# Design Document for Word Fillable Forms Data Collection

**Author:** Andrew Charest

**Document Version:** 1

**Date:** 09/27/2019-10/1/2019

**Design Document Approval date: 10/1/2019**

## Overview:

This document serves as the technical reference that outlines the solutions overall design and specific features to meet the requirements as outlined in the Business Requirements document for “Word fillable form data collection”.

## Environment:

Windows PC (Windows 7, 8, 8.1, 10)

Office 2010 (Desktop 2016 and 2019)

Visual Basic for Applications is used and originally built with Office 2010

## Specific Features:

1. The fillable form with content control for field data along with field labels
   1. All data fields will have a field label and a content control for data entry
      1. If original source document or form doesn’t have a field label it is allowable for one to be created to implement best practices. Solution owner will have final say on acceptance of change over existing form provided.
   2. All content controls will have tags.
   3. All tags on content controls will be unique.
   4. Tables will be used for Field labels and content controls.
      1. A field should be able to be moved on the form easily to change look of form but should not impact the output of what is entered in the field.
   5. A document will be created that covers how to create a form with fillable content controls using tables.
      1. This is a specific request for Your Computer First Aid.
         1. Thisis for internal reference when working on client forms.
      2. To customize the forms look should be completed by an advanced user of Word.
         1. Ongoing changes driven by marketing concerns will necessitate a responsible person determining how to proceed.
            1. Having an employee or other staff who is qualified make the change
            2. Request additional service to make specific changes. $65 an hour.
   6. Form will be locked for editing and require a password be entered.
      1. PW: %companyname%123
2. The printable fillable form will not print field prompts
   1. This will be done by hiding the font for the field prompt font in the specific document.
      1. This should be included in the document that covers creating form with tables and content controls.
   2. This should be a copy of the final fillable form.
   3. This form will need to be recreated when the softcopy fillable form is finalized or changed after being finalized.
   4. The printed form will retain same look and feel as existing document should look more professional.
   5. Form will be locked for editing and require the same password be entered.
      1. PW: %companyname%123
3. The master document will have one macro that is available from the macro menu. It will be the same macro that is executed from the button on the document to “process forms”.
   1. Instructions to include.
      1. All files for processing should be placed in a “source” folder.
         1. All files should be the same form. (I)
         2. The first file in the folder will be used as the guidance as to what form will be compared to all other files in the folder. All files that are not the same form are considered invalid. (Logic)
      2. Select the folder where files were placed for processing.
         1. Select the folder using the button “source folder for processing.”
         2. The source folder selected is displayed on the processing form after being selected.
      3. Type of file for output selector Radio with default to .csv file or tab delimited with .txt extension or a .CSV file.
         1. .csv file will open right up in Excel.
         2. .txt file will need to be imported either open all file types and selecting it or using the text import wizard and selecting the file.
      4. Once files are placed and folder selected click “process files” button.
         1. if nothing selected
         2. if no forms in source folder
         3. if other than .doc files in folder
      5. The following folders will be created in the source folder. If they currently exist and are empty, processing will continue. If a file or files exist in them processing will stop and user informed and instructed to move or delete the existing files/folders.
         1. Processed
            1. All valid files processed
            2. Output file to be opened or imported into Excel based on file type selected.
         2. RAW
            1. a copy of all the files placed in source folder for processing.
         3. invalid
            1. All files that are not valid will be placed here with a text file on the issue seen. Missing tags, not a unique tag, different form from first form found %filename% of first file.
         4. Results
            1. The CSV or tab delimited file with all records processed in it.
            2. Results summary file

total files to process

total files processed

total invalid files

combined records file location

* + 1. When processing is done a message will be displayed
       1. Showing summary of results.
          1. Total files to process
          2. Total files processed
          3. Total invalid files
       2. With option to “Open Results folder” with the output file in it or close the dialogue.
          1. open results will display the windows explorer opened to the folder.
  1. A Message box should be displayed once processing is done that shows the folder where the output file is located.
     1. Clicking ok on message should then open the folder location for user to act on.
  2. The delimiter should be able to be specified by the user. .csv and .txt file extensions should be used according ly. .txt for tab delimited file and .csv for comma separated files.
  3. Output file should account for fields being left blank
  4. Field prompt text should not be included in output file
  5. Ideally invalid characters typed into form fields should be gracefully handled but this is not a requirement and can be addressed as the need arises. errors in processing should be seen as invalid files
  6. A file should be placed in the invalid folder that list each file seen as invalid and what check it failed.
     1. missing tag
     2. non-unique tags

## Coding optimization architecture:

1. code optimization will be broken down into the following code blocks
   1. Open file dialog that sets a variable.
      1. This is reused for source folder, invalid folder, processed folder, etc.
   2. Validation checks on files.
      1. Passed a parameter of the source folder.
      2. Each file is checked and is either valid or invalid.
         1. File is a .docx
         2. All content controls have tags
         3. All tags are unique.
         4. First valid file’s list of tags is loaded to array to check rest of files.
            1. Any files not matching on list of tags is invalid.
      3. If file is deemed invalid
         1. The invalid Count # goes up for use in results summary msg and file.
         2. If invalid count is 1 then Invalid file details file created.
         3. Name of file invalid and which check it failed written on next line of Invalid file details.
         4. call to file operations to move file to invalid folder.
   3. File operations
      1. Passed a parameter of the source folder.
      2. Parameter is passed to for specific action
         1. option 1
            1. Checks if processed raw and invalid folders exist in source. if so checks if the folders are empty.

if exist and not empty then stop and report to user

if exist and empty continues

* + - * 1. “Copies all files in source folder to the RAW folder and create it if it is not there.
        2. If any access issues to source file locations they should be detected here and all processing stopped and user informed.
      1. option 2
         1. If file is deemed invalid for processing it will be moved to the invalid folder and creates it if it is not there.
      2. Option 3
         1. Processed files are moved after processing (Reading of file into array) is done and before starting processing of next file.
         2. Moved from the source folder and placed into the processed folder and creates it if it is not there.

The date time of processing appended to the name of the file in the processed folder.

Would need the name of the file passed to it?

* + - 1. Option 4
         1. open the results folder in a windows explorer window.

1. The main sub that triggers the processing of data is the only accessible macro from the document and should do calls to other methods that do the work.
   1. GetData()
      1. counts files in source folder and saves to variable numfilestoprocess for use in results summary msg and file.
      2. calls file operation 1 and makes a copy of all files into raw folder
      3. calls validation checks
      4. output file created
      5. Tags written to output file header line.
      6. file data is read and written to the output files next line
      7. Count of processed files goes up
      8. Call to file operation option 3 to move and rename file with date and time.
      9. all files closed invalid details, output file, Results summary file
      10. results summary message displayed
          1. if option for open results folder selected file operation option 4 called.
          2. Message closed if closed clicked.

## Expected outcomes of processing

1. The source folder should be empty of files once processing is done.
   1. Exception being if there was an access issue to the folder all the files will still be in the source folder.
2. Raw folder should have a copy of all files that were in source.
3. Invalid folder will only exists if invalid files detected.
4. Output file in results folder.
5. All processed files in processed folder with date and time stamp on them.

## Initial coding and starting point

Original code is looking at the same file for the field data and should be used to create the two fillable form documents and as an initial starting point for coding.

Variables used in current sub:

total # of controls

contentControl

Total # of controls missing tags

tag not unique

Data of content control

array for the tags

array for the record fields

string to print/write tag names to header of output file

string to print/write each record to the output file

Long variables for looping and iterations

variable to refer to freefile for file operations

string for the filepath

total # of controls

## Version control

Version control is done in GitHub and location of project is the following.

<https://github.com/TheAZAndyman/Word_Fillable_Forms>

## Implementation Team:

This includes all analysts and developers and company/client responsible party contacts.

### Designer: Andrew Charest

### Developer: Andrew Charest

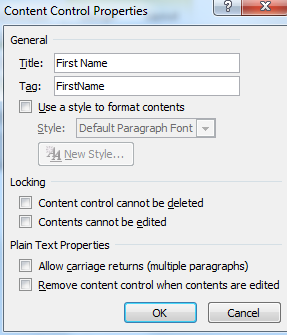
### Business Analyst: Andrew Charest

### Quality Assurance Tester: Andrew Charest

## Glossary:

Any technical terms used in this document that could mean different things or scope for different people and skill levels are to be clarified here.

Tags : Referring to the properties of a content control.



## References:

This is to list out any specifics on resources referenced for education or example codes or guidance used in the developing of the solution.

F**uture**

Allow for specifying what tags should be seen in the form and if not matching the file is considered invalid. Current design uses the first valid file as source of expected tag list and only compares all files have same list of tags all files that do not match are moved to invalid.

If sub folders exist and have files in them user will be prompted as original design but given option to delete/archive the folder. The folder Archive will be created with a subfolder of date and time and existing folders and contents moved to the new archive folder.