**VBA Functions added to RASController for Sensitivity Analysis**

Sub GetDetailedTS()

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'Demonstrates the Plan\_Names subroutine and \_

Plan\_GetFilename function

'Written by Christopher Goodell as GetWSELandVelocity

'August 24, 2013

'Edited by Emma House and renamed -- this version still has a single loop but gets the TS

'April 08, 2024

'Opens an existing HEC-RAS project, executes it and \_

prints the results to a spreadsheet.

'Tested with the BEAVCREK.prj data set.

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'Instantiate a new RAS620.HECRASController Class

Dim RC As New RAS620.HECRASController

'Specify the HEC-RAS project, open it, execute it.

Dim strRASProject As String 'The path and filename of the \_

desired HEC-RAS Project

Dim lngMessages As Long 'number of messages returned by \_

the RASController

Dim strMessages() As String 'an array of messages \_

returned by the RASController

Application.ScreenUpdating = False

strRASProject = "L:\NSF\_Iowa\_L\RAS\_Toy\" & \_

"RAS\_Toy\_Copy.prj"

RC.Project\_Open strRASProject

' Comment the following line to save time if you don't need to run

'RC.ShowRAS

RC.Compute\_CurrentPlan lngMessages, strMessages()

'Find the number of Profiles (Time steps)

' !! Changed from number of Nodes (River Stations)!

'Declare variables for the River ID and Reach ID. Since \_

there is only one river and reach, \_

their ID's can both be hardcoded to 1.

Dim lngRiverID As Long, lngReachID As Long ' River ID and \_

Reach ID

Dim lngNum\_PF As Long 'Number of profiles - HEC-RAS \_

Controller will populate.

' !! Changed from lngNum\_RS (number of nodes)

Dim strPF() As String 'Array of names of the profiles

' !! Changed from strRS (names of the nodes)

lngRiverID = 1

lngReachID = 1

RC.Output\_GetProfiles lngNum\_PF, strPF()

'old:

'RC.Geometry\_GetNodes lngRiverID, lngReachID, lngNum\_PF, \_

' strPF(), strProfType()

'Declare single arrays for all detailed vars

' !! Maybe have to make these more than single arrays if you want multiple profiles

Dim sngWS() As Single 'Array of water surface elevations

Dim sngAvgVel() As Single 'Array of Avg Velocities

Dim sngQtot() As Single 'Array of total flows

Dim sngFlowA() As Single 'Array of flow areas

Dim sngSf() As Single 'Array of friction Slopes

Dim sngHydD() As Single 'Array of hydraulic depths

Dim sngDSWS() As Single 'Array of water surface elevations

Dim sngDSAvgVel() As Single 'Array of Avg Velocities

Dim sngDSQtot() As Single 'Array of total flows

Dim sngDSFlowA() As Single 'Array of flow areas

Dim sngDSHydD() As Single 'Array of hydraulic depths

'Redimension single arrays to the number of profiles

ReDim sngWS(1 To lngNum\_PF)

ReDim sngAvgVel(1 To lngNum\_PF)

ReDim sngQtot(1 To lngNum\_PF)

ReDim sngFlowA(1 To lngNum\_PF)

ReDim sngSf(1 To lngNum\_PF)

ReDim sngHydD(1 To lngNum\_PF)

ReDim sngDSWS(1 To lngNum\_PF)

ReDim sngDSAvgVel(1 To lngNum\_PF)

ReDim sngDSQtot(1 To lngNum\_PF)

ReDim sngDSFlowA(1 To lngNum\_PF)

ReDim sngDSHydD(1 To lngNum\_PF)

' !! Changed from lngNum\_RS (number of river stations)

ReDim strPF(1 To lngNum\_PF) ' added

'Declare variables for the Water Surface Elevation and \_

Average Velocity IDs.

Dim lngWS\_ID As Long 'The Water Surface Elevation ID is 2.

Dim lngAvgVel\_ID As Long 'The Average Velocity ID for the \_

total cross section is 23.

Dim lngQtot\_ID As Long 'The total Flow ID is 9.

Dim lngFlowA\_ID As Long 'The flow area ID is 10.

Dim lngSf\_ID As Long 'The friction slope ID is 68.

Dim lngHydD\_ID As Long 'The hydraulic depth ID is 126.

' removed this one since it's constant 1,000 ft for toy lngLen\_ID = 42

lngWS\_ID = 2

lngAvgVel\_ID = 23

lngQtot\_ID = 9

lngFlowA\_ID = 10

lngSf\_ID = 68

lngHydD\_ID = 126

'Search for profiles and populate the single var, and strRS() arrays

' !! changing the i loop to go through each profile instead of node

' !! need to specify the XS we want to read (also add DS XS & its vars)

Dim i As Long 'Index for the time For-Next Loop

For i = 1 To 10

'For i = 1 To lngNum\_PF

Application.StatusBar = "retrieving data for " & i & " of " & lngNum\_PF

sngWS(i) = RC.Output\_NodeOutput(lngRiverID, \_

lngReachID, 5, 0, i, lngWS\_ID)

' !! changed to hard code the node and iterate through profiles

sngAvgVel(i) = RC.Output\_NodeOutput(lngRiverID, \_

lngReachID, 5, 0, i, lngAvgVel\_ID)

' !! changed to hard code the node and iterate through profiles

sngQtot(i) = RC.Output\_NodeOutput(lngRiverID, \_

lngReachID, 5, 0, i, lngQtot\_ID)

' !! added

sngFlowA(i) = RC.Output\_NodeOutput(lngRiverID, \_

lngReachID, 5, 0, i, lngFlowA\_ID)

' !! added

sngSf(i) = RC.Output\_NodeOutput(lngRiverID, \_

lngReachID, 5, 0, i, lngSf\_ID)

' !! added

sngHydD(i) = RC.Output\_NodeOutput(lngRiverID, \_

lngReachID, 5, 0, i, lngHydD\_ID)

' !! added

sngDSWS(i) = RC.Output\_NodeOutput(lngRiverID, \_

lngReachID, 6, 0, i, lngWS\_ID)

' !! added (DS Node)

sngDSAvgVel(i) = RC.Output\_NodeOutput(lngRiverID, \_

lngReachID, 6, 0, i, lngAvgVel\_ID)

' !! added (DS Node)

sngDSQtot(i) = RC.Output\_NodeOutput(lngRiverID, \_

lngReachID, 6, 0, i, lngQtot\_ID)

' !! added

sngDSFlowA(i) = RC.Output\_NodeOutput(lngRiverID, \_

lngReachID, 6, 0, i, lngFlowA\_ID)

' !! added

sngDSHydD(i) = RC.Output\_NodeOutput(lngRiverID, \_

lngReachID, 6, 0, i, lngHydD\_ID)

' !! added

strPF(i) = i ' doesn't give the names but gives the profile ID's at least

' !! old: strRS(i) = RC.Geometry.NodeRS(lngRiverID, \_

lngReachID, i

Next i

'Create a spreadsheet with Profiles, Water Surface \_

Elevations and Velocities, and more.

'First check to see if the worksheet "RASResults" already \_

exists. If not, add a new sheet and name it.

Dim blnSheetExists As Boolean 'Boolean tag to set to true \_

if a "RASResults" sheet already exists.

For i = 1 To Sheets.Count

If Sheets(i).Name = "RASResults" Then

blnSheetExists = True

End If

Next i

'If "RASResults" does not exist as a sheet, we'll add it.

If blnSheetExists = False Then

Sheets.Add

ActiveSheet.Name = "RASResults"

End If

'Select the Output sheet (should change this to be the new name that changes each iteration...)

Sheets("Output").Select

'Clear all the data so that can construct a fresh table of data (probs remove this)

Cells.Select

Cells.Delete

'Give the sheet a title.

Range("A1").Select

ActiveCell.Value = "Results from HEC-RAS " & \_

"run: Spring 2019 event with" & strPF(i)

'Include the Active Plan Name

Range("A2").Select

ActiveCell.Value = GetCurrentPlanName(RC)

'Populate the table headers.

' !! May need to reorder these since we have those calculations set up differently

Range("A3") = "Profile"

Range("B3") = "Tot Q (cfs)"

Range("C3") = "W.S. El. (ft)"

Range("D3") = "Avg. Vel. (ft/s)"

Range("E3") = "Flow Area (sqft)"

Range("F3") = "Frctn. Slope (-)"

Range("G3") = "Hyd. Depth (ft)"

Range("H3") = "Len. Chnl. (ft)" ' keeping it here so I can fill it with the val later

Range("I3") = "DS Tot Q (cfs)"

Range("J3") = "DS W.S. El. (ft)"

Range("K3") = "DS Avg. Vel. (ft/s)"

Range("L3") = "DS Flow Area (sqft)"

Range("M3") = "DS Hyd. Depth (ft)"

'Loop through the HEC-RAS results arrays and populate the \_

spreadsheet. Water surface values are rounded to 9 \_

decimal places. Velocities to 9 decimal places, too.

'First, declare string values for conversion in the Round \_

function.

Dim strWSEL As String, strAvgVel As String, strQtot As String, \_

strFlowA As String, strSf As String, strHydD As String, strLen As String

Range("A4").Select

'For i = 1 To 100 ' !! Changed from lngNum\_RS

For i = 1 To lngNum\_PF ' !! Changed from lngNum\_RS

Application.StatusBar = "tabulating data for " & i & " of " & lngNum\_PF

strWSEL = Round(sngWS(i), 9)

strAvgVel = Round(sngAvgVel(i), 9)

strQtot = Round(sngQtot(i), 9)

strFlowA = Round(sngFlowA(i), 9)

strSf = Round(sngSf(i), 9)

strHydD = Round(sngHydD(i), 9)

strDSWSEL = Round(sngDSWS(i), 9)

strDSAvgVel = Round(sngDSAvgVel(i), 9)

strDSQtot = Round(sngDSQtot(i), 9)

strDSFlowA = Round(sngDSFlowA(i), 9)

strDSHydD = Round(sngDSHydD(i), 9)

'Step through the table, populating it with HEC-RAS \_

output.

ActiveCell.Value = strPF(i)

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = strQtot

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = strWSEL

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = strAvgVel

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = strFlowA

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = strSf

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = strHydD

ActiveCell.Offset(0, 2).Activate

ActiveCell.Value = strDSQtot

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = strDSWSEL

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = strDSAvgVel

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = strDSFlowA

ActiveCell.Offset(0, 1).Activate

ActiveCell.Value = strDSHydD

ActiveCell.Offset(1, -12).Activate

Next i

'Close HEC-RAS

MsgBox "Done!"

RC.QuitRAS

Application.ScreenUpdating = True

End Sub

Sub MonteCarloScenarios()

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'Demonstrates running HEC-RAS in a Monte Carlo Experiment.

'Written by Christopher Goodell

'November 7, 2013

'Edited by Emma House and renamed -- this version codes input values rather than getting from distribution

'April 30, 2024

'(Old def) This subroutine randomly samples n values about a normal \_

distribution and applies that n value to the main \_

channel of each cross section. The HEC-RAS project \_

is re-run after each sampling. Water surface \_

elevations are stored at the completion of each \_

realization and finally sorted to determine elevation \_

exceedance probabilities.

'Tested with the CRITCREK.prj data set. Code would have \_

to be slightly modified to work with other data sets.

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'Keep track of time

Dim timStartTime As Variant, timNowTime As Variant, \_

timElapseTime As Variant

timStartTime = Timer

'Define the n values to use ( will want to define the other parameters to permutate, later )

Dim Nvals(0 To 3) As Double

Nvals(0) = 0.02

Nvals(1) = 0.03

Nvals(2) = 0.05

Nvals(3) = 0.1

'Define the bed slope values to use to calulate new bed elevation values (also do that!)

'Dim S0vals(0 to 3) As Double

'S0vals(0) = 0

'S0vals(1) = 0.0001

'S0vals(2) = 0.0005

'S0vals(3) = -0.0001

'Define the boundary conditions to assign

'Upstream hydrographs assigned to all scenarios

'Dim USBC(0 to 3) As Integer ' or likely a different data type for the time series

'USBC(0) = hydrograph with a quick recession and low intensity

'USBC(1) = hydrograph with a quick recession and high intensity

'USBC(2) = hydrograph with an attenuated recession and low intensity

'USBC(3) = hydrograph with an attenuated recession and high intensity

'Downstream hydrographs permutated for only mild sloped channels (S0vals i=1&2)

'Dim DSBC(0 to 3) As Integer ' if we are assigning a depth above ND, or different data type for time series

'DSBC(0) = Normal Depth

'DSBC(1) = Slight backwater effect

'DSBC(2) = Moderate backwater effect

'DSBC(3) = Strong backwater effect

'Open the HEC-RAS project

Dim RC As New RAS620.HECRASController

Dim strRASProj As String 'HEC-RAS Project

Dim sngWSElev() As Single

'The array of computed water surface elevations for \_

each realization.

strRASProj = Range("C4").Value

'Open the HEC-RAS project

RC.Project\_Open (strRASProj)

'Get all of the geometry into a Geometry Type

Dim typGeom As TypeRASGeom ''\*\*\*Update: changed mSamples.TypeRASGeom to TypeRASGeom

Dim blnGotIt As Boolean

blnGotIt = GetRiversReachesNodes(typGeom, RC)

'Define variables used in the For-Next Loop

Dim lngNumMessages As Long '\*\*\*Update: Changed intNumMessages to lngNumMessages

Dim strMessages() As String

Dim blnDidItCompute As Boolean

Dim strRiv As String, strRch As String, strRS As String

Dim sngNL As Single, sngNCh As Single, sngNR As Single

Dim strErrMsg As String

Dim sngSumMean As Single, sngCompMean As Single

Dim sngAllRandN() As Single

'Loop through each realization

Dim intNumRealizations As Integer

intNumRealizations = 3 '## CHANGED to number of n values we defined (can soft code this later)

'ReDim sngAllRandN(1 To intNumRealizations)

Dim i As Integer, j As Integer, k As Integer, l As Integer

For i = 0 To intNumRealizations

'Get n value to use by reading from the n values you've defined

sngNCh = Nvals(i)

'Apply new n values to geometry

' ## Maybe need to change to reflect those input variables we set up manually

' ## CAN WE INCLUDE the other permutating variables here, too, like slope and boundary conditions?

' ## Try Edit\_GeometricData(), Edit\_MultipleRun(), PlanOutput\_SetMultiple(), Geometry() subroutines

For j = 1 To typGeom.nRiv

With typGeom.Riv(j)

strRiv = .RivName

For k = 1 To .nRch

With .Rch(k)

strRch = .RchName

For l = 1 To .nNode

strRS = .Node(l).RiverStation

RC.Geometry\_SetMann\_LChR strRiv, \_

strRch, strRS, sngNL, sngNCh, \_

sngNR, strErrMsg

Next l

End With

Next k

End With

Next j

'Save the project with new Manning's n values

RC.Project\_Save

'Compute the HEC-RAS project

'RC.Compute\_HideComputationWindow

blnDidItCompute = RC.Compute\_CurrentPlan \_

(lngNumMessages, strMessages()) ''\*\*\*Update: Changed intNumMessages to lngNumMessages

'Get water surface elevation output at River 1, Reach \_

1, River Station 7, for profile 1. The Output ID \_

for water surface elevation is 2.

' ## CHANGE to get the vars we are interested in

' ## CAN WE CALL GetDetailedTS() subroutine

ReDim Preserve sngWSElev(i)

sngWSElev(i) = RC.Output\_NodeOutput(1, 1, 1, 0, 1, 2)

'Show Progress in Excel

timNowTime = Timer

timElapseTime = \_

Round((timNowTime - timStartTime) / 60, 2)

Application.StatusBar = "Finished computing " & \_

"realization #" & i & " of " & intNumRealizations \_

& ". Sampled N Value = " & Round(sngNCh, 4) & \_

". All Samples Mean N value = " & sngCompMean & \_

". Elapsed Time: " & timElapseTime & "minutes."

Next i

'Close HEC-RAS

RC.QuitRAS

'Send the output data to Excel

'Send a message about the runtime to a message box.

timElapseTime = Round((timNowTime - timStartTime) / 60, 1)

MsgBox "Total time: " & timElapseTime & " minutes." & \_

Chr(13) & strOutput & "Computed Mean = " & \_

CStr(sngCompMean)

End Sub

Sub SensitivityScenarios()

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'Demonstrates running HEC-RAS in a Monte Carlo-esque Experiment.

'Written by Emma House, with code from Chris Goodell's scripts.

'May 14, 2024

'This subroutine runs through different parameter permuations (already \_

created flow and geometry files) in HEC-RAS and extracts the detailed \_

data required for the momentum terms analysis.

'Define the parameters and create plan file to run -

' A loop that runs through all permutations and sets up a plan

' With the plan set, the first loop then runs the HEC-RAS project

' Then another loop: for each time step of the simulation, extracts detailed data

' Exits second for loop and moves onto the next plan file

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'Help the code run faster

Application.ScreenUpdating = False

'added to solve the OLE message!

Application.DisplayAlerts = False

'Keep track of time

Dim timStartTime As Variant, timNowTime As Variant, \_

timElapseTime As Variant

timStartTime = Timer

'Instantiate a new RAS620.HECRASController Class

Dim RC As New RAS620.HECRASController

'Define the plan and flow files and temporary files to write to

Dim strPlanFile As String

Dim strNewPlanFile As String

Dim strFlowFile As String

Dim strNewFlowFile As String

Dim strGeomFile As String

Dim strNewGeomFile As String

Dim strRASProject As String

strRASProject = Range("C4").Value '"L:\NSF\_Iowa\_L\RAS\_Toy\Sensitivity\_analysi.prj"

strPlanFile = Range("C5").Value '"L:\NSF\_Iowa\_L\RAS\_Toy\Sensitivity\_analysi.p01" L:\NSF\_Iowa\_L\RAS\_Toy

'strPlanFile = "C:\temp\Sensitivity\_analysi.p01"

strNewPlanFile = Range("C6").Value '"L:\NSF\_Iowa\_L\RAS\_Toy\Sensitivity\_analysi.tempp01"

' Define plan with loop of geometry and flow files we've created

' Geometry files: Sensitivity\_analysi.g01-17 (permutations of values of S0 & n)

' Flow files: Sensitivity\_analysi.u01-03 (same US Q TS with diff DS boundaries)

Dim flowFiles As Variant

Dim geomFiles As Variant

Dim k As Integer

Dim j As Integer

'Define the arrays of flow and geometry files

geomFiles = Array("g04", "g05")

flowFiles = Array("u07", "u08", "u09", "u10", "u11", "u12")

'Loop through each combination of flow and geometry files

For k = LBound(flowFiles) To UBound(flowFiles)

For j = LBound(geomFiles) To UBound(geomFiles)

strFlowFile = flowFiles(k)

strGeomFile = geomFiles(j)

' Get into plan file and rewrite flow and geometry

Open strPlanFile For Input As #1

Open strNewPlanFile For Output As #2

Do While Not EOF(1)

Line Input #1, strLine

If InStr(strLine, "Geom File") Then

Print #2, "Geom File=" & strGeomFile

ElseIf InStr(strLine, "Flow File") Then

Print #2, "Flow File=" & strFlowFile

Else

Print #2, strLine

End If

Loop

'Replace the original flow file with the \_

temporary one, then delete the temporary one.

Close #1

Close #2

FileCopy strNewPlanFile, strPlanFile

Kill strNewPlanFile

'Open HEC-RAS

RC.Project\_Open strRASProject

Debug.Print "Project opened: " & strRASProject

'Run HEC-RAS

Dim blnDidItCompute As Boolean, lngMsg As Long, \_

strMsg() As String

'If HideRAS Then RC.Compute\_HideComputationWindow

blnDidItCompute = RC.Compute\_CurrentPlan(lngMsg, strMsg())

'DoEvents ' allows Excel to process other events (hopefully helps get rid of window??)

' Check if the computation was successful

If Not blnDidItCompute Then

Debug.Print "Computation failed for Flow: " & strFlowFile & " and Geom: " & strGeomFile

RC.QuitRAS

Exit Sub

End If

'Find the number of Profiles (Time steps)

'Declare variables for the River ID and Reach ID. Since there \_

is only one river and reach, ID's can be hardcoded to 1.

Dim lngRiverID As Long, lngReachID As Long ' River ID and Reach ID

Dim lngNum\_PF As Long 'Number of profiles - HEC-RAS Controller will populate.

Dim strPF() As String 'Array of names of the profiles

lngRiverID = 1

lngReachID = 1

RC.Output\_GetProfiles lngNum\_PF, strPF()

'Declare single arrays for all detailed vars (from GetDetailedTS)

'Set dimension of single arrays to the number of profiles

Dim sngWS(1 To 14401) As Single 'Array of water surface elevations

Dim sngAvgVel(1 To 14401) As Single 'Array of Avg Velocities

Dim sngQtot(1 To 14401) As Single 'Array of total flows

Dim sngFlowA(1 To 14401) As Single 'Array of flow areas

Dim sngSf(1 To 14401) As Single 'Array of friction Slopes

Dim sngHydD(1 To 14401) As Single 'Array of hydraulic depths

Dim sngDSWS(1 To 14401) As Single 'Array of water surface elevations

Dim sngDSAvgVel(1 To 14401) As Single 'Array of Avg Velocities

Dim sngDSQtot(1 To 14401) As Single 'Array of total flows

Dim sngDSFlowA(1 To 14401) As Single 'Array of flow areas

Dim sngDSHydD(1 To 14401) As Single 'Array of hydraulic depths

ReDim strPF(1 To 14401)

'Declare variable IDs.

Dim lngWS\_ID As Long 'The Water Surface Elevation ID is 2.

Dim lngAvgVel\_ID As Long 'The Average Velocity ID for the cross section is 23.

Dim lngQtot\_ID As Long 'The total Flow ID is 9.

Dim lngFlowA\_ID As Long 'The flow area ID is 10.

Dim lngSf\_ID As Long 'The friction slope ID is 68.

Dim lngHydD\_ID As Long 'The hydraulic depth ID is 126.

lngWS\_ID = 2

lngAvgVel\_ID = 23

lngQtot\_ID = 9

lngFlowA\_ID = 10

lngSf\_ID = 68

lngHydD\_ID = 126

'Search for profiles and populate the single var, and strRS() arrays

Dim i As Long 'Index for the time For-Next Loop

For i = 1 To 14401 ' 20

'For i = 1 To lngNum\_PF ' doesn't work

Application.StatusBar = "retrieving data for " & i & " of " & lngNum\_PF

sngWS(i) = RC.Output\_NodeOutput(lngRiverID, \_

lngReachID, 20, 0, i, lngWS\_ID)

sngAvgVel(i) = RC.Output\_NodeOutput(lngRiverID, \_

lngReachID, 20, 0, i, lngAvgVel\_ID)

sngQtot(i) = RC.Output\_NodeOutput(lngRiverID, \_

lngReachID, 20, 0, i, lngQtot\_ID)

sngFlowA(i) = RC.Output\_NodeOutput(lngRiverID, \_

lngReachID, 20, 0, i, lngFlowA\_ID)

sngSf(i) = RC.Output\_NodeOutput(lngRiverID, \_

lngReachID, 20, 0, i, lngSf\_ID)

sngHydD(i) = RC.Output\_NodeOutput(lngRiverID, \_

lngReachID, 20, 0, i, lngHydD\_ID)

sngDSWS(i) = RC.Output\_NodeOutput(lngRiverID, \_

lngReachID, 21, 0, i, lngWS\_ID)

sngDSAvgVel(i) = RC.Output\_NodeOutput(lngRiverID, \_

lngReachID, 21, 0, i, lngAvgVel\_ID)

sngDSQtot(i) = RC.Output\_NodeOutput(lngRiverID, \_

lngReachID, 21, 0, i, lngQtot\_ID)

sngDSFlowA(i) = RC.Output\_NodeOutput(lngRiverID, \_

lngReachID, 21, 0, i, lngFlowA\_ID)

sngDSHydD(i) = RC.Output\_NodeOutput(lngRiverID, \_

lngReachID, 21, 0, i, lngHydD\_ID)

strPF(i) = i ' doesn't give the names but gives the profile IDs

'DoEvents ' allows Excel to process other events

Next i

'Create a spreadsheet with time series of output variables.

'Create a unique sheet name based on the current combination

Dim sheetName As String

sheetName = "RASResults\_" & strFlowFile & "\_" & strGeomFile

' Check if the sheet already exists

Dim blnSheetExists As Boolean 'Boolean tag true if sheet exists.

blnSheetExists = False

For Each ws In Worksheets

If ws.Name = sheetName Then

blnSheetExists = True

Exit For

End If

Next ws

' If the sheet does not exist, add it

If Not blnSheetExists Then

Sheets.Add After:=Sheets(Sheets.Count)

ActiveSheet.Name = sheetName

End If

' Select the newly created or existing sheet

Sheets(sheetName).Select

'Clear all the data so that can construct a fresh table

Cells.Select

Cells.Delete

'Name the sheet with selected flow and geometry files

Range("A1").Select

ActiveCell.Value = "Results from HEC-RAS " & \_

"\_" & strFlowFile & "\_" & strGeomFile

'Populate the table headers.

' !! May need to reorder these since we have those calculations set up differently

Range("A3") = "Profile"

Range("B3") = "Tot Q (cms)"

Range("C3") = "W.S. El. (m)"

Range("D3") = "Avg. Vel. (m/s)"

Range("E3") = "Flow Area (sqm)"

Range("F3") = "Frctn. Slope (-)"

Range("G3") = "Hyd. Depth (m)"

Range("H3") = "Len. Chnl. (m)" ' keeping the column open so I can fill it with the val later

Range("I3") = "DS Tot Q (cms)"

Range("J3") = "DS W.S. El. (m)"

Range("K3") = "DS Avg. Vel. (m/s)"

Range("L3") = "DS Flow Area (sqm)"

Range("M3") = "DS Hyd. Depth (m)"

'Loop through the HEC-RAS results arrays and populate the \_

spreadsheet. Water surface values are rounded to 9 \_

decimal places. Velocities to 9 decimal places, too.

'First, declare string values for conversion in the Round \_

function.

Dim strWSEL As String, strAvgVel As String, strQtot As String, \_

strFlowA As String, strSf As String, strHydD As String, strLen As String

Range("A4").Select

For i = 1 To 14401 ' 14401

'For i = 1 To lngNum\_PF ' !! Changed from lngNum\_RS

Application.StatusBar = "tabulating data for " & i & " of " & lngNum\_PF

Cells(i + 3, 1).Value = strPF(i)

Cells(i + 3, 2).Value = Round(sngQtot(i), 9)

Cells(i + 3, 3).Value = Round(sngWS(i), 9)

Cells(i + 3, 4).Value = Round(sngAvgVel(i), 9)

Cells(i + 3, 5).Value = Round(sngFlowA(i), 9)

Cells(i + 3, 6).Value = Round(sngSf(i), 9)

Cells(i + 3, 7).Value = Round(sngHydD(i), 9)

Cells(i + 3, 9).Value = Round(sngDSQtot(i), 9)

Cells(i + 3, 10).Value = Round(sngDSWS(i), 9)

Cells(i + 3, 11).Value = Round(sngDSAvgVel(i), 9)

Cells(i + 3, 12).Value = Round(sngDSFlowA(i), 9)

Cells(i + 3, 13).Value = Round(sngDSHydD(i), 9)

'DoEvents ' allows Excel to process other events (hopefully helps get rid of window??)

Next i

Next j

Next k

' Disable error handling after the loop

'On Error GoTo 0 ' Resume normal error handling

'Close HEC-RAS

RC.QuitRAS

' or kill project (OUTSIDE OF RAS)

' or shell function in VBA

' i.e,

' Sub TerminateHECRAS()

' Dim taskKill As String

' taskKill = "taskkill /IM ras.exe /F"

' Shell "cmd.exe /c " & taskKill,

' vbHide

' End Sub

'Show Progress in Excel

timNowTime = Timer

timElapseTime = \_

Round((timNowTime - timStartTime) / 60, 2)

Application.StatusBar = "Finished computing " & \_

"realization #" & i & " of " & intNumRealizations \_

& ". Sampled N Value = " & Round(sngNCh, 4) & \_

". All Samples Mean N value = " & sngCompMean & \_

". Elapsed Time: " & timElapseTime & "minutes."

'Restore screen updating

Application.ScreenUpdating = True

'added to solve the OLE message!

Application.DisplayAlerts = True

MsgBox "Done!"

End Sub