Letter Generation

Created by:	Orhan Celiloglu	Reviewed by:	Marek Prazak	Create Date:	17 April 2019

Contents

1. Purpose

- This lesson will introduce you how to generate letters on PDF and DOC formats.
- The lesson should take approximately 1 hour to finish.
- Compared to other lessons, as we are focusing on one use case, the learning material is taking apart different section to give you proper explanation.

2. Definition

- Letter generation is very manual task which can be easily automated. Both formats PDF and DOC are possible on letter generation.

3. Letter Generation

- There are various methods to generate letters such as from a selection, from a worksheet or from a workbook directly. Also, it's possible to fill a word template and generate the letter accordingly.
- Below you'll see syntax for various option to print the letter.

3.1. Letter from an active sheet

- Macro will export content of currently active sheet into specified path as PDF.

```
ActiveSheet.ExportAsFixedFormat Type:=xlTypePDF, _
Filename:="C:\Users\marks\Documents\Saved PDF.pdf"
```

3.2. Letter from an active workbook

- Macro will export content of currently active workbook into specified path as PDF.

```
ActiveWorkbook.ExportAsFixedFormat Type:=xlTypePDF, _
Filename:="C:\Users\marks\Documents\Saved PDF.pdf"
```

3.3. Letter from a selection

- Macro will export selected into specified path as PDF.

```
Selection.ExportAsFixedFormat Type:=xlTypePDF, _
Filename:="C:\Users\marks\Documents\Saved PDF.pdf"
```

3.4. LETTER FROM A RANGE

- From Sheet1, from range A1:H20 export PDF file into specified folder.

```
Sheets("Sheet1").Range("A1:H20").ExportAsFixedFormat Type:=xlTypePDF, _
Filename:="C:\Users\marks\Documents\Saved PDF.pdf"
```

3.5. LOOP THROUGH

- This option will go through each sheet to print the documents

```
Dim ws As Worksheet

For Each ws In ActiveWorkbook.Worksheets

    ws.ExportAsFixedFormat Type:=xlTypePDF, Filename:=ws.Name

Next
```

3.6. SPECIFIC WORKSHEETS

- In this example you'll select Array that contains Sheet1 and Sheet2 and then export active sheet (in this case sheets) as PDF to specified path.

```
Sheets(Array("Sheet1", "Sheet2")).Select

ActiveSheet.ExportAsFixedFormat Type:=xlTypePDF, _
Filename:="C:\Users\marks\Documents\Saved PDF.pdf"
```

3.7. TIPS FOR LETTER GENERATION

- If the Filename **path** is **not provided**, the printed document is **saved** within **the same folder** and with **the same name** as the Excel document.
- If the Filename code **does not contain file path**, but **contains the name**, the document is saved into the **same folder** as the Excel document, but with the **specified name**.
- If the Filename code does not include type of the file (.pdf) at the end, it will use **the last section** of text after the final "\" **as the file name** and **add the .pdf** suffix **automatically**.
- If a PDF with same name already exists in the location, it will be overwritten. It may be necessary include file handling procedures to prevent overwriting existing documents and handle errors.

4. Using Templates

- You can also generate PDF file from template that you have. Logic for this is similar to word's mail merge. In the macro you will create variables that will work as "code words" that will be replaced by the macro in the template and then saved into the PDF.
- **The syntax** (with description) for filling a word template and generating a letter from that template.

Declare your variables

```
Sub mailMerge()

    'Declare variables
    Dim DocLoc As String
    Dim CustCol As Long, CustRow As Long, LastRow As Long
    Dim tagname As String, tagvalue As String, Filename As String
    Dim Worddoc, WordApp As Object
```

Main part of the syntax

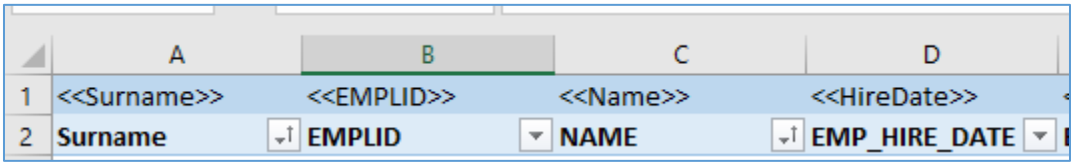
```
'Use With to avoid select
With ThisWorkbook.Sheets(1)

    'Assign object to variable, object is MS word application
    Set WordApp = GetObject("", "Word.Application")

    'Find last row, based on rows in column A
    LastRow = Range("A:A").End(xlDown).Row

    'Assign template to variable, define template location
    DocLoc = "\\ant\dept-eu\" & "\template.docx"

    'Loop through all empoyee rows
    For CustRow = 2 To LastRow

        'Open word from specified location and assign it to worddoc variable
        Set Worddoc = WordApp.Documents.Open(Filename:=DocLoc, ReadOnly:=False)

        'Loop all columns for current row
        For CustCol = 1 To 8

            'Assign current column name to variable
            tagname = Cells(1, CustCol).Value

            'Assign current column value to variable
            tagvalue = Cells(CustRow, CustCol).Value

            'Use feature called Find and Replace(ctrl+H), Field Names in Word replaced with Values from Excel source
            With Worddoc.Content.Find
                .Text = tagname 'what are we looking for
                .replacement.Text = tagvalue 'found text replaced with specified text
                .wrap = wdFindContinue 'find each occurence
                .Execute Replace:=wdReplaceAll 'replace all
            End With

        'loop to next column
        Next CustCol

        'Save filled in word template with specified name as PDF
        Filename = "\\ant\dept-eu\" & "\test.pdf"
        Worddoc.ExportAsFixedFormat outputfilename:=Filename, exportformat:=wdExportformatPDF
        Worddoc.Close False

        'Save filled in word template with specified name as WORD
        Filename = "\\ant\dept-eu\" & "\test.docx"
        Worddoc.SaveAs Filename

    'loop to next row
    Next CustRow

    'close word application
    WordApp.Quit
```

Close the sub

```
End Sub
```

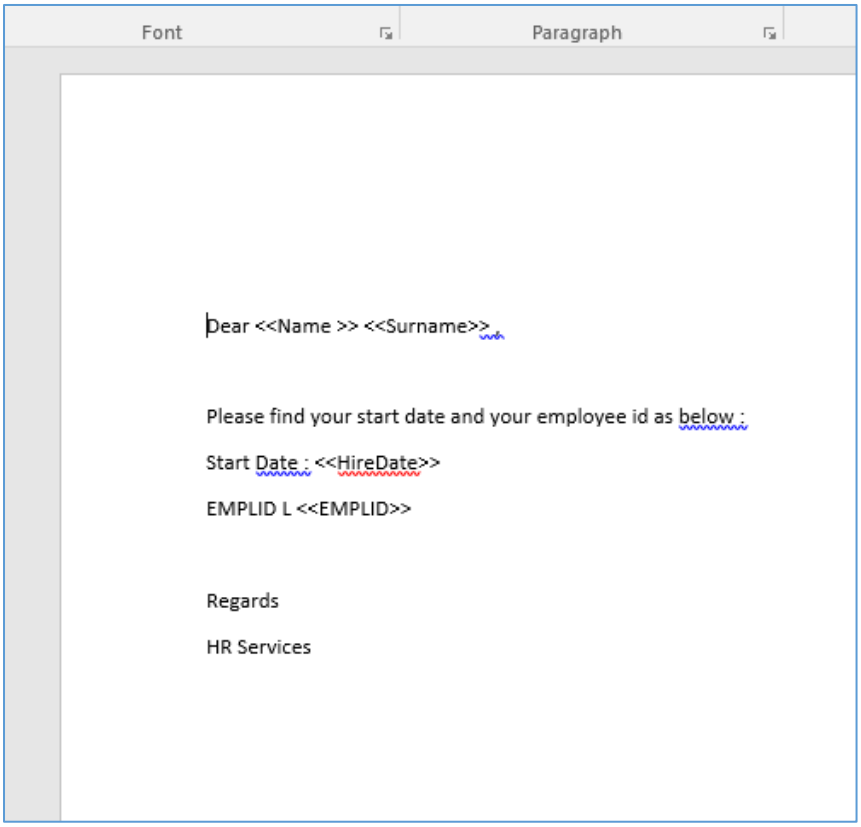
- Below is example of table used for excel mail merge.
- First row contains “the code words” that are placed in Word.
- Second row and further contains the value that will be auto filled in the template.

Note: *This logic can be applied to whole paragraphs as well, which allows you to generate contracts based on pre-set up rules.*



	A	B	C	D
1	<<Surname>>	<<EMPLID>>	<<Name>>	<<HireDate>>
2	Surname	EMPLID	NAME	EMP_HIRE_DATE

- Your word template will look like below.



Note: *Make sure that you create unique code words, as just simply using the word "Name" instead of "<<Name>>" might result in situation where the macro will replace all mentions of "Name".*

5. Closing Exercise

Follow the below instruction to create working letter generating macro.

1) Create macro in existing source file
2) Add button to start the macro
3) Update existing word template to contain todays date that would automatically update itself.
4) Use the excel source file for the mail merge, which would print out as word(.docx) and save them automatically into folder named letters.
5) The mail merge will update Street, City, Country, Manager name, Effective date, Employee name.

6. Additional knowledge

- Save excel file as PDF - https://powerspreadsheets.com/save-excel-file-pdf-vba/
- Filling Word templates - https://www.makeuseof.com/tag/integrate-excel-data-word-document/