Syllabus Synthesizer

By Dan Burner

VBA Project for ISys 540, Dr. Allen

Executive Summary

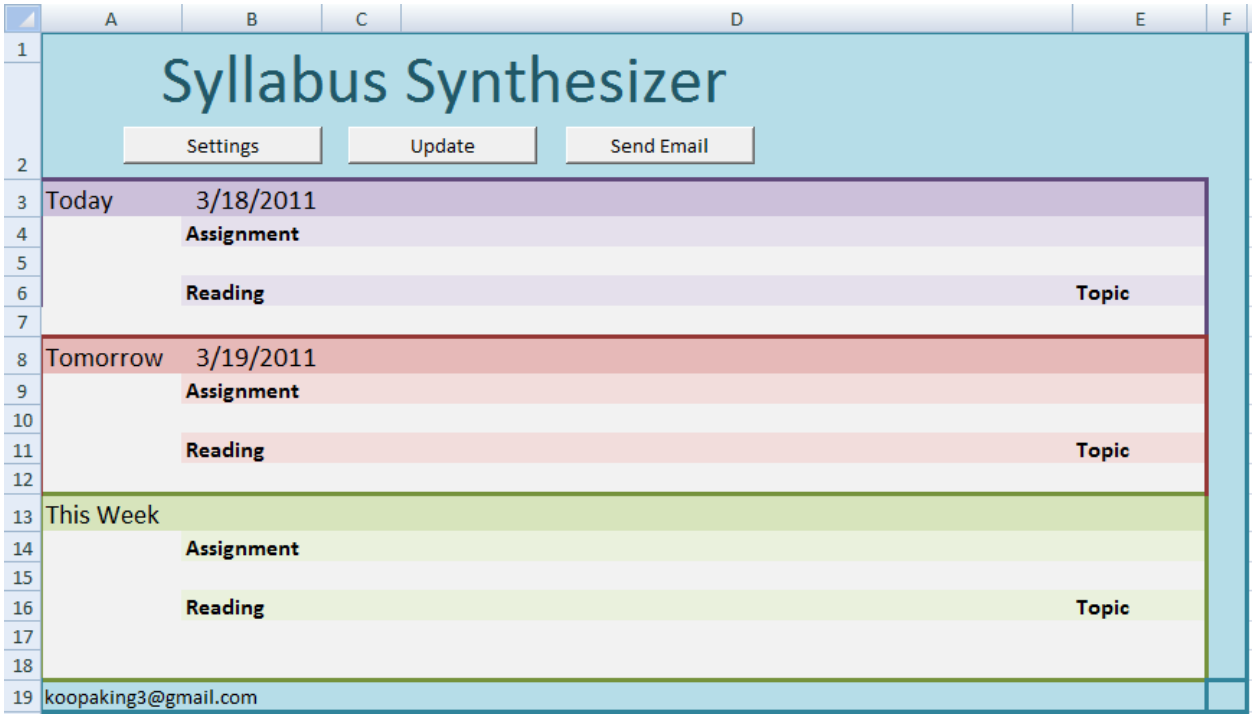
Every semester each student gets a syllabus from each of their professors. Usually this is 4-6 syllabi per student. Nowadays, a syllabus can be found online, in a Word document, or (rarely) just on a piece of paper. Some students manually create their own synthesized schedules of all the assignments found in all theirs syllabi and some just fly by the seat of their pants, trying to remember all five class assignments at once. Students need a central source of organization for syllabus data, specifically what the assignments are and when they are due.

Syllabus Synthesize takes Syllabi from three different sources: the BYU Syllabus builder website, a Word document, or just something pasted or typed into Excel. The program takes these syllabi through a user interface and asks the user to identify certain attributes about the syllabi. After that the user need enter in no more data; the program will do all of the work. The program puts the syllabi into their own separate sheets. Then, the program will take all the data from each sheet and synthesize it in the Home sheet. This synthesizing pulls all assignments and readings due today, tomorrow and throughout the next week and organizes them.

The user clicks the Update button in order to update information on the various syllabi sheets into the home sheet. The user can make changes to the syllabi for each class if the professor changes a date or something. The user can also delete syllabi sheets from the user interface. The program sends out email reminders for assignments that are formatted much the same way as the Home sheet.

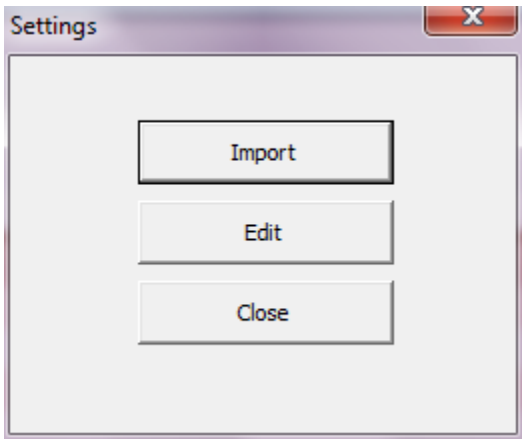
Documentation

In this section, I will describe the details of the project. I have organized it by subsections that each represents an action that can be performed from the UI. The description will detail what is going on behind the scenes to make that interaction work. Below is a screenshot of the UI with labels referencing the sub section.



1. Home

The Home sheet is where all the magic is displayed. All actions are launched from the



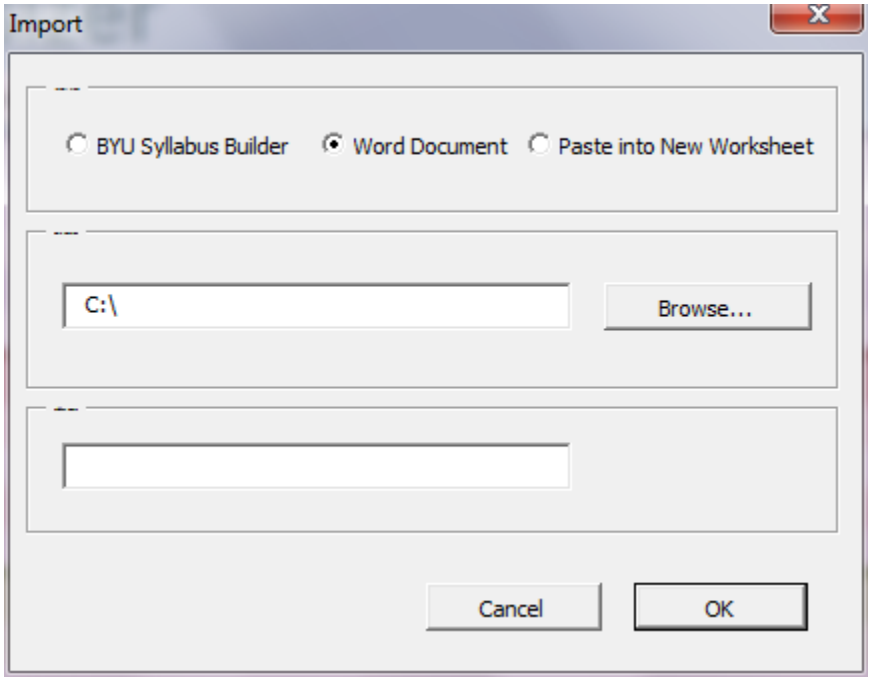
2. Settings

The settings button launches another form with three buttons on it: Import, Edit, and Close. Close just closes the form; import and edit are described below.

2.1 Import

When the user clicks import, another form is made visible. The user has three options represented by radio buttons: Import from Syllabus Builder, Import from Word, or Paste. When the Word option is selected, the Browse button is enabled, allowing the user to click it and get a file finder window and search for the Word document. If the Internet option is selected, the Browse is disabled and the text field is populated with http:// for ease of use in putting in a web address. If Paste is selected, none of this is available.

In the next field the user puts in the name that they wish to call the syllabus.



2.1.1 OK

When the OK button is clicked, the program checks for any errors or missing or duplicate names. Next the program creates a new sheet, names it, formats it for text and runs a sub procedure that will get a table from a Word Doc or from the Internet, based on the option selected.

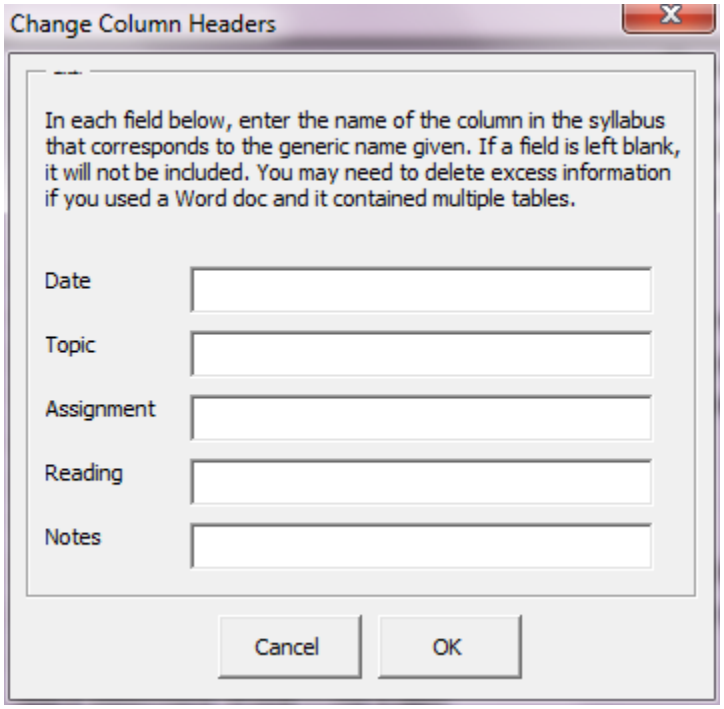
The program uses the Web Query Wizard to go to the provided website and pull the schedule table. Since BYU Syllabus Builder websites are created the same for each class, the program always looks for the same table and copies it straight into the new worksheet. The program then formats the data by removing extra white space and any extra text included in the table (like the Export link that the tables all have). Some times when the table is copied, extra line breaks create blank cells. The program removes these cells and shift the data to be aligned as it should be.

If the program is importing from Word, then it will launch an input box where the user must indicate where the table is located in the document. This was needed because often syllabi have more than one table and there is no way to figure it out independently. The program then creates a Word object,

makes Word not visible, finds the table and selects it. Once it is selected, the program removes extra line breaks then copies the table and pastes it into the new sheet.

When table is copied into the new sheet, the program auto formats it by removing borders, resizing and auto fitting text to the cells. Then, it shows the Column Name form.

2.1.2Column Naming



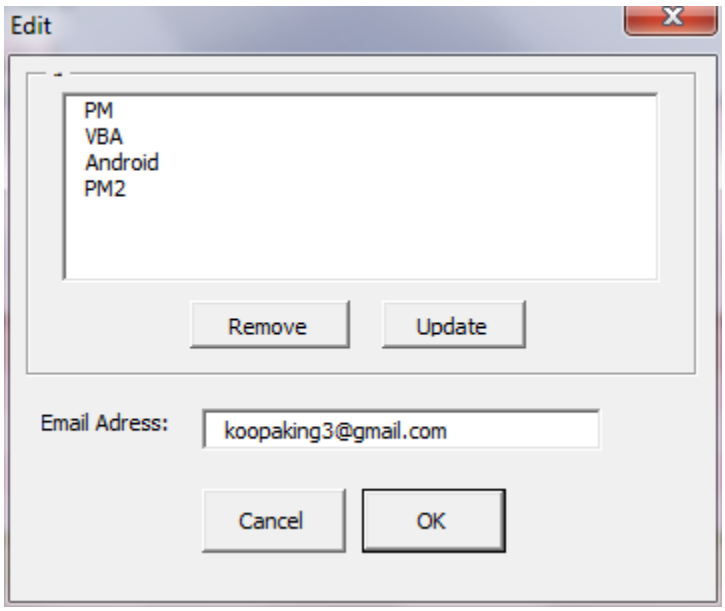
The Column Name form requires the user to enter their syllabus' version of the default column header names that the program requires. I had to do this because syllabi have many different column names that are different depending on what the teacher decides.

Once the user puts in their versions of the column names, they hit OK. When OK is hit, the program cycles through the column names in the sheet and the column names provided and replaces the old with the new. A Date column is always required, so if none is included an error message is displayed.

Since we want all the dates to be in the same format for when we synthesize them later on, a sub procedure is run that formats the dates in the Date column. First, the program replaces any months with text (like "Jan"), with the corresponding number. Then, an array is created with all possible values that can be in a date. If any values are in the date cell that are not in the array, then that value is removed.

After the column naming, the program formats the dates so that they all use the same standard format.

2.2 Edit



2.2.1 Remove/Update

Selecting a syllabus from the list and clicking remove with delete the sheet (the user cannot delete the sheet by right clicking it and selecting delete. I have disabled this). Selecting a syllabus and hitting update will take you to that sheet (activate it).

2.2.2 Email

Here the user enters their email address. When the OK button is hit, the email address in the hidden template sheet will be update so that it can be copied every time the Home sheet is updated. The program places the email address in a specific cell so it can reference it later.

3. Update

This button does not show any visible form but grabs data from syllabi sheets and organizes it in the Home sheet.

The program starts by copying a template of the Home sheet to the Home sheet so that we can get a fresh start each time it is updated. The program then starts by cycling through each of the syllabi sheets. For each sheet it finds any dates that are the same as the current day. Then it selects the columns for Assignment, Reading, and Topics for the row associated with that date. It then copies that data to the Home sheet into the appropriate place under “Today.” At the same time, the program copies the data to a variable that holds HTML formatting that will be sent in the email later. This process of cycling through each sheet and looking for a specific date is repeated for “Tomorrow” and “Next Week.”

The place where an assignment needs to be placed on the home sheet is always changing because the number of assignments for any given day is variable. So, each of the three sections has tow counter variables that keep track of where new information is placed and pass that information to the next section so that the next section knows where to start placing its own data.

If there are any errors in this processing, a message box will appear showing the error and showing which worksheet contains the error so that the user can fix it. Finally, the data is formatted for alignment and auto fitting.

3. Send Email

When the send email button is hit, the HTML message is assembled from the parts created during the Update and some static headers and other text. Then it calls a function that sends an email through Syllabus.Synsthesizer@gmail.com

Lessons Learned

The most trouble was placing the assignments in the right cells on the Home sheet after cycling through all of the syllabus sheets and pulling out the right information. Since the number of rows is always variable I had to devise my own system of references that would show each section where to start. This was a lot of work to figure this out so it would work in any case (empty assignments, tons of assignments and everything in between). So I learned a lot about moving about a sheets and how to make my program dynamic.

I also had a decent amount of trouble importing tables from Word. Creating the Word object was easy enough, but many documents have extra white space that I didn't want and would mess up the dates lining up with their assignments in the syllabus sheet. So I devised a way to replace new line characters. Formatting the table correctly in Excel also proved to be difficult, but I learned to just import it with special paste and then remove all formatting it tried to impose. I learned a lot about string parsing.

Finally, formatting dates to all be recognized by dates took some work to figure out. I ended up using white listing to get rid of characters I did not want and replacing certain characters so that they would be in a standard date format.