# Implemented Design Overview

## List of subs

* Select source folder button makes call to Private Sub CommandButton1\_Click
  + open file dialogue
* Source Folder selected displayed Private Sub SetccText()
  + fills in the text field on processing document
* Process files button makes call to Sub ProcessForms()
  + if no source folder is selected Tells user
  + if there are no files in the source folder tells user
  + Only macro available from macro button on developer tab
  + Calls Sub ResultFolderClear
  + Calls loop pass it option to get the file name of each file and load into array global variable rawfilelist.
  + All files in srcfolder are copied to Raw folder and it is created if it doesn’t exist.
  + Calls loop pass it option to do validation check on each file.
    - Calls Sub ValidationChecks
    - invalid files are added to invalid filelist
    - If more than one form no files are processed.
  + move the invalids out to the invalid folder and create it if it doesn’t exists.
  + Calls loop to get data of each remaining file in the filestoprocess folder since only valid files should be there now. As each is processed it’s name is added to global array fileprocessedfilelist.
* Check if results folders exist or not and if they exist are they empty
  + Private Sub ResultFolderClear(ByRef Rdy As Boolean)
  + checks if they exist or not, if not then processing continues
  + If they exist, checks if they are empty or not
  + if not empty processing stops user notified.
* Check the raw files if ready to process
  + Recieves the file name from file loop sub.
  + Private Sub ValidationChecks
  + loops through all files in folder (takes the file and checks)
    - is file a .docx
      * if not it is invalid
    - if it is then open file and look through tags
      * if missing or a space in Tags it is invalid
    - if file still valid are tags unique
      * if first file the load into global variable tagArray
        + Calls sub ElementsUnique
      * if not the first form file is it the same form
        + Calls CompareStringArrays
        + if not same stop processing
      * `

## TODO

Add invalid filename array for use when printing out invalid file list.txt file.

Before validation checks but after results folder checks get list of files in folder. Some of these might be invalid but we need a list of all files seen there to start with and a count of those files.

change to case or byref choices on sub fileloop

'used to refer to a contentControl

Dim CC As ContentControl

'need a long to refer to index in array and looping iterations

Dim r As Long

'When we loop through the valid file we only need to get the record array not the tags.

'array for the record fields

Dim RcrdArray() As Variant

For Each cc In ActiveDocument.ContentControls

'only add to array if placeholder text does not equal range.text

If cc.Range.Text <> cc.PlaceholderText Then

RcrdArray(r) = cc.Range.Text

Else

'else add empty index ("").

RcrdArray(r) = ""

End If

r = r + 1

Next

'loop through Rcrdarray and write each to a string variable with a tab between them.

For r = LBound(RcrdArray) To UBound(RcrdArray)

If r = UBound(RcrdArray) Then

RcrdText = RcrdText & RcrdArray(r)

Else

RcrdText = RcrdText & RcrdArray(r) & vbTab

End If

Next r

MsgBox (RcrdText)

## Coded today

troubleshooting and breaking out the validation checks so we can have tag array populated so we can move back to getting the record data into an output file. then we work on creating the results folders and moving invalids etc.

Think we have that fixed and ready to put only valid files in folder and work on the get record data.

Here are the code in progress working on.

Private Sub getRecordData()

'used to refer to a contentControl

Dim cc As ContentControl

'1-open invisible instance of word

'it will be used to get info from file

Dim wrd As Word.Application

Set wrd = New Word.Application

wrd.Visible = False

'Get the fist file in the source folder

'Loop through each file and this is the name of the file.

Dim strFileName As String

'This set the folder and all files in folder.

Dim strFileSpec As String: strFileSpec = srcFldr & "\\*.\*"

'This will retrieve the name of first file only.

'This doesn't care if extensions are hid for known file types. it gets the extension

strFileName = Dir(strFileSpec)

'Starts the loop through the files.

Do While Len(strFileName) > 0

1-open the file

'varialbe for the current file working on using our invisible word instance (wrd)

Dim file As wrd.Document

'Open the file

Set file = Documents.Open(srcFldr & "\" & strFileName)

2-copy the data to the array

3-loop array and write a string with tab delimiters

4-close the file

5-write to output data file

Loop

'Close instance of word after done looping files

wrd.Quit

End Sub

Tagarray code

If IsEmpty(tagArray) Then

MsgBox ("First form found " & strFileName)

'fill the public array with all the tags on the content controls.

'with #2 validation check of file we have confirmed all content controls have a tag on them.

'TODO Again here redim once, get a count and rediim to that and then fill the array with the loop

'Since didn't use preserver the array is being rewritten each time. bug.

For Each cc In file.ContentControls

ReDim tagArray(t)

tagArray(t) = cc.Tag

MsgBox tagArray(t)

t = t + 1

Next