# Implemented Design Overview

Last date reviewed and updated: 1/10/2020

## List of subs

* Select source folder button makes call to Private Sub SourceFolderSelect\_Click()
  + open file dialogue for folderpicker
* Source Folder selected displayed Private Sub SetccText()
  + fills in the text field on processing document
* Process files button makes call to ProcessForms\_Click()
  + 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 Private Sub GetData()
  + 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.
* Main procedure is the Private Sub GetData()
  + First it determines if results folders are ready for processing files
    - Checks if results folder are present and if so are they empty. it does this by calling Private Sub ResultFolderClear which returns True or false for Rdy for processing to continue
  + Second it gets the number of and name of the initial list of files
    - Uses two procedures to do this.
      * Private Sub FileLoopcnt(ByRef cnt As Long)
        + Can be passed a variable to be populated to allow use of initial files to process, number of invalid files, number of processed files

Global populated rawFilecount

* + - * Private Sub FileLoopNames(ByRef ary As Variant)
        + likewise can be passed different variables to populate with list of filenames

Global populated rawFileList

* + Third TODO Make a copy of all files into Raw folder before opening them.
    - if unable to move or create folder then user should be informed and processing stops. The initial results folder check should cover if user has permission to process files where they are.
  + Fourth it Checks if files are valid for processing
    - Currently implemented in 3 procedures that return Rdy true or false
      * vcIsDocx Rdy
      * vcTags Rdy
      * vcTagsUnique Rdy
        + This populates the global tagarray which will be used for the header row of output file.
    - The current development plan is to move forward with getting record data into an output folder. These three work individually but will not work with all test files in folder. Current testing is to work with only valid forms in folder so we can continue to develop the getrecord data.
  + Fifth TODO: Move invalids to invalid folder
  + Sixth get a new file count and name of valid files to process
    - Reuses the two procedures to do this.
      * Private Sub FileLoopcnt(ByRef cnt As Long)
        + Global populated validFilecount
      * Private Sub FileLoopNames(ByRef ary As Variant)
        + Global populated processedFileList
  + Seventh TODO: loop the files and get the record data
    - Implement capturing record data from all files and put in output file #8
* Private Sub ResultFolderClear(ByRef Rdy As Boolean)
  + Checks if results folders exist or not and if they exist are they empty
  + checks if they exist or not, if not then processing continues
  + If they exist, checks if they are empty or not
  + if not empty Rdy is false and passed to getdata procedure where processing should stop and user notified to check source files.
* Check the raw files if ready to process
  + Private Sub ValidationChecks (initial plan but programming difficulties and need to produce workable software has put this one procedure on hold and 3 individual subs created that will be reviewed for optimization back into one procedure or called from one procedure.
  + Private Sub vcIsDocx(ByRef Rdy As Boolean)
    - loops through all files in folder (takes the file and checks)
    - is file a .docx
      * if not it is invalid
  + Private Sub vcTags(ByRef Rdy As Boolean)
    - Open each file and look through tags
      * if missing or a space in Tags it is invalid
  + Private Sub vcTagsUnique(ByRef Rdy As Boolean)
    - Checks if tags are unique and gets the header row for output file.
      * if first file the tags are loaded into global variable tagArray
        + Calls sub ElementsUnique
        + if not unique erases the array and file is reported as invalid.
        + processing continues to next file.
      * if not the first valid form file is it the same form
        + Compares to tagArray to see if equal
        + if not same stop processing
        + if same file is valid and processing continues to next file.

If any other files are not same form processing stops

## TODO

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

Implement capturing record data from all files and put in output file #8

## Coded today

## Code in progress Scratch

'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)