HFAC Accounting Dashboard Program

This document includes detailed information about the Dashboard workbook that I created. This program was written for specific needs at my workplace in the HFAC accounting office on BYU campus.

This document includes an executive summary, a detailed overview, and a list of difficulties encountered when designing the program.

Executive Summary

The HFAC is full of artists. Brilliant artists, but not artists who care about accounting or expense reports. As such, we accountants have a *lot* of transactions to manage, and a lot of guiding professors to get what we need in order to process their purchases. In order to process transactions, we need to receive receipts for charges that show up on BYU's Y-Expense site. I work for three different departments, which only compounds the magnitude of transactions I need to manage, and receipts I need to request.

BYU's existing Y-Expense site (purchasing.byu.edu, a.k.a. Y-Expense) posts information about transactions, but in a way that requires lots of toggling, page switching, etc. Basically, it is very difficult to see *everything* that needs to be done. Users can only see pieces of it at a time.

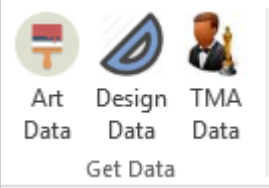
Beyond viewing the problem, it is time-consuming to communicate the problem to faculty and get receipts. With the amount of daily transactions spread throughout three departments (about 62 employees), we need to continually be in contact with faculty about their countless transactions. Writing individual emails for each faculty takes lots of time, but when we do not take that time to email everyone, we don't get receipts. After 15 days, unprocessed charges become university compliance issues. So transactions need to be processed less than 15 days from their creation.

The Dashboard workbook is an optimizer for working with Y-Expense transactions and communicating with faculty about documentation needs. This workbook automates essentially two processes: 1) interacting with Y-Expense to get *all* of the data we need and organize it into an easy-to-use report, and 2) filter through outstanding transactions and automatically email them to the corresponding faculty. This email will contain a standardized message, with a changing table of transactions.

Details about the inner workings of this program are provided hereafter, but a quick overview of the important workings of this program are provided in a table on the next page:

Note, the code for this program will only work for authorized employees in the HFAC department of Art, Design, or Theater and Media Arts. Currently, there should only be four people that this program will work perfectly for.

Also, faculty email addresses have been erased for privacy. The email macro of this program has been disabled, and email addresses that WOULD receive emails are printed in the immediate window of the VB editor. An example list of transactions has been put into the Y Expense sheet to test the email portion of the program.

Control Sub Procedure	Purpose	Ribbon Button
Art/design/tmaSub	• Connect to internet explorer • Log in to Y-Expense (controlled by user) • Directs internet explorer to transaction overview page	
consolidateControl	• Trigger download of online transaction data for specified department • For downloaded workbooks, bring data into the Dashboard workbook • Format data and autofilter for manipulation	
emailControl	• Using downloaded transaction information and saved faculty email list, send emails to all faculty who need to turn in receipts. • User Form appears to enter in faculty information for instance when new faculty are hired, or existing information is no longer accurate.	
deleteControl	• Clear the Y-Expense sheet for a new round	

	of downloads. See detailed explanation to understand why this is necessary	

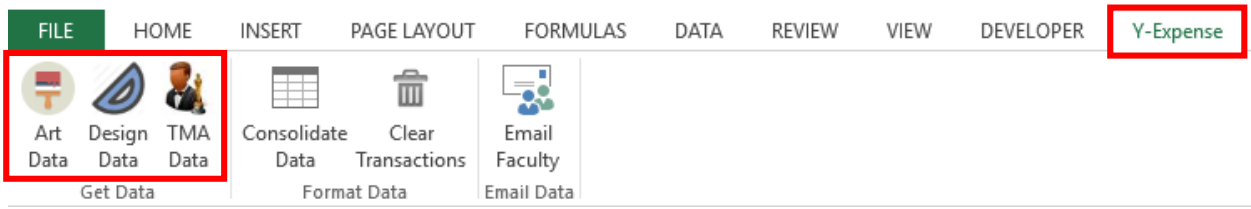
Detailed Overview

This program is primarily divided into four separate "control sub-procedures" designed to perform four basic functions:

1) Navigate Internet explorer with *Art,design,or TMASub:* this is used for navigating through Internet Explorer to the appropriate pages (depending on the department in question) where transaction information is available and downloading that information (contained in a workbook)
2) Summarize and format data with *ConsolidateControl:* This is used in bringing the summary data from the downloaded workbook into the Dashboard workbook, and formatting it such that it can be altered by additional programs
3) Email various faculty with *emailControl:* This sub procedure manages emailing faculty their outstanding transactions as determined from the downloaded and summarized information from Y-Expense.
4) Reset the worksheet with *deleteControl:* This deals with deleting the data to start over.

Navigating Internet Explorer

The first sub-procedure is divided into three separate control sub-procedures for each of the three departments I work for (Art, Design, and Theater and Media Arts, also called TMA). This is done because depending on the department the user is getting information for, a different url will be used. The user selects the department to get data simply by clicking on one of the toolbar buttons found in the Y-Expense tab of the ribbon (this tab was created specifically for this Dashboard).



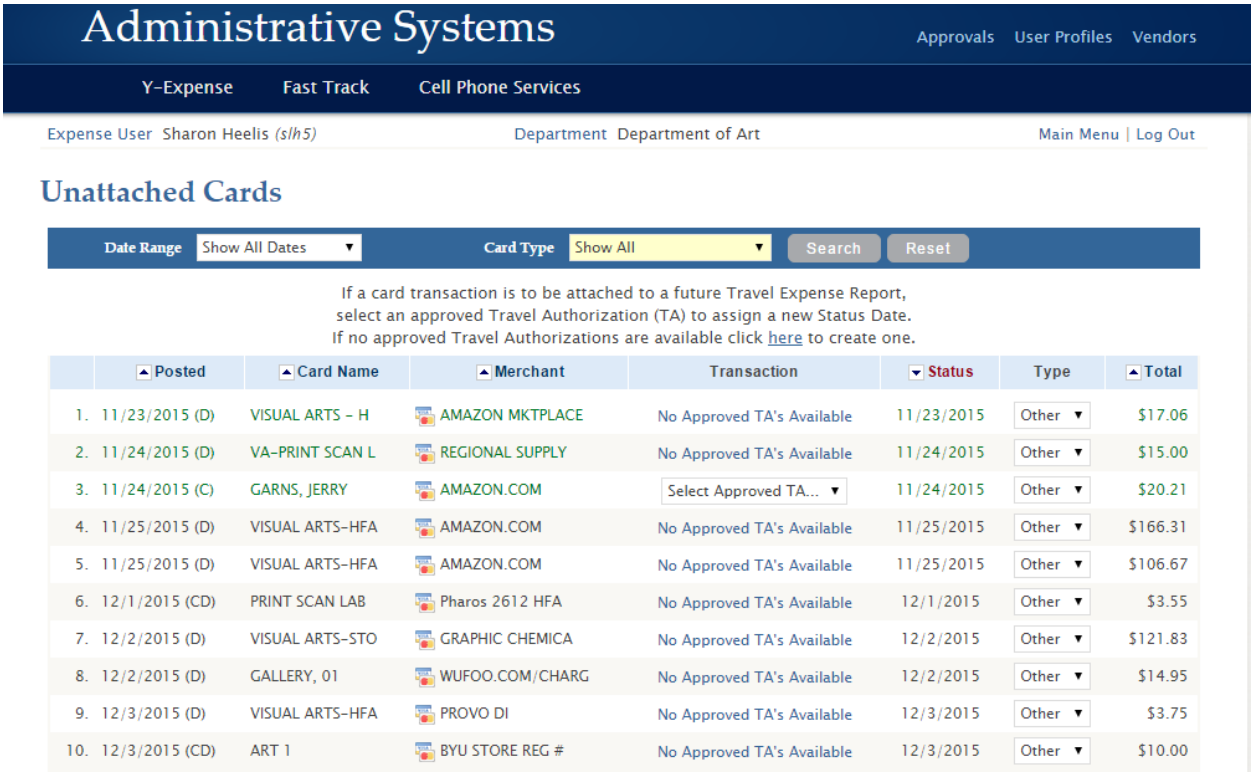
We will start by examining what happens when the user clicks the "Art Data" button to retrieve data from Y-Expense for the Art department.

artSub

Upon clicking this button, a control sub called "artSub" is called. This sub will vary depending on the department selected (designSub or TMAsub) and contains the following sub procedures (function of sub procedures also explained):

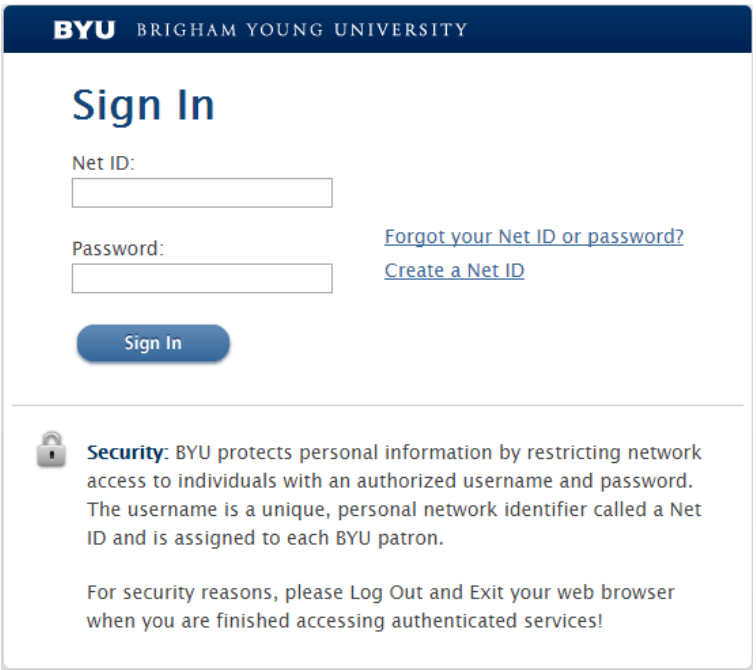
accessIE url

This sub procedure creates an Internet Explorer object with which we will gather data from Y-Expense. To do this, the object first is directed to a page of Y-Expense where the user can download summary data into a workbook for outstanding transactions (performed by passing the url variable specific to the department). This page is seen below:



	Posted	Card Name	Merchant	Transaction	Status	Type	Total
1.	11/23/2015 (D)	VISUAL ARTS - H	AMAZON MKTPLACE	No Approved TA's Available	11/23/2015	Other	$17.06
2.	11/24/2015 (D)	VA-PRINT SCAN L	REGIONAL SUPPLY	No Approved TA's Available	11/24/2015	Other	$15.00
3.	11/24/2015 (C)	GARNS, JERRY	AMAZON.COM	Select Approved TA...	11/24/2015	Other	$20.21
4.	11/25/2015 (D)	VISUAL ARTS-HFA	AMAZON.COM	No Approved TA's Available	11/25/2015	Other	$166.31
5.	11/25/2015 (D)	VISUAL ARTS-HFA	AMAZON.COM	No Approved TA's Available	11/25/2015	Other	$106.67
6.	12/1/2015 (CD)	PRINT SCAN LAB	Pharos 2612 HFA	No Approved TA's Available	12/1/2015	Other	$3.55
7.	12/2/2015 (D)	VISUAL ARTS-STO	GRAPHIC CHEMICA	No Approved TA's Available	12/2/2015	Other	$121.83
8.	12/2/2015 (D)	GALLERY, 01	WUFOO.COM/CHARG	No Approved TA's Available	12/2/2015	Other	$14.95
9.	12/3/2015 (D)	VISUAL ARTS-HFA	PROVO DI	No Approved TA's Available	12/3/2015	Other	$3.75
10.	12/3/2015 (CD)	ART 1	BYU STORE REG #	No Approved TA's Available	12/3/2015	Other	$10.00

However, this is only a page that authorized employees can visit. So, if the user has not logged in recently, Internet Explorer will automatically redirect the user to the sign in page seen below. Now, the program will actually be running a loop until the Locationurl property of the internet explorer object is not this sign in page. If it is, then Internet will become visible, allowing the user to log in to Y-Expense. Otherwise, it will just run in the background.



Once the user has logged in to Y-Expense, Internet explorer disappears, and this sub procedure ends. The control sub then calls the next sub procedure: downloadFile

downloadFile url

This sub procedure downloads the information we need. On the current internet page, VBA will push a button to download the file we need.[1] This is done by cycling through all of the elements of the Internet Explorer document by value and clicking on the value that is "View Report." It will also create a message box with some simple reminder instructions for the user to review. Once the user has closed the message box, Internet Explorer will once again appear. The program will be paused by another message box while the user interacts with Internet Explorer. All the user must do is press the "Open" button as prompted by Internet Explorer's popup window:[2]

Once this is clicked, the user is brought back to excel, where a message box instructs them to click "yes" on the system-generated prompt that will appear, and to use the "Consolidate data" tool before downloading any more data.

At this point, the file is downloaded and the macro ends. Note, that the internet explorer variable is set to nothing, but Internet Explorer is *not* quit. This is because the user will repeat this same process very soon with another department, and Internet Explorer spends some time to close in the background, which prohibits the macro from being run again.

---

[1] To do this, pop-ups need to be enabled for the site. On the first run, enable pop ups always, and click the "view Report" button again. The user must control this outside of VBA, controlling that was outside my capabilities.
[2] The user was instructed to do so in the aforementioned message box.

The user, as prompted, must proceed to the “Consolidate Data” button found in the toolbar. This concludes the first control sub.

Note. In this program, the sub procedure “waitForLoad” was used several times. This is a sub procedure created by Professor Gove Allen and included in his Agent object..

Summarize and Format Data

The previous sub procedure downloaded a file named “export” (typically concatenated with a number) containing all of the transaction data, and more. The next sub procedure will bring that information into the dashboard workbook, onto the sheet named “Y Expense.” In doing so, it will shed lots of noisy data that we don’t care about, and turn on the auto-filter mode. The detail of this process is outlined below:

<u>*ConsolidateControl*</u>

By clicking the above highlighted button in the above highlighted tab of the excel ribbon, the user will initiate the *ConsolidateControl* control sub procedure. This simply calls the following macros to perform their functions:

copyOverData

This sub procedure cycles through all open workbooks and finds any that have “export” in the first 6 letters of their name. It then uses a function to delete all undesired columns to prepare the data for import. Once the data has been prepared, the program will search for the next available cell on the Y Expense sheet in the Dashboard workbook (thus putting this data at the bottom of existing data rather than replacing old data. This is done so a user can view multiple departments combined).

Sizer

This sub procedure sizes the columns and rows of the newly imported data to make them more presentable

deleteHeaders

When data is imported and pasted in at the bottom of existing data, headers of the new data will be included. When pasted in, they will thus be put somewhere in the middle of the new data. This sub procedure cycles through all the rows of the combined data, looking for headers (by their title) and deletes the entire row that they are on.

deleteExports

This sub procedure deletes the export workbook, so that newly downloaded workbooks can be opened. This is why the user must use the “Consolidate Data” button after each download. Otherwise, excel won’t open two workbooks with the same name.

filterData

This sub procedure adds an auto-filter to the new data, if there is not one already in place.

The result of all of these sub procedures is a combined list of all transactions outstanding, accompanied by their faculty, dates of transactions, and vendors for each department selected.

	A	B	C	D
1				
2	POST_DATE	MERCHANT	USD_AMOUNT	EMPLOYEE
3	11/23/2015	AMAZON MKTPLACE PMTS	$17.06	Heelis, Sharon
4	11/24/2015	REGIONAL SUPPLY	$15.00	Heelis, Sharon
5	11/24/2015	AMAZON.COM	$20.21	Garns, Jerryl
6	11/25/2015	AMAZON.COM	$106.67	Heelis, Sharon
7	11/25/2015	AMAZON.COM	$166.31	Heelis, Sharon
8	12/1/2015 7:34:06 AM	Pharos 2612 HFAC 270F	$3.55	Schiffman, Sonya L
9	12/2/2015	GRAPHIC CHEMICAL & INK	$121.83	Heelis, Sharon
10	12/2/2015	WUFOO.COM/CHARGE	$14.95	Lanegan, Jason A
11	12/3/2015	BYU STORE REG #101	$10.00	Heelis, Sharon
12	12/3/2015	NATIONAL CAR TOLLS	$6.00	Barton, Garold C.
13	12/3/2015	Skyroom 301	$74.70	Schiffman, Sonya L
14	12/3/2015	PROVO DI	$3.75	Heelis, Sharon
15	12/7/2015	THE HOME DEPOT 4407	$192.22	Dean, David L
16	12/7/2015	CAFE ZUPAS PROVO	$82.80	Heelis, Sharon

Email Each Faculty Their* Specific *Outstanding Transactions

The next control sub procedure is very complex, and was thus put into its own module in VBA. This program will filter through the newly added Y-Expense data by faculty and, referencing a master list of faculty and emails, send emails to specific faculty containing their specific outstanding transactions.

emailControl

This control sub procedure calls the following sub procedures to carry out its calling in life:

createDataSheet

This creates a sheet called “Data” (first deleting any existing sheets called Data) for storing information on the faculty that need to be emailed and their email addresses.

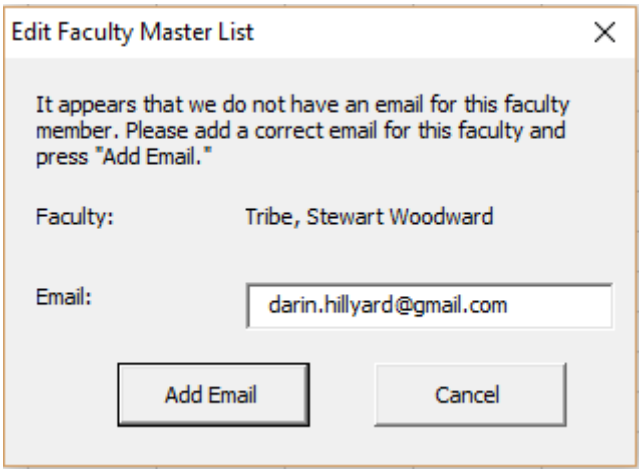
specificFaculty transSheet, masterlist

This sub procedure is passed two range variables. TransSheet is a worksheet variable referencing the sheet onto which the fresh Y Expense data was just downloaded, and masterlist is a range variable referencing two columns of data on the “Faculty Email List” Sheet. The first column is filled with faculty names. The second is filled with their corresponding emails.

First, this sub procedure will gather all of the faculty names from the list of transactions, remove duplicates, and paste them into the data sheet

Then, the sub procedure will populate the column next to these faculty with their corresponding emails from the “Faculty Email List” worksheet.

Next, the program will check the faculty and email list created on the data sheet for any missing emails. This represents a faculty who has not yet been entered into the master list on the “Faculty Email List” worksheet. For each missing email, the program will cycle through a user form that allows the user to input an email for the faculty in question.



Once the user clicks “Add Email,” this email will be added to both the data sheet (used for the current email process) and to the master list in the “Faculty Email List” sheet.

Finally, this program will set array variables for the faculty names and emails on the data sheet for use in the emailing sub procedure.

Note: in order to make this work, Professor Gove Allen wrote some code to clean the faculty names that were somehow being concatenated to asc character 160.

colorLate

This sub procedure colors the date text for the transactions that are more than 15 days old red so that faculty know which transactions to focus on (copy of message below).

filterEmployee x and sendMail x

The reason that I bunched these two sub procedures together is because they are both in a for loop. This loop runs for each value of x from 0 to the upper bound of the email array set earlier.

The filterEmployee sub procedure uses the autofilter on the Y expense sheet to only display data relating to the first faculty in the array set on the data sheet.

The sendMail sub procedure then copies that data, and pastes it after a standardized message in an HTML format. A copy of one such emails is below:

Hello, you are receiving this email because we need receipts from you for the charges below.

The dates of items that are more than 15 days old are in red text. Please make turning these in a priority.

POST_DATE	MERCHANT	USD_AMOUNT	EMPLOYEE
12/8/2015	DBC*BLICK ART MATERIAL	$214.46	Heelis, Sharon

This program loops through each employee on the Y Expense sheet and sends them an email containing their outstanding transactions.

Note: the sub procedure used to format this excel table so that it could be pasted into an email was created by Ron de Bruin. I simply copied the code and modified it to fit my project.

Delete Transactions and Reset the Project

The final sub procedure is very simple, but is necessary. Since data is being continually brought down from the internet and attached to a growing list of transactions, a separate procedure is needed to delete the list when it is done being added to. So, when a user wants to refresh the list, they will use this macro and go through the process from the beginning.

deleteControl

This sub procedure deletes the entire row for all of the transactions found on the Y Expense worksheet.

Greater Detail

If you are interested in greater detail feel free to examine the code of the project. Comments have been left to explain the function of each individual piece.

Difficulties in preparing this project

1. Working with Internet Explorer
   I chose to learn the hard way rather than use Professor Allen's agent object. Specifically, I had to learn how to press buttons and toggle through the site. Originally, I was downloading the pages as an HTML file and reading them with the web query wizard, but I decided it would be better to just download a workbook.

   Deciding to have the user download was painful, because I had to start down a whole new path of learning. I tried fruitlessly to avoid having to click through the ribbon that appears when a file is downloaded. I learned about using the "sendkeys" method, but had trouble making that work with different pages always getting in the way.

   I also had to learn to pause the macro in various ways while the user interacted with the internet. One way I paused was by creating a loop that would wait for the URL to change, and then proceed once it had (for logging in to a secure page). Unfortunately I left some code in that loop that continually pinged the website, and I was locked out for a day. That was a good lesson.

   The other way I learned to pause was by using message boxes. This was effective to wait for the user to interact with the internet browser, but also instruct the user how to do so.

2. Downloading files
   Since I download a file with the same name multiple times, I ran into issues with opening two workbooks with the same name. I could not figure out how to change the name of a file once it was downloaded, so rather I forced the user to first copy over and delete an existing workbook before downloading more files.

3. Converting excel tables into text in an email
   Rather than attach the transactions, I wanted the faculty to be immediately able to see what transactions they had to work on. This was a little more complicated than I expected, so simply found complex code to convert the cells into HTML formatted text.

4. I had some frustrations with strings that I had Professor Allen help me with. It turns out that a space is not always a space in a string, and I learned to consider the asc text that was in a string before jumping to conclusions. Comparing strings is DEFINITELY not as easy as comparing numbers.

5. General coding lessons:
   I had actually had this project in a completely different layout, everything beginning with a user

form. But, the discovery of ribbon manipulation changed EVERYTHING. I had to re-organize the layout of the project and how everything would trigger. This lead me to greater appreciate the value of having “control procedures” and making small subs that are called in order in a program. This way, it is easy to see where the sub starts, and how it progresses. This is especially helpful with all of the moving parts of using the internet.

6. Amazon Solver

   I actually have another program that I created for this dashboard. We have another problem in our department with Amazon charges. Amazon will send us a list of transactions, but will charge our account based on when things leave the warehouse. This means that we can be charged for things we did purchase, but in any random combination. This leads to lots of trouble when trying to reconcile a long list of Amazon charges to credit card charges.

   To solve this, I used a solver program that loops through Amazon transactions, and attempts to add up user-inputted receipts until it can find a solution. I got it to work, but I couldn’t connect it to the other programs in a clean way. That is something that will take a little more work.