

Executive Summary: BYUSA Blue Crew Email Processing

Business Overview: An Introduction to BYUSA Blue Crew

For this project, I choose to do a little volunteer work for an old mission companion, Bryon Anderson. Here's currently serving with BYUSA Blue Crew and requested my help with a project a little earlier this semester. BYUSA Blue Crew is an organization devoted to providing BYU students with an opportunity to get involved by volunteering at whatever events student leaders may need help with. BYUSA is BYU's student service organization and is similar to student governments at other universities although there are a few differences.

Problem Description: Email Processing

For some time now, Bryon and others involved with BYUSA Blue Crew have had to do a lot of monotonous work to keep their hefty email list regularly updated. This involves downloading new changes from the internet, formatting those changes into a .txt file, making some changes to the text so that their bulk mailer can correctly process the information, and then uploading it to Listserv (Listserv is a bulk emailing service used by a number of organizations on campus to communicate with student participants).

Implementation Documentation: Process and Solution Descriptions

I began by sitting down with Bryon and going through the entire process of adding participant subscribers to their list from end to beginning. We began by going over the process students go through to make a submission, the way that the submission arrives to Bryon, how he downloads and formats the submission, and how he then uploads the changes to Listserv. Below is a breakdown of the process:

1. Student submits contact information to BYUSA website
2. System sends an email to the Blue Crew email with the updated email.
3. Bryon grabs the new information and pastes it (piece by piece) into a text file
4. Bryon formats the text file (including adding brackets around every email) and
5. Bryon uploads the updates to the Listserv server
6. Bryon also keeps a copy for himself of the master list (downloadable from Listserv) as a backup.

Thankfully some changes were recently made to the BYUSA website that make it easier to download the updated changes in batches—that greatly bettered the process!

After going through the process and determining what parts could be most aided by automation, I determined that the most helpful part would be helping with step number 4 as it took by far the most time for him. I also wanted to help with 3 and 5, but 3 was already partially completed by the recent system update and both would require access to APIs that went well beyond the scope of the project. At some point in the future someone could consider accessing and studying the Listserv API to upload the updates, but I don't know that it's worth the effort at this point as the task is completed only once or twice a month and takes less than a minute to complete.

I then set to work creating the attached program which has several main pieces as shown below:

1. A GUI for accessing the programmed macros. This includes:

- a. **An introductory page** with explanations and warnings about how to maintain the integrity of the macros
- b. **7 buttons on the Review tab of the ribbon:**
 - i. **Add Brackets/Quotation Marks:** In the event that something were to fail within the macro due to human error or alterations, I wanted to leave them with something that they could use to do the most tedious portions of their tasks. These buttons enable them to add brackets or quotations marks almost infinitely to a continuous column of data
 - ii. **Show/Hide Folders:** To create a backup of the data, I created hidden folders in the folder that I'll give to them where the system automatically saves backups. To make this more user friendly while still trying to maintain integrity, I placed these buttons here so they could more easily access the backups should they need them while having them "hidden" out of harm's way for most of the time.
 - iii. **Download Submissions:** Because of the limitations of the BYUSA website I ended up settling with a button that opened up the website where updates can be downloaded. While this doesn't save them a lot of time, it does help them to have all the pieces of the process in one easy to access place.
 - iv. **Process File:** This button does the bulk of the work that I completed. I'll discuss its purpose in greater detail below.
 - v. **Upload Emails:** Again, like with the 'Download Submissions' button, it would've been impossible to access the necessary portions of the website using the limited capabilities of the embedded Excel browser and likely would have required further implementation using Listserv's API. However, I placed this button here to facilitate the process as much as possible. It loads the necessary webpage for the upload where the user can enter the necessary password and then upload the needed file.
- c. **The actual code:** Here's a breakdown of what is actually in the code, sub-procedure by sub-procedure
 - i. **processFile:** While the meatiest sub-procedure of this project is not very long, it does a lot of work by calling upon a number of other various sub-procedures to do most of its work. See the "Discussion of Learning" for more elaboration on this idea.
 1. Opens the downloaded workbook for formatting
 2. Formats the workbook for two separate uses (viewing by Blue Crew leaders and uploading to Listserv) and saves a copy of each of these in the hidden backup folders. Another copy is placed in the main folder for use by Blue Crew leaders.
 - ii. **Assisting Sub-procedures:**
 1. **cleanPhoneNumbers:** After processing *removeChars*, this sub-procedure formats the selected data according to the standard US phone formatting (i.e. (555)555-5555). In my test data, it caught all but one phone number which couldn't be formatted as the number contained superfluous digits. I bold such numbers to easily show the user that the information is incorrect.

2. **removeChars:** Meant for use primarily with cleaning up the formatting of the submitted phone numbers, I use this sub-procedure to initially clean out most common non-number characters to prepare the phone numbers for further formatting.
3. **processTXTFile:** This sub-procedure goes through the formatting process needed for the files to be uploaded to the Listserv server. This entails removing superfluous columns (everything but name and email) and then adding brackets around the emails.
4. **processXLSXFile:** This sub-procedure goes through the formatting process needed for the files to be used for viewing by student leadership. This entails removing superfluous columns (everything but name, email, and formatting) and then making the workbook a little more usable by adding frozen and bolded headers, cleaning up phone numbers, bolding incorrect phone numbers, etc.
5. **freezePanels:** Provides an easy way to freeze rows and columns. Used to freeze the top row in the *processXLSXFile* sub-procedure.
6. **addSurroundings:** Allows a user to dynamically dictate what they want to surround given text with. I made this to facilitate the process of surrounding email addresses with brackets or names with quotation marks.
7. **addBrackets/QuotationMarks:** Extensions of *addSurroundings* for use on the ribbon. This is one of the most time consuming processes that they perform so I included it on the ribbon should they need it elsewhere.
8. **saveAsTXTFile:** Used to easily save a copy of a file as .txt. See the section below for more discussion on saving.
9. **saveAsXLSXFile:** Used to easily save a copy of a file as .xlsx. See the section below for more discussion on saving.
10. **deleteFile:** Although it's unused in the project at the current time, I created this method based off of some online code to more easily fully delete a given workbook should the need arise (the original project called for it)
11. **openWebsite:** I used this to facilitate creating the buttons for opening the downloading/uploading websites for the project. They also employ message boxes with further instructions for the users about how to proceed on the website.
12. **hide/showHiddenFolders:** As discussed above I included hidden folders for use in this project to store backup data in a hard-to-reach place. I placed buttons on the ribbon to show this data more easily when needed.

See the section below for a greater discussion of what specific areas of knowledge I needed most to study to complete this project.

Discussion of Learning: What I Learned Doing This

I feel like overall there was a lot that I learned completing this project. I'll split it into two groups: (1) principles and (2) particulars. The first group will cover overarching principles that I feel I came to better understand while completing this project, things that could apply generally to any coding project, regardless of language, problem, or logic. The second group will describe briefly the particular pieces of programming that I needed learn to complete the project, some of which I may not have fully implemented or completed learning, but at least touched on during the progress of the project.

Principles

1. **Use methods whenever possible.** I definitely came to see a little more clearly the benefits of using methods, especially ones that are not hard-coded for the project at hand. It often means making the main sub-procedure of the project clearer to understand, reduces reduplication of effort and can allow you to expand your personal library of methods to make future projects easier. I love the feel of creating a method that I know I can call upon to do specific pieces of work. During this project I actually began assembling my own library of sometimes useless, but fun methods just to reinforce this idea.
2. **Don't get scared by hard things.** Although I didn't end up researching and implementing the API necessary to work with Listserv, the experience of realizing that I *could* do that made me realize that in programming sometimes it just takes making the choice to do the slightly harder thing that will enable us to become master programmers. We can't be too afraid of what we might face and need to be willing sometimes to just hash it out for a few hours in order to be successful in the end. The difference I think is just being willing to endure those extra few hours of pain to be able to achieve the end goal. It was really empowering to realize that and I want to be more careful to allow myself to take the time necessary in the future to do the hard thing.
3. **The unique importance of recording macros in VBA.** This was huge. I'm used to just looking things up on the internet when I had questions, a principle that has taken me some time to learn, but when it comes to VBA, the very best tool that you can use is the ability to record macros. When I realized this during the project, things really started speeding up. The code that it records, strangely enough, is not always one-for-one the code that you need to use (yeah, that was weird to figure out), but it can be extremely helpful and is one unique and honestly really awesome aspect of VBA that doesn't really exist elsewhere that I've seen. One other aspect of this that was helpful was learning to type into a cell the next thing that I was attempting to do—once I started doing this it made it even easier to unravel what was actually happening in the code. Really an awesome thing to understand.

Particulars

1. **Creating Ribbon buttons.** This was a great thing to learn how to do—the custom UI tool was really great. I spent a few hours one night (while watching the new Spiderman movie) playing around with the tool trying to figure it out. The clincher came when I realized that they actually have a tool to see whether or not you've made mistakes. Before then I couldn't figure out why things weren't showing up.
2. **Saving As and Saving.** Figuring out how to work saving was a task. It wasn't as straightforward as I initially thought. After I had figured out how to save, I turned it into two unique methods for further use.

3. **ThisWorkbook and Workbook objects.** One of the things that frustrates me about VBA is the ambiguity that sometimes crops up (i.e. the difference between Range and Cells). It took me a little while to figure out exactly how thisworkbook seems to work and how to use statically created workbook objects to keep track of where I'm actually doing things. I made a few mistakes here and deleted (unalterably) a few things that I just had to totally redo because I thought I was working in one workbook and was actually in another. A major part of this was learning to remember that ranges, workbooks, and worksheets are all objects and that you have to use set when defining their values.
4. **Dynamic naming conventions for files.** This was a fun part of the project, learning how to create file names based upon changes in dates and times. See the next item.
5. **Hidden and veryhidden.** In addition to creating dynamic names for files, I wanted a way of numbering them and having an ongoing incremented number. The purpose for implementing this was to allow for a system of automatic deletion later on. The way that I ended up doing this was by creating a veryhidden worksheet in my workbook that has a counter that I increment during the *processFile* sub-procedure. It made me feel sneaky and proud to have found such an awesome way to hide things from the non-coding user.
6. **Functions vs. Sub-procedures.** After finishing this project I feel like a master of the differences between functions and sub-procedures. Before I really was unsure of how to set them up, but after creating so many it feels like second nature.
7. **Using send-keys.** One of the last things that I did was learn how to use application.sendkeys. I only used it in one specific place, but I think there are likely some really helpful places for using the send keys and I look forward to using them more in the future.

Overall, I thought that the project was a great one for learning how to work with folders, to create and reuse methods, to work with workbooks, files, and folders and to just gain confidence throughout the entire process. Thanks again for the project, Dr. Allen!