

Final Project

Calendar Word Conversion

Name: Laura Cutler

Date: 12/11/2014

Executive Summary

Business Overview

Vertex42 LLC creates calendars, which it provides via download on its website vertex42.com. The calendars are attractive, easy to customize, and even include holidays; these features make them some of the most popular calendars on the internet. The site appears among the first internet results when someone searches for “calendar template” on Google or other search engines.

Currently, all of the calendars available on Vertex42.com are Excel files or PDF’s. However, customers have recently expressed a desire to have Word-based calendars. Many of them feel more comfortable using Word than Excel, and they would like the customizability that is available through Word over the static PDF versions.

Desired Outcome

It would be possible for employees of Vertex42 LLC to copy the existing Excel calendars available online into Word documents and format them in that application. However, this process would be time consuming and repetitive for a single year, and would be unreasonable to execute for multiple years.

Because the website offers many different kinds of calendars for almost any year, the company hopes to be able to generate the Word calendars quickly and accurately.

Accordingly, the CEO of Vertex42 LLC asked me to expand the functionality of the company’s Excel-based calendar generator to also produce calendars in Word. He provided me with the Excel workbook company employees use to create Excel-based calendars, which has formulas that compute the day-of-the-week and holidays. Working from this foundational Excel workbook, I was to build new functionality that would enable the resulting calendars to be transferred directly into Word.

Solution Overview

The solution spans both applications. The VBA code I wrote is all held within the Excel workbook and relies on the workbook’s existing formulas. It opens an existing Word template and generates individual calendars for the twelve months of a user-specified year. These calendars are each copied into the Word-based template and are formatted to present a clean layout, responsive to changes in page size or orientation.

This solution will enable Vertex42 LLC to quickly generate 12-month calendars for any number of years. The company will provide these calendars online, giving customers the flexibility of downloading Word-document calendars instead of Excel-based calendars. I will make the code available to the company, so future employees can make small changes to the formatting and generate additional calendars with a different look and feel, to appeal to a large and varied customer base.

Implementation Documentation

The main functionality of this solution is in Excel. Formulas, formatting, and macros are all used to generate monthly calendars. However, the deliverable product for this solution is created in Word. Therefore, both applications need to be opened and accessible for this solution to be successful.

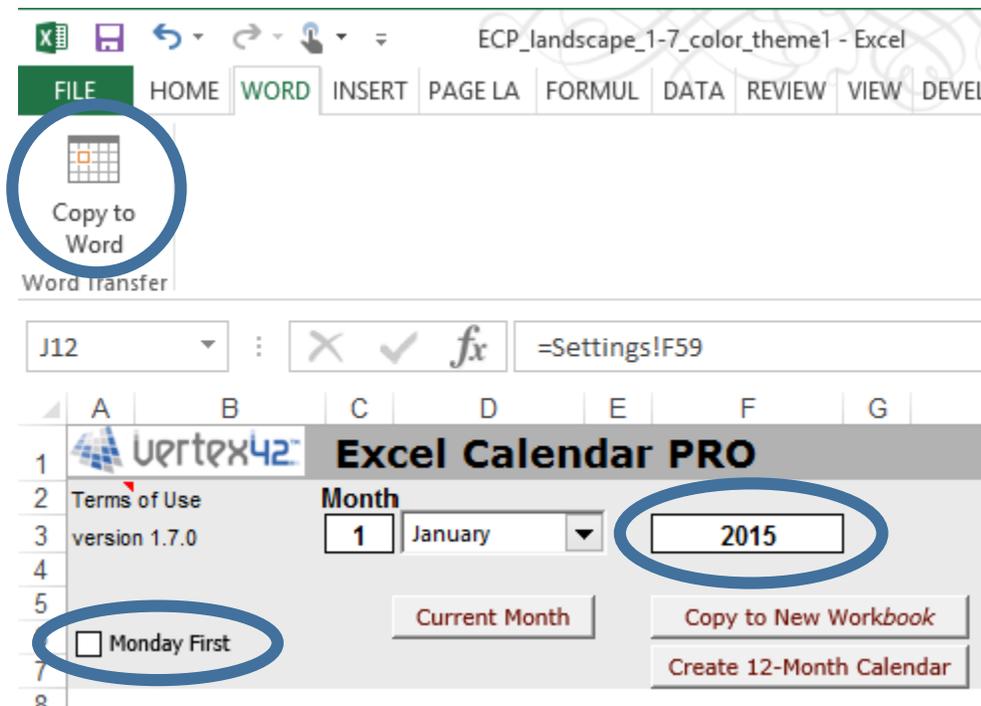
To run the code, I first open the Excel workbook that contains the VBA solution and workbook formulas. I then select the “Copy To Word” function to begin the transfer process.

The copying process involves working with both applications, adding a single-month calendar and title, formatting the calendar and title, and repeating the process for 12 months.

Transfer steps summary

The following screenshots demonstrate three stages of the copying and formatting process.

All of the functionality of the transfer process is accessible via a single button on the Ribbon. The user first updates the year in cell F3. Next, the user will activate the transfer process by clicking the “Copy to Word” button on the “WORD” tab of the Ribbon. An additional feature is the capability of employees to create calendars with Monday first, rather than Sunday. This dashboard is displayed in the following screenshot.



Adding a month calendar

This screenshot shows the calendar calculator in Excel. Cell F3 holds the year for which the calendar will be generated. Once the program begins, the code will update the month integer held in cell C3, and formulas within the sheet will update the resulting calendar.

Within the Excel workbook, I generate a single monthly calendar at a time, beginning with January of the user-specified year. I then copy both the title and the calendar to Word and format them.

The screenshot displays the 'Excel Calendar PRO' application interface within an Excel spreadsheet. The interface is organized into several sections:

- Title Bar:** Shows 'Vertex42 Excel Calendar PRO' and '© 2014-2015 Vertex42, LLC'.
- Menu Bar:** Includes 'Update Colors'.
- Control Panel:**
 - Month:** A dropdown menu set to 'January' and a text box containing '2015'.
 - Color Selection:** A color bar with 'Hue' and 'Saturation' sliders.
 - Buttons:** 'Current Month', 'Copy to New Workbook', and 'Create 12-Month Calendar'.
 - Monday First:** A checkbox that is currently unchecked.
 - Title:** A text box labeled 'For the Document Name' containing '2015 Calendar'.
- Calendar Content:**
 - Title:** '[Calendar Title]' followed by 'January 2015' in a large blue font.
 - Grid:** A calendar grid with columns for days of the week (Sunday to Saturday) and rows for dates. The first row shows '1 New Year's Day', '2', and '3'. The last row shows '25', '26', '27', '28', '29', '30', and '31'. 'ML King Day' is noted on Monday, January 19th.
 - Summary:** Small calendar thumbnails for 'December '14', 'February '15', and 'March '15' are visible in the top right.
- Notes Section:** A 'Notes' field at the bottom right contains the text: 'Calendar Templates by Vertex42.com', 'http://www.vertex42.com/calendar/', '© 2015 Vertex42 LLC. Free to print.', and two links labeled '2015 Calendar'.

Opening the Word template and pasting the title

The code first performs the following functions, to ensure both applications can work together.

- Open a blank calendar template in Word, which I have previously created
- Assign variable names to the Word application and document so they can be referenced throughout the code
- Display the Word template so I can see the final product update as my code executes

This screenshot shows the Word document immediately after the title has been pasted. It is part of a single-cell table, is a large font size, has large spacing before and after it, and is left-aligned.



Formatting the title and pasting the table

The following screenshot shows the Word document immediately after the calendar has been pasted. The title has been converted to text and formatted with the appropriate spacing, size, color, and alignment.

To copy and format the title, I perform the following steps.

- Copy the title and paste it in the Word template
- In Word, remove the spacing and tabs after the title
- Center align the title
- Shrink the font size so it will fit on one page
- Adjust the font color

However, at this point in the code execution, the table has not been formatted. It spans two pages and has inconsistent alignment.

The screenshot displays a calendar for January 2015. The title "January 2015" is centered at the top in a blue font. Below the title is a table representing the calendar grid. The table is split across two pages. The first page shows days 1 through 24, and the second page shows days 25 through 31. The table has inconsistent alignment and spacing, particularly in the second row of the second page where the text "Calendar Templates by Veeva42.com" is visible.

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
				1 New Year's Day	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19 ML King Day	20	21	22	23	24
25	26	27	28	29	30	31

2018 Calendar Template © 2014 Veeva42 LLC <http://www.veeva42.com/calendar/2018.html>

Calendar Templates by Veeva42.com
<http://www.veeva42.com/calendar/>
© 2018 Veeva42 LLC. All rights reserved.
2018 Calendar 2018 Calendar

Formatting the table

The final screenshot shows the Word document once the table has been formatted.

To copy and format the monthly calendar, I do the following.

- Select the calendar in Excel, copy it, and paste it in the Word template
- Within Word, select the table for formatting
- Center align the header row, left align the other cells
- Auto-fit Table so it fits horizontally inside the Word document
- Delete one row so the calendar will fit on a single page

At this point, the calendar fits on a single landscape page, has the date in the upper left corner, and shows each day of the week as a title centered above its appropriate column.

January 2015						
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
				1 New Year's Day	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	18 MLK Day	20	21	22	23	24
25	26	27	28	29	30	31
		Notes	Calendar Template by Vertex42.com http://www.vertex42.com/Calendar/ © 2015 Vertex42 LLC. Free to print. 2015 Calendar 2015 Calendar			

2015 Calendar Template © 2015 Vertex42 LLC <http://www.vertex42.com/Calendar/2015.html>

Providing a full year

Before running the code, the user simply types the desired year into the Excel workbook. The code repeats for each month in the year, generating each calendar in Excel and adding it as a new page within the Word document.

The finished product is a Word document with 12 pages, each containing the title and calendar for a single month. (See the image below for an example.)

Learning

While developing this solution, I learned important lessons about to apply the principles of VBA programming in Excel to the Word application. Two important lessons were the translation techniques between Word and Excel variables and different text selection methods.

Necessity of Word enumerations

Several properties in Word are typically specified in VBA with codes such as `wdAlignParagraphLeft`. However, these code names only work if the VBA code is run from Word, not from Excel. So for each of the Word-specific properties I used in my formatting, I had to find the integer value it represents and declare that as a variable within my VBA code. Here are two examples of variables that I assigned the same name as their Word counterparts, but had to specify what their numerical value should be, whereas in the Word version of VBA those declarations would be redundant.

```
Const wdAlignParagraphLeft = 0
```

```
Const wdAutoFitWindow = 2
```

Methods of text selection

One of the simplest steps in an Excel function is selecting text. With commands such as `Cells`, `Range`, and `Mid`, the desired text can quickly be identified and processed as desired. Notably, Word does not have such convenient text selection methods. For the purposes of this solution, the most effective way to select the title after pasting it in Word involved a multistep process, which I have outlined below, including code snippets. This particular process begins after the title has been copied from Excel.

- Move to the end of the current paragraph

```
wdApp.Selection.EndKey Unit:=wdStory, Extend:=wdMove
```

- Use a variable to hold the current location, which will be the beginning of the title

```
rngFrom = wdApp.Selection.Start
```

- Paste the title

```
wdApp.Selection.Paste
```

```
Application.CutCopyMode = False
```

- Use a variable to hold the location of the end of the title

```
rngTo = wdApp.Selection.End
```

- Select the range between the beginning and end of the title

```
wdDoc.Range(rngFrom, rngTo).Select
```

These lines of code are necessary to select a small range in Word, whereas a single simple line of code in Excel could accomplish the same result.

It is important to note that the selection property is used in the application level, not the document level, of Word. So if a the document that one thinks is selected is not actually the active document, the text selection and manipulation will be performed on the incorrect document.

Difficulties

As I learned more about VBA in the course of this project, I also faced several difficulties. One was an important decision about how to approach the project, and another was the inability of VBA to provide a desirable feature.

Role of the template

One other difficulty that arose as I prepared this solution was the difficulty of choosing the best method of transferring data between the Excel document and a Word document. Originally I wrote a solution that relied heavily on the Word document: the Word template had bookmarks for each date, and the text of the bookmarks would be replaced with the text in the Excel document. However, this approach was not scalable, because the Word document required over 65 bookmarks for a single month. It was also more prone to error, because of the complications around defining ranges and selecting text via VBA in Word.

I also considered a solution that would create a new word document each time the procedure was run, formatting the page orientation, margins, footer, and file properties, and finally saving the document. However, Vertex42 LLC's calendars required a more flexible approach. They often span multiple years, and having employees run the procedure the desired number of times is the simplest approach. Furthermore, the calendars frequently feature headings, decorations, or other content that would be easiest for employees to modify from an existing example, rather than to recreate from scratch.

Impossibility of using theme colors

The most disappointing part of this project was the realization that VBA is not compatible with theme colors. My ideal solution was to apply theme colors to everything in the Word document. I envisioned having one color for the title and another for the background cell color of the weekends. This way, customers would be able to change the look of the whole yearly calendar simply by choosing a different color scheme from Word's built-in theme functionality.

I tried many different ways of implementing this functionality, but none of them were successful. After hours of internet searches and many different tries, I learned that it is not possible to assign theme colors as the font color within VBA code. I can give a section of text the same color as a specific theme color, but that text's color is static and will not update if a different theme is chosen. This incompatibility was surprising because themes have been an integral part of Word and Excel for years. Unfortunately,

the advantages of quickly changing an entire document's coloring is not possible if those colors have been assigned via a VBA procedure.

Assistance

I personally wrote the code that generates multiple calendars in Excel and transfers them into Word. Vertex42.com created the original Workbook, with its holiday calculations and single-month output.