method
- AddDatatransform allows one to add a data transform

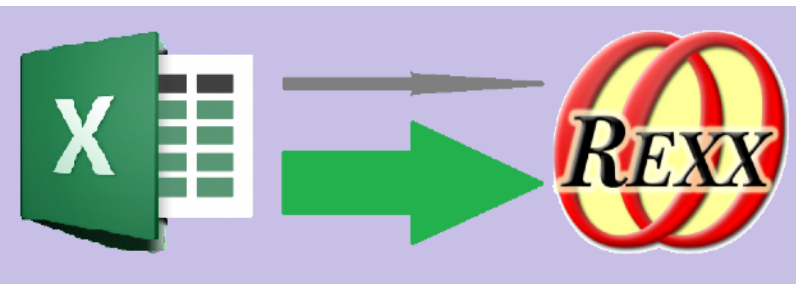


Data Transforms – Provided Samples

- 1) arrayTable_dataTransform_mySqlEscape

```
/* ===== */  
::CLASS arrayTable_dataTransform_mySqlEscape subclass arrayTable_dataTransform public  
/* ===== */  
::method process  
/* ----- */  
use arg txt  
RETURN txt~changeStr('\', '\\')~changeStr('"', '\"')~changeStr("'", "\'")~changeStr('00'x, '\'|'00'x)  
/* ===== */
```

- Subclasses arrayTable_dataTransform
- Implements a process method performing the data transform

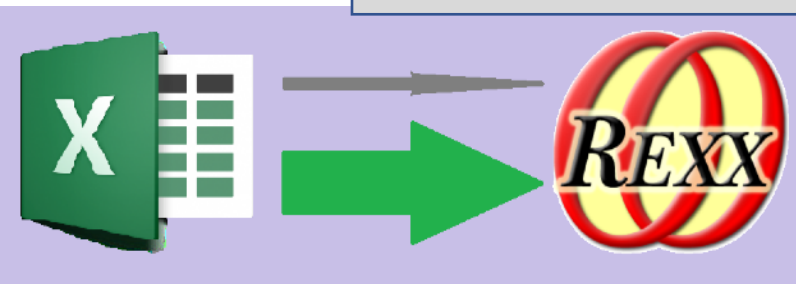


Data Transforms – Provided Samples

- 2) arrayTable_dataTransform_ifNil
- Converts uninitialized cells from the nil object to a string
- Accepts a replacement string on the new method

```
::CLASS arrayTable_dataTransform_ifNil subclass arrayTable_dataTransform public
::method init
  expose defValue
  use arg defValue = ''
  self~init:super

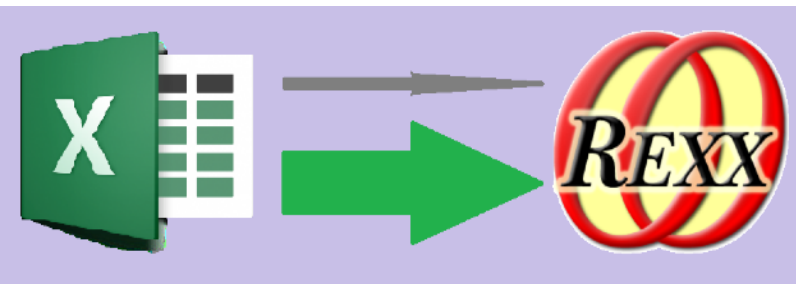
::method process
  expose defValue
  use arg data
  if data~isNil
  then do
    data = defValue
    self~endPostProcess? = .true -- no more datatransforms will be processed
  end /* DO */
RETURN data
```



Data Transforms – Provided Samples

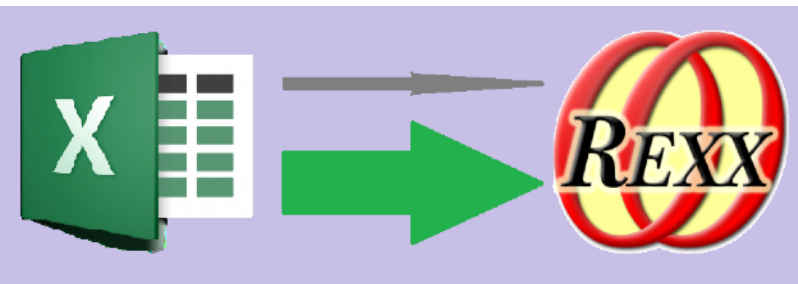
- 3) arrayTable_dataTransform_dittos
- A frivolous example - replaces the word 'ditto' with value of cell above
- Utilizes 'row', 'col' & 'array' attributes provided by superclass

```
::CLASS arrayTable_dataTransform_dittos subclass arrayTable_dataTransform public
::method process
use arg data
  if 'ditto'~caselessEquals(data)
  then do
    data = self~array[self~row - 1, self~col]
    self~array[self~row, self~col] = data -- write it to array table in
  end /* DO */ -- case of cascading dittos
RETURN data
```



Data Transforms

- Let us have a Demo



Any Questions?

