Scripting notes – data framing excel

6/13/21

Update tblObject.xlsm

1. Workbook status able to toggle between usage and best VBA performance
   1. Import latest SaveWkbkStatus and ResetWkbkStatus
   2. Add global constants for sStatusRun and sStatusOrig

tblRowsCols

Provision

FormatTbl

* ReadSetting [modUtilities]
* RefreshTblClsHeader [tblRowsCols]
* ~~SetTableWidthAndNumFormat [tblRowsCols]~~ only with ColInfo
* ~~SetTableComments [tblRowsCols]~~
* ~~AutosizeComments [tblRowsCols]~~
* SetTableNames [tblRowsCols]

3/26 - 29/2021

Item #9 below – getting Validation code untangled from proprietary Test project workbooks

Scenario model can be detected in workbook if Setting exists with metadata about the model

Settings is a good way to keep a [hidden] inventory of what models exist in a workbook and also to gain access to metadata details about the Scenario Model’s configuration (IsLite, IsCalculator etc.

1. Add generic sht/model name and Filename columns to Validation worksheet (10 sets to allow import of up to 10 configuration files

* Sht/Model = name of [destination] worksheet in test workbook
* Filename = name of [source] workbook at Validation\_path path name

1. Proof of concept with single table (tblPorts) import
2. Move all generic rows/columns config imports to new approach (delete existing Val columns for those)
3. Convert remaining Test File tables to generic so that all external calls are to New\_tbl function
   * + 1. tblNetwork
       2. tblFormulation
       3. tblXMLWrite (need to set up tblClass to Provision and format it)
       4. tblProcess
       5. tblColInfo
4. Convert tblF Config import to generic (ability to import just a few columns that are in Config file –not all columns in table; should work already based on matching column names but need to verify)
5. Convert **IsDiagnosticPrint** to Scenario Model setting
6. Eliminate cellhome as **SetTableAttributes** argument (set it in Provision instead)
7. Move header refresh to Provision since names are needed for wayfinding to sense ncols
8. Add settings Scenario Model Provision (keep track of IsCalculator etc.)
9. Add ability to generically import Scenario Model info
10. Delete legacy code from proprietary workbook

3/16 2021

Re-factor model validation in proprietary workbook – Overall plan

1. ~~Move modValidation to ValTest workbook~~
2. ~~Move scenario model to separate worksheet~~
3. ~~Re-order ValTests columns~~
4. ~~Re-order Validation columns~~
5. Add WriteValue Step to poke settings into Val test workbook
6. Move Compare tables to separate sheets
7. Add auto refresh to name Compare tables
8. Add cell compare scenario model that auto names
9. Rationalize customization in modValidation in proprietary test workbook

Step 5: re-order columns on ValTests worksheet (and add formatting)

Step 4: Move validation table to right to make room for button –working in 0817\_1620 version but need to resolve some validation Fails

Step 3: re-order columns on Validation worksheet (and add formatting)

Detoured to add some formatting functions in MOD utilities. These can take a range of rows and an array of column ranges and apply formatting to the intersection of the rows and columns for the entire array

Step 2: move scenario model to separate work sheet. – working in\_0817\_0757 version

This involved getting PrepExcelSteps into ValTests workbook because the lite version of scenario model requires this

Step 1: Move modValidation to ValTest workbook

Test ability to instantiate a class from another project by adding a reference to that workbook

MSDN with instructions on how to do (their “early binding” example matches expectation from prior experience): <https://docs.microsoft.com/en-us/previous-versions/office/troubleshoot/office-developer/set-up-vb-project-using-class>

* Add properties to tblGeneric Class (should do this with all)
* Add subroutine new\_tbl() to modUtilities to instantiate tblGeneric, tblProcess etc. This should be standard

<https://stackoverflow.com/questions/18389562/cannot-run-the-macro-the-macro-may-not-be-available-in-this-workbook>

A couple of watchouts to work through before getting working code below:

1. Path + file name needs to be enclosed in single quotes like this (matches MSDN article):

sRunCommand = "'" & sFileTest & "'!New\_tbl"

1. Placed New\_tbl subroutine in test file (modUtilities) per MSDN article

Graphical user interface, text, application

Description automatically generated

1. Trust Center change (see Stackoverflow post) was not needed
2. MSDN advice not needed
   1. don’t need to set Class instancing property to PublicNotCreatable
   2. don’t need to add Property Get/Let Name to Class

Working demo subroutine:

Graphical user interface, text, application, email

Description automatically generated

Needed in ValTests:

* mod Validation module
* Constants
* tblGeneric Class object

Need to switch RunTests function to open test workbook instead of ValTests which is now “ThisWorkbook”

Need to modify tbl process provision routine to allow workbook object as optional input. TblGeneric already has this